Study on Nonlinear Regression with Missing Values Based on Hybrid Models Using Quasi-Linear Kernel

Huilin ZHU

November 2022

Waseda University Doctoral Dissertation

Study on Nonlinear Regression with Missing Values Based on Hybrid Models Using Quasi-Linear Kernel

Huilin ZHU

Graduate School of Information, Production and Systems
Waseda University

November 2022

Abstract

Nonlinear regression is a kind of regression analysis and has been widely used in many practical applications, such as health forecasting, environmental monitoring, and electric load forecasting. Moreover, the missing data problem, which is usually caused by mechanical and human factors, is a common issue encountered in predictive analytics. Improper processing of the missing information will directly affect the accuracy of forecasting. Thus, nonlinear regression analysis under missing data scenarios has become a prevailing problem in the research field. However, it is difficult in most cases to estimate all missing values with complete accuracy due to some reasons and the incorrect estimation result of missing data can lead to noisy data. Since some classical solutions are difficult to deal with such complex nonlinear relationships among samples, it is motivated to develop more robust and powerful regression models to solve the nonlinear regression problem with missing values.

As an extension version-based support vector machine (SVM), support vector regression (SVR) is adopted for regression. In this way, it is possible to determine how much error is acceptable and then match the data with an appropriate line or hyperplane in higher dimensions through kernel functions. However, SVR with implicit nonlinear kernel functions such as Gaussian kernel may cause severely over-fitting when processing datasets with characteristics of high noise.

To improve the robustness of the regression prediction model to noisy data, in this thesis, multi-local linear models or piecewise linear models are constructed. They are identified in the same way as an SVR with quasi-linear kernel composed using the data information as obtained in the fill-in missing values step. SVR with quasi-linear kernel is a nonlinear modeling method based on the divide-and-conquer strategy. In contrast to standard kernel functions, SVR with quasi-linear kernel can utilize information on data structure in nonlinear modeling. Therefore, two-part hybrid models can be applied to solve the nonlinear regression with missing values. On the one hand, partition information is modeled and utilized to generate gating mechanisms while using autoencoders to fill in missing values. On the other hand, gated linear networks are constructed to implement multi-local linear models or piecewise linear models by incorporating the gating mechanism, whose parameters are formulated by using SVR with quasi-linear kernel.

Though quasi-linear kernel has been utilized in classification tasks, it is still challenging to exploit it by tackling regression issues, especially in the case of missing values.

Therefore, in this thesis, using SVR with quasi-linear kernel, we propose a series of hybrid models to solve nonlinear regression problems under the missing data scenario.

Chapter 1 briefly introduces the background of the nonlinear regression problem under the missing data scenario and its research status. Then, we introduce two-part hybrid models and quasi-linear kernel. At last, several challenges are listed, and we propose different corresponding modeling methods in the following chapters.

Chapter 2 proposes a hybrid model consisting of an autoencoder and a gated linear network for solving the regression problem under the missing value scenario. A sophisticated modeling and identifying algorithm is developed. Firstly, an extended affinity propagation (AP) clustering algorithm is applied to obtain a self-organized competitive net dividing the datasets into several clusters. Secondly, a multiple imputation tool with top p% winner-take-all denoising autoencoders (DAE) is introduced to realize better predictions of missing values, in which rough estimates of missing values by using mean imputation and similarity method within the clusters are used as teacher signals of DAE. Finally, a gated linear network is designed to construct a local linear regression model with interpolations in the exact same way as an SVR with quasi-linear kernel composed using the cluster information obtained in the AP clustering step. Based on the experiments on five datasets, our proposed method demonstrates its effectiveness and robustness compared with other traditional kernels and methods with different percentages of missing data as 10%, 20%, 30%, 40%, 50%, and 60% when the missing are completely at the random.

Chapter 3 increases the role of autoencoders and proposes a winner-take-all (WTA) autoencoder-based piecewise linear model, which consists of two parts: an overcomplete WTA autoencoder and a gated linear network. The overcomplete WTA autoencoder is a stacked denoising autoencoder (SDAE) designed to play two roles: 1) to estimate the missing values; 2) to realize a sophisticated partitioning by generating a broad set of binary gate control sequences. Besides, an iterative algorithm with renewed teacher signals is developed to train the SDAE. On the other hand, the gated linear network with the generated binary gate control sequences implements a flexible piecewise linear model for nonlinear regression. By composing a quasi-linear kernel based on the gate control sequences, the piecewise linear model is then identified in the same way as a support vector regression. Two comparative experiments are conducted based on different missing data mechanisms. The accuracy and robustness of the proposed model have been verified by both experimental results of real-world datasets. Even for a large fraction of missing data, the role of our proposed model is also apparent.

Chapter 4 proposes an improved hybrid model based on previously proposed models to solve the nonlinear regression problem under missing data scenarios, consisting of two parts: an overcomplete WTA autoencoder and a multilayer gated linear network. The WTA autoencoder is trained in an adversarial training process by taking advantage of gradually renewed teacher signals and the discrimination of missing values and observed values, and is designed to play two roles: 1) to impute missing components conditioned on observed samples; 2) to generate gate control sequences. On the other hand, the multilayer gated linear network with the generated gate control sequences implements a powerful piecewise linear regression model, whose parameters are optimized by formulating an SVR with deep quasi-linear kernel. Experimental results about air quality datasets that originally have missing values in this chapter show that the proposed model achieves the best performance in each case. Moreover, results based on another missing data mechanism also prove that our proposed model yields the best prediction accuracy with a wide range of missing values.

Chapter 5 concludes the dissertation and provides future work. In summary, three different hybrid models are proposed to solve nonlinear regression problems under the missing data scenario. Numerical experimental results demonstrate the effectiveness and robustness of the proposed hybrid models.

Preface

The general theme of this dissertation is to solve nonlinear regression problems under missing data scenarios. This dissertation is organized in five chapters. Most of the materials have been published.

The material in Chapter 2 can be found in

- H. Zhu, Y. Tian, Y. Ren and J. Hu, "A Hybrid Model for Nonlinear Regression with Missing Data Using Quasi-Linear Kernel", *IEEJ Trans. on Electrical and Electronics Engineering*, Vol.15, No.12, pp.1791-1800, Dec 2020.

The material in Chapter 3 can be found in

- H. Zhu, Y. Ren, Y. Tian and J. Hu, "A Winner-Take-All Autoencoder Based Piecewise Linear Model for Nonlinear Regression with Missing Data", *IEEJ Trans. on Electrical and Electronics Engineering*, Vol.16, No.12, pp.1618-1627, Dec, 2021.
- H. Zhu, Y. Ren and J. Hu, "Establishing a Hybrid Piecewise Linear Model for Air Quality Prediction Based Missingness Challenges", in *Proc. of 2021 IEEE International Conference on System, Man, and Cybernetics* (SMC 2021) (Melbourne), pp.1705-1710, Oct 2021.
- H. Zhu and J. Hu, "Air Quality Forecasting Using SVR with Quasi-Linear Kernel", in *Proc. of the 2019 International Conference on Computer, Information and Telecommunication Systems* (CITS 2019) (Bejing), pp.1-5, Aug 2019.

The material in Chapter 4 can be found in

- H. Zhu and J. Hu, "An Improved Hybrid Model for Nonlinear Regression with Missing Values Using Deep Quasi-Linear Kernel", *IEEJ Trans. on Electrical and Electronic Engineering*, Vol.17, No.10, PP.1-9, 2022.

Acknowledgements

I would like to express my gratitude to all those who helped me during my PhD at Waseda University.

My deepest gratitude goes first and foremost to my supervisor, professor Jinglu Hu, for his constant encouragement and guidance. He has walked me through all the stages of my PhD stage. Without his consistent and illuminating instruction, I would not havefinished the degree.

Second, I am very grateful to other professors who proofread all the thesis. Many thanks are for their valuable comments.

Last my thanks would go to my beloved family for their loving consideration and great confidence in me all through these years. I also owe my sincere gratitude to my friends and all the group members in the Neurocomputing Lab (Xin Yuan, Hangyu Deng, Jingyu Yang, Xiao Fu et al.) who gave me their help and time in listening to me and helping me work out my problems. Thank you.

Kitakyushu, Japan

Huilin ZHU

June 1th, 2022

Contents

List of Figures

List of Tables

Abbreviations

AI	Artificial Intelligence
ANN	Artificial Neural Networks
AP	Affinity propagation
RMSE	Root Mean Squared Error
SVR	Support Vector Regression
SVM	Support Vector Machine
DAE	Denoising Autoencoder
SDAE	Stacked Denoising Autoencoder
KNN	*k*-Nearest Neighbor
EM	Expectation Maximization
MCAR	Missing Completely at Random
MAR	Missing at Random
MNAR	Missing not at random
RBF	Radial Basis Function
WTA	Winner-Take-All
ReLU	Rectified Linear Unit
MICE	Multivariate Imputation by Chained Equation
MLP	Multi-layer Perception
GAN	Generative Adversarial Nets
QP	Quadratic Programming
IQR	Interquartile Range
SICE	Single Center Imputation from Multiple Chained Equation

Symbols

x	Original input data
z	Completed dataset after filling-in missing data
Ω_j, b_j	Parameter set of the j-th base model
$\Omega_j^{(i)}, b_j^{(i)}$	Parameter set of the j-th base model in the i-th layer
δ_j	Local offset of local partition
σ	Radius of local partition
μ	Centers of local partition
M	The number of local partition
M_i	Number of linear base models (nodes) in the i-th layer
$\varphi(x)$	Mapping function transforming input space to feature space
Θ	Linear parameter vector
Φ	Regression vector
$K(x_k, x_l)$	Kernel function of SVR
$\sum$	Sum
$\alpha, \alpha*, \mu, \mu*$	Lagrange multipliers of SVR
$\mathbf{H}$	Hint mechanism
S	Number of layers in the multilayer gated linear network

Chapter 1

Introduction

1.1 Nonlinear Regression Problems with Missing Data

The objective of regression analysis in the research field is to establish a model which can examine the relationship between a response variable (y) and one or more independent variables (x). Among these, nonlinear regression is a kind of regression analysis in which the data is fit into a model and then expressed as a function $f(x) \rightarrow y$ [1, 2]. As is known to all, simple linear regression connects two variables as a straight line ($y = kx + b$), while nonlinear regression connects two variables as a curvilinear relationship. Nowadays, nonlinear regression analysis has been widely used in many piratical applications, such as utilizing the natural environment to complete health forecasting [3]. Therefore, it is important to develop effective and robust predicting models.

Another inevitable problem becoming more popular is the missing data problem [4, 5, 6], which has been a common phenomenon in the prediction domain. Missing data (or missing values) are defined as data values that are not stored while observing the variables of interest and can skew following problems from clinical trials to economic analysis. Firstly, missing data can lead to bias in parameter estimates. Secondly, it will reduce the representativeness of the sample. Thirdly, it may complicate the analysis of the research. Each distortion could threaten the validity of the test and could also

lead to invalid conclusions. Missing data is usually due to mechanical and human factors which include unobserved samples and observation. Improper processing of the missing information will directly affect the accuracy of forecasting. Before processing the missing data problem, it is necessary to understand the mechanism of missing data, which is divided into three main categories [7, 8]:

- Missing completely at random (MCAR) [9], if there are no relationships between the absence of missing data and the value itself or any other attributes, it can be classed as MCAR.
- Missing at random (MAR) [10], usually MCAR is the most common mechanism in related research, but there is another possibility that missingness only depends on other observed attributes but not its own value, which is corresponding to MAR.
- Missing not at random (MNAR) [11], when we say data are MNAR, it means that the propensity of the value to be missing is closely related to its own value.

Therefore, all these factors prove that missing data problems are ubiquitous and inevitable in the nonlinear research field. In this thesis, we analyze the research in recent years and propose innovative methods to tackle nonlinear regression problems under the missing data scenarios.

1.2 Research Status

In the literature, people always regard nonlinear regression problems with missing data as two separate problems, that is, the missing data problem is firstly solved and then they try to solve nonlinear regression problems based on fill-in datasets. Therefore, we try to introduce their research status from the perspective of two separate problems. Methods for solving missing values are described in Fig.1.1 and methods for tackling nonlinear regression problems are illustrated in Fig.1.2.

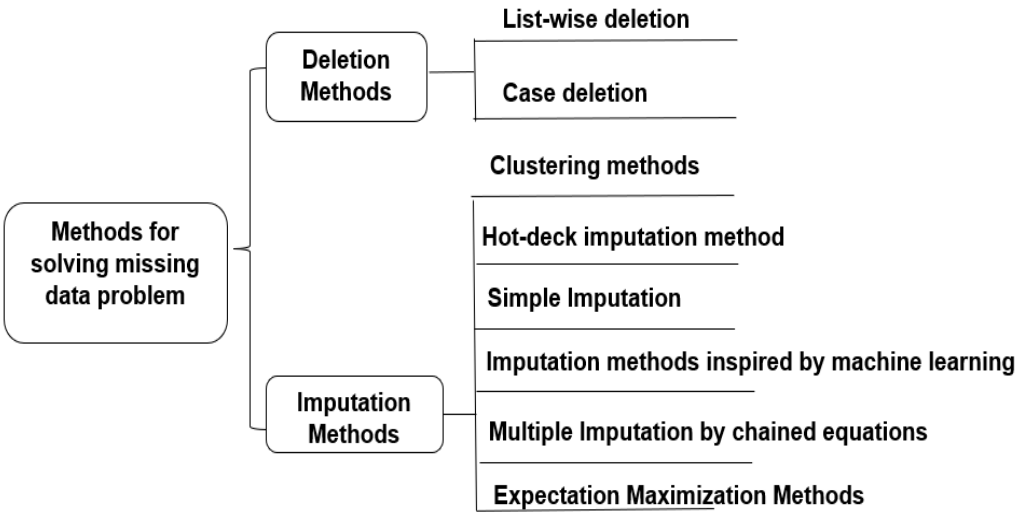


FIGURE 1.1: Methods for solving missing values

1.2.1 Approaches for Solving Missing Data Problems

Deletion Methods

The deletion method is the simplest method for dealing with missing values. According to different perspectives of data processing, it can be divided into two types: List-wise or case deletion [12]. Case deletion is suitable for small proportions of missing values [13], and List-wise deletion may work well when this attribute has little impact on the research objectives. In popular statistical software packages [14], List-wise deletion has become the default choice. Accordingly, the deletion method is only valid in the following circumstances:

- First Case: If the potential impact of the missing data is minimal, missing data may then be ignored in further analysis.
- Second Case: Only the dependent variable has missing values and it has little influence on the results.
- Third Case: Data satisfy the MCAR assumption.

When the proportion of missing data is large, especially when the missing data is not randomly distributed, the deletion method may lead to deviation of the data and lead to wrong conclusions.

Imputation Methods

Imputation methods are more common ways of dealing with missing values compared with discarding [15, 16]. Filling in the missing data through a certain method to form the complete dataset is crucial for subsequent data processing, analysis, and modeling. The commonly used imputation methods are as follows:

- Simple imputation

  Simple imputation requires replacing missing values for individual value by using the quantitative or qualitative attributes of all non-missing values [17]. With simple imputation, missing data can be treated in different ways, such as the mode, mean or median of available values. Though it is easy, this method may produce bias or unrealistic results when dealing with high-dimensional datasets. Moreover, with the generation of big data emerging, simple imputation has been proved that it is inadequate for such datasets because of poor performance.

- Hot-deck imputation method

  Hot-deck method handles missing data by matching the missing values with other complete samples which are the most similar in the dataset and then fills in missing values by using similar objects [18]. The hot-deck imputation method is well known in all single imputation methods because it produces rectangular data, which can be used by secondary data analysts. In addition, this method does not rely on model fitting to replace missing values, which makes it less delicate to model specification compared with parametric models (such as regression interpolation). The method also reduces the deviation of no-response. Although this method has been widely used, its concept is not mature enough compared with other imputation techniques. [19]

In Sullivan and Andridge [20], a hot-deck imputation method has been proposed, which allows for the investigation of the impact of missing mechanisms and uses the information contained in fully observed covariates. Bias and coverage of estimates from the proposed method have been also investigated by simulation. In another study by Christopher et al. [21], a fractional hot-deck imputation method has been proposed to deal with missing values, which is applicable to MAR. The proposed method produces a small standard error compared with list-wise deletion, mean imputation methods.

- Clustering method

  For missing data processing, clustering methods such as hierarchical clustering [22] and the k-means clustering [23] are generally tried in the literature. In study by Jocelyn T. Chi et al. [24], a missing data imputation method based on the k-means clustering technique has been proposed, which is divided into two steps. In the first step, k-means clustering is used to obtain clusters, and then missing values are processed with clustering information. This method is somewhat similar to the hot-deck method. If the nearest neighbor samples only consider the nearest sample, it will degenerate into hot-deck method. Besay Montesdeoca et al. [25] then proposed a big data k-means clustering and a big data fuzzy k-means missing values approach which provides robust and efficient output for big data and reasonable execution time. The fuzzy k-means method was proved to provide better results with high percentages of missing values in the data, while the k-means method performed better on the dataset with lower amounts of missing values. Zhang et al. [26] also proposed multiple imputation clustering-based approach to deal with missing values in large-scale longitudinal test data of e-Health. The results show that it could be adapted for different kinds of clustering in e-Health services.

- Expectation maximization method

  Expectation maximization (EM) imputation is an iterative algorithm for dealing with missing values in numerical datasets, which uses the approach of ”impute, estimate and iterate until convergence” [27]. Each iteration consists of two

phases: expectation and maximization. The expectation step estimates the missing value given observed samples, while in maximization, the estimated values are used to maximize the probability of all data [28]. In Rubin et al. [29], the method of handling missing data has been investigated using a dataset that analyzed the effects of feeding behavior in drug-treated and untreated animals. The EM method has been used and compared with other methods. The authors then concluded that the EM algorithm is the best method for the type of data they use. However, it may lead to the results being specific to idiosyncrasies in the dataset. The EM algorithm has been also used to tackle the problem of training Gaussian mixture in large high-dimensional datasets with missing values in another study [30]. The results showed that the performance was significantly improved compared with other basic imputation methods. However, it has led to expensive matrix calculations.

- Imputation methods inspired by machine learning

  With the advent of the era of big data, it is difficult for traditional learning methods to deal with missing data. Computational methods based on machine learning are complex techniques and mainly involve developing a predictive method that uses unsupervised or supervised learning to deal with missing values. These techniques try to handle missing values depending on the information obtained from the non-missing values in the data using unlabelled or labeled data [31]. In most cases, if available samples have useful information to deal with missing values, high predictive precision can be maintained.

  State-of-the-art imputation algorithms for solving missing data problems are primarily based on deep learning since latent relationships among samples can be learned [32]. The denoising autoencoder (DAE)-based approaches [33, 34, 35] work well in practice but require complete data during the training process. In many cases, missing values are inherent problems of the structure, and obtaining a complete dataset is impossible. In the existing methods, mean imputation is often used as teacher signals during training the DAE. In [36], a generative model for missing data imputation is proposed, and various experiments show that it is obviously superior to state-of-art imputation algorithms.

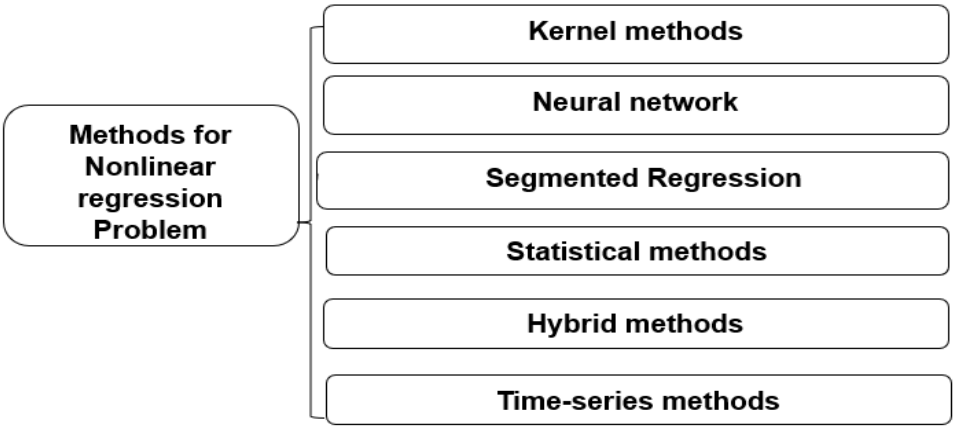


FIGURE 1.2: Methods for solving nonlinear regression problems

- Multiple imputation by chained equations method

  Multiple imputation by chained equations method is a kind of traditional multiple imputation algorithms [37, 38]. It is used to replace missing data under the MAR or MCAR assumption, that is the probability of missing a value depends only on the observed values, not on the unobserved values. Many of the original multiple imputation algorithms assumed large joint models of all variables, such as joint normal distributions. MICE is an alternative, flexible method to these joint models and has been applied in datasets with thousands of observations hundreds of variables [39, 40]. In the MICE, a series of regression models are run where each sample with missing data is modeled from other variables in the dataset, which means that each variable can be modeled based on its distribution. For instance, binary variables can be modeled with logistic regression, and continuous variables can be modeled with linear regression.

1.2.2 Methods for Solving Nonlinear Regression Problems

Neural Network Methods

Neural network methods are based on mathematical models of the human brain [41]. There are complex relationships between networks of prediction methods that allow for response variables. In the literature, many researchers have proposed forecasting

methods using machine learning algorithms instead of traditional classical methods. Artificial neural networks (ANN) can well understand the nonlinear mechanism of atmospheric phenomena and have high predictive performance, so it is widely used in forecasting research [42, 43]. Since then, researchers have focused their attention on the improvement of prediction accuracy and started to focus on the development of hybrid models combining various optimization methods with ANN instead of traditional simple neural networks [44, 45]. Currently, as a field of machine learning for artificial intelligence, deep learning has been successfully applied to computer vision [46], speech recognition [47], natural language processing, and so on. In addition, it is also widely used for nonlinear regression [48]. Despite the increased complexity of neural network architectures for modern applications, they still consist of a combination of basic structures such as multi-layer perceptions (MLP), recurrent neural networks, and convolutional neural networks, which are well known and explored for decades. Nevertheless, these models are all edified with complete datasets.

Statistical Methods

In statistics, the goal of the nonlinear regression problem is to make the sum of squares as small as possible. In nonlinear regression, when the sum of squares of residuals is the smallest, the likelihood can be maximized [49]. For the nonlinear regression, the forecasting model depends on one or more parameters nonlinearly. In general, statistical methods can be applied when the relationship between variables follows the function in a specific way or the dataset is very small.

Segmented Regression

Segmented regression, which is known as broken-stick regression or piecewise regression, is a popular method in regression analysis. The independent variables are divided into several intervals, and each interval is fitted with a separate line segment. The segmented regression method can also analyze multivariate data by dividing various independent variables. It may have better results especially when independent variables

are clustered into different groups and show different relationships between variables in these regions. The boundaries between line segments can be regarded as breakpoints. Moreover, segmented linear regression, also known as the piecewise linear regression method [50], is a kind of segmented regression method whereby the relations among the intervals are acquired by linear regression analysis.

Kernel Methods

Kernel methods are a kind of algorithm, and their importance has increased in the machine learning field since the 1990s. The most famous example of kernel methods is supoort vector machine (SVM) [51], which is the latest technology for classification problems. Kernel methods do not care about data structure or dimension. They are defined by operating on the kernel function. For the regression problem, the re-expression of SVM is soon produced named support vector regression (SVR) [52], but SVR has some problems related to its expression and efficiency. Least squares support vector machine (LS-SVM) [53] or kernel ridge regression [54] is an improvement of standard SVM, trying to overcome these shortcomings. Kernel ridge regression combines ridge regression (linear least squares and L2 norm regularization) with the kernel technique. Therefore, it learns linear functions in the space induced by their respective kernels and data. For the nonlinear kernel, it corresponds to the nonlinear function in the original space.

Time Series Methods

The time series problem is a kind of typical nonlinear regression problem which is ordered lists of parameters or variables and provided at equal time intervals. Forecasts are continuous patterns over time such as sales growth, stock market analysis, or gross national product [55]. The most common time-series method is named Autoregressive Integrated moving average which was first proposed in the early 1970s and can divide the time-series problem into three phases: identification, testing and estimation, and application. With the popularity of neural networks, the most commonly used methods for

solving time series problems are recurrent neural networks and long short-term memory. The recurrent neural network is a generalization of the feedforward neural network but it can utilize internal state (memory) in order to process sequences of inputs. The long term short term memory network is an improved version of the recurrent neural network that makes it easier for memory to remember past data [56].

Hybrid Methods

In recent years, researchers have become interested in combining different methods for more accurate predictions. These combinations are also called hybrid models. Each hybrid model consists of completely independent and effective prediction methods. The main purpose is to use their characteristics for reducing the prediction error. For instance, in [57], authors used back-propagation neural network and GM (1,1). In order to combine these methods, appropriate weights should be assigned. In this work, the weight is calculated based on Shapley value distribution.

1.3 Hybrid Modeling Methods

In most real applications, the estimation of missing values cannot achieve the 100% because of specific reasons and the imperfect estimation result of missing data can be served as noisy data. Some classical solutions are difficult to model such complex nonlinear relationships among samples. Therefore, it is motivated to develop more robust and powerful regression models to solve the nonlinear regression problem with missing values.

In this thesis, we construct multi-local linear models or piecewise linear models because of their robustness to noise data, and they are identified in the exact same way as an SVR with the quasi-linear kernel composed using the data information obtained in the fill-in missing values step. SVR with quasi-linear kernel is a nonlinear modeling method based on *divide-and-conquer* strategy, which is utilized to solve nonlinear regression problems. Different from standard kernel functions that are black-box models

and unable to incorporate prior knowledge of datasets, SVR with quasi-linear kernel can utilize information on data structure in nonlinear modeling. Therefore, two-part hybrid models can be proposed to solve the nonlinear regression with missing values. In one part, partition information is modeled and utilized to generate gating mechanisms while using autoencoders to fill in missing values. For the other part, gated linear networks are constructed to implement multi-local linear models or piecewise linear models by incorporating the gating mechanism, whose parameters are formulated by using SVR with quasi-linear kernel. In this way, SVR with quasi-linear kernel can realize nonlinear regression by using a composed data-dependent kernel. To easily understand it, we first briefly address SVR with general kernels.

1.3.1 SVR Based Regression

SVR is an extension version-based SVM that is used for regression. It maintains all the main features that fit the algorithm. In the case of regression, the main idea is to minimize error, individualizing the hyper-plane which maximizes the margin.

Consider a d-dimensional dataset $S=\{(\mathrm{x}_1,\mathrm{y}_1),...,(\mathrm{x}_n,\mathrm{y}_n)\}$, where $x_i \in R^d$ is the i-th input feature vector and $y_i \in R$ is the target output of i-th sample. A regression function is defined by using a kernel function as:

$$f(x) = w^T \varphi(x) + b \tag{1.1}$$

where $\varphi(\cdot)$ denotes the feature function which maps data from input data to a high-dimensional feature space.

In the following, we will focus on how to estimate linear parameters using SVR formulation. Based on the structural risk minimization principle as:

$$\begin{aligned} &\min_{w,b,\xi_i,\xi_i^*} \frac{1}{2} w^T w + C \sum_{i=1}^{N} (\xi_i + \xi_i^*) \\ &\text{s.t.} \begin{cases} w^T \varphi(x) + b - y_i \leq \epsilon + \xi_i^* \\ y_i - w^T \varphi(x) - b \leq \epsilon + \xi_i \\ \xi_i, \xi_i^* \geq 0, \;\; i = 1, 2, ..., N \end{cases} \end{aligned} \tag{1.2}$$

where y_i denoted the ideal output of z_i, C is a non-negative weight to determine the penalization of prediction errors, N is the number of observations, and ξ_t, ξ_t^* are slack variables. By introducing Lagrange multipliers $\mu \geq 0$, $\mu* \geq 0$, $\alpha \geq 0$, $\alpha* \geq 0$, we can construct the Lagrange function as:

$$\begin{aligned} &L(w, \xi_t, \xi_t*, \alpha, \alpha*, \mu, \mu*) = \frac{1}{2} w^T w + C \sum_{i=1}^{N} (\xi_i + \xi_i^*) \\ &+ \sum_{i=1}^{N} \alpha_i (f(x_i) - y_i - \epsilon - \xi_t) + \sum_{i=1}^{N} \alpha_i (-f(x_i) + y_i - \epsilon - \xi_i^*) - \sum_{i=1}^{N} (\mu \xi_i + \mu^* \xi_i^*) \end{aligned} \tag{1.3}$$

Then it can be solved through getting the saddle point:

$$\begin{aligned} &\frac{\partial L}{\partial w} = 0 \rightarrow \Theta = \sum_{i=1}^{N} (\alpha - \alpha^*) \varphi(x_i) \\ &\frac{\partial L}{\partial \xi} = 0 \rightarrow C = \alpha + \mu \\ &\frac{\partial L}{\partial \xi^*} = 0 \rightarrow C = \alpha^* + \mu^* \end{aligned} \tag{1.4}$$

After converting the Lagrange function into its dual problem, we can get:

$$\begin{aligned} \max\ W(\alpha,\alpha^*) = &-\frac{1}{2}\sum_{i,j=1}^{N}(\alpha_i-\alpha_i^*)(\alpha_j-\alpha_j^*)K(x_i,x_j) \\ &+\sum_{i=1}^{N}(\alpha_i-\alpha_i^*)y_i-\epsilon\sum_{i=1}^{N}(\alpha_i+\alpha_i^*) \\ \text{s.t.}\ \ &\sum_{i=1}^{N}(\alpha_i-\alpha_i^*)=0.\quad \alpha,\alpha^*\in[0,C]. \end{aligned} \tag{1.5}$$

where $K(x_i,x_j)=\varphi^T(x_i)\varphi(x_j)$ is the kernel function. From the above, with the Lagrange multipliers α_i and α_i^* obtained, the regression model can finally be represented as:

$$f(z)=\sum_{i=1}^{N}(\alpha_i-\alpha_i^*)K(x,x_i)+b \tag{1.6}$$

Traditional basis kernel function includes Linear Kernel, Polynomial Kernel and Radial Basis Function (RBF) Kernel [58, 59], defined by:

$$K(x_i,x_j)_{linear} = x_i^T x_j \tag{1.7}$$

$$K(x_i,x_j)_{poly} = (1+x_i^T x_j)^d \tag{1.8}$$

$$K(x_i,x_j)_{RBF} = \exp\left(-\frac{\|x_i-x_j\|^2}{\sigma^2}\right) \tag{1.9}$$

For SVR, choosing an explicit kernel function is very important, which maps the sample from the original space to another high-dimensional feature space. However, SVR with implicit kernel mapping such as RBF kernel may not have a good performance on nonlinear regression problems, since the dataset is always large and complex which may cause over-fitting problems.

1.3.2 SVR with Quasi-Linear Kernel

Suppose a given dataset whose sample is $x \in R^d$. We then build the gated linear network realizing a multi-local linear model, which is defined by:

$$f(x) = \sum_{j=1}^{M} \left(\Omega_j^T x + b_j\right) g_j(x) + b \tag{1.10}$$

where M is the number of linear base models $\Omega_j^T x + b_j$, $\{\Omega_j, b_j, b\}$ is the parameter set, and $R_j(x) \in \{0, 1\}$ is a gate signal controlling whether the j-th base model works.

In order to construct Eq.(1.10), the functions $g(x)$ need to be evaluated. B.Zhou et al [60] used the RBF to formulate $g(x)$, defined by:

$$\tilde{g}_j(x) = e^{-\frac{(x-\mu_j)^2}{\lambda\sigma_j^2}}, \quad g_j(x) = \frac{\tilde{g}_j(x)}{\sum_{i=1}^{M} \tilde{g}_i(x)} \tag{1.11}$$

where M is the number of local linear partitions, λ is an appropriate scale parameter, μ_i and σ_i are i-th local-linear cluster's center and width which need to be evaluated.

After that, W. Li et al. [61] proposed a new method to generate the gated signals by pre-training a winner-take-all autoencoder. The gated signals can be regarded as outputs of autoencoder, defined by:

$$g_j(x) = \begin{cases} 1, & \boldsymbol{w}_j^T x + \theta_j > 0 \\ 0, & \boldsymbol{w}_j^T x + \theta_j \leq 0 \end{cases} \quad j = 1, \ldots, M. \tag{1.12}$$

where $\boldsymbol{w}_j$ and θ_j are the weights and bias and M is the number of nodes of the pre-trained autoencoder.

By importing two vectors Φ(z) and Θ defined as:

$$\Phi(x) = \left[g_1(x), x^T g_1(x), \ldots, g_M(x), x^T g_M(x)\right]^T \tag{1.13}$$

$$\Theta = \left[b_1, \Omega_1^T, \ldots, b_M, \Omega_M^T\right]^T \tag{1.14}$$

Eq.(1.10) can be expressed as a linear-in-parameter form as:

$$f(x) = \Theta^T \Phi(x) + b. \tag{1.15}$$

where Φ(x) is the regression vector and Θ is called linear parameter vector. In the following, like the standard SVR, we concentrate on how to estimate linear parameters using SVR formulation. Based on the structural risk minimization principle, a QP optimization problem is obtained:

$$\begin{aligned} &\min_{\Theta,b,\xi_i,\xi_i^*} \frac{1}{2}\Theta^T\Theta + C\sum_{i=1}^{N}(\xi_i + \xi_i^*) \\ &\text{s.t.} \begin{cases} \Theta^T\Phi(x) + b - y_i \le \epsilon + \xi_i^* \\ y_i - \Theta^T\Phi(x) - b \le \epsilon + \xi_i \\ \xi_i, \xi_i^* \ge 0, \quad t = 1,2,...,N \end{cases} \end{aligned} \tag{1.16}$$

By applying the Lagrange function through introducing Lagrange multipliers $\mu \ge 0$, $\mu* \ge 0$, $\alpha \ge 0$, $\alpha* \ge 0$, the QP optimization problem can be converted into its dual problem , we can get:

$$\begin{aligned} \max\ W(\alpha,\alpha^*) = &-\frac{1}{2}\sum_{i,j=1}^{N}(\alpha_i - \alpha_i^*)(\alpha_j - \alpha_j^*)K(x_i,x_j) \\ &+ \sum_{i=1}^{N}(\alpha_i - \alpha_i^*)f(x) - \epsilon\sum_{i=1}^{N}(\alpha_i + \alpha_i^*) \\ \text{s.t.}\quad &\sum_{i}^{N}(\alpha_i - \alpha_i^*) = 0. \quad \alpha,\alpha^* \in [0,C]. \end{aligned} \tag{1.17}$$

where $K(x_i, x_j)$ is a data-dependent composed kernel called quasi-linear kernel, defined as:

$$\begin{aligned} K(x_i,x_j) &= \Phi^T(x_i)\Phi(x_k) \\ &= (1 + x_i x_j)\sum_{k=1}^{M} g_k(x_i)g_k(x_j) \end{aligned} \tag{1.18}$$

From the above, with the Lagrange multipliers α_i and α_i^* obtained, the regression model can finally be represented as:

$$f(x) = \sum_{i=1}^{N} (\alpha_i - \alpha_i^*) K(x, x_i) + b. \tag{1.19}$$

1.3.3 SVR with Deep Quasi-Linear Kernel

Then an extension version has been proposed in Ref. [62], and it takes advantage of the pre-trained deep neural network to achieve the task. The multilayer gated linear network realizing a powerful piecewise linear model is defined by [62, 63]:

$$f(x) = \sum_{j=1}^{M_S} \left(\Omega_j^{(S)T} a_{S-1}(x) + b_j^{(S)} \right) g_j^{(S)}(x) + b \tag{1.20}$$

$$a_i(x) = \sum_{j=1}^{M_i} \left(\Omega_j^{(i)T} a_{i-1}(x) + b_j^{(i)} \right) g_j^{(i)}(x) \tag{1.21}$$

$$i = 1, 2, ..., S-1; \quad a_0(x) = x$$

where S is the number of layers in the multilayer gated linear network, M_i is the number of linear base models $\Omega_j^{(i)T} a_{i-1}(x) + b_j^{(i)}$ in the i-th layer, $\{\Omega_j^{(i)}, b_j^{(i)}, b\}$ is the parameter set, and $g_j^{(i)}(x) \in \{0, 1\}$ is a gate signal controlling whether the j-th base model in the i-th layer works.

When the gate signals are given, the multi-layer gated linear network can reduce to a linear model. By denoting $\Phi_0 = x$ and $\Theta_0 = []$, we import two vectors $\Phi_S(\mathrm{x})$ and Θ_S defined as:

$$\Phi_i(x) = [g_1^{(i)}(x), \Phi_{i-1}^T(x) g_1^{(i)}(x), ..., g_{M_i}^{(i)}(x), \Phi_{i-1}^T(x) g_{M_i}^{(i)}(x)]$$

$$= [\mathbf{g}^{(i)T}(x) \otimes [1 \;\; \Phi_{i-1}^T(x)]]^T \tag{1.22}$$

$$\Theta_i = [\mathbf{\Omega}^{(i)T} \otimes [1 \;\; \Theta_{(i-1)}^T]]^T \quad (i = 1, 2, ..., S) \tag{1.23}$$

where $\mathbf{g}(x) = [g_1(x), ..., g_M(x)]^T$, and $\otimes$ represents Kronecker production, $\mathbf{\Omega}^i = [b_1^{(i)}, \Omega_1^{(i)T}, ..., b_{M_i}^{(i)}, \Omega_{M_i}^{(i)T})]^T$.

$\Phi_S(x)$ gives a multi-linear mapping since for each given $\mathbf{g}^{(i)}(x)$ it is a linear mapping. Therefore, the S-layer gated linear network can be compactly expressed as a linear-in-parameter form as:

$$f(x) = \Theta_S^T \Phi_S(x) + b \tag{1.24}$$

Same as subsection 1.3.2, the Eq.(1.24) can be optimized by using SVR with the deep quasi-linear kernel in a recursive form defined by:

$$\begin{aligned} K_i(x_l, x_j) &= {\Theta_i}^T(z_l)\Theta_i(x_j) \\ &= (1 + K_{i-1}(x_l, x_j)){\mathbf{g}^{(i)}}^T(x_l)\mathbf{g}^{(i)}(x_j) \\ (i &= 1, ..., S) \end{aligned} \tag{1.25}$$

where $K_0(x_l, x_j) = x_l^T x_j$. Therefore, the regression model can finally be represented as:

$$f(x) = \sum_{l=1}^{N} (\alpha_l - \alpha_l^*) K_S(x, x_l) + b \tag{1.26}$$

1.4 Challenges

As mentioned above, since we try to propose two-part hybrid models consisting of autoencoders and gated linear networks, from the modeling perspective, we need to consider the following three issues:

- Unlike traditional methods, since missing values are existed, accurate teacher signals are needed before training the overcomplete autoencoder. Improper processing of teacher signals will directly affect the accuracy of forecasting. Therefore, a more suitable and accurate method should be used as the preprocessing procedure to generate accurate teacher signals especially the proportion of missing data is large. Moreover, how to better train autoencoders to fill in missing values should be considered simultaneously.

- Considering gating mechanisms, since there are many ways to realize partition, the key points we need to consider are which partitioning method is more suitable for application in our proposed model and how to utilize data information to generate gate mechanisms.
- Finally, how to formulate a support vector regression (SVR) with quasi-linear kernel to optimize parameters should also be considered.

1.5 Goals of the Thesis

The main goal of this research is to focus on solving nonlinear regression problems with missing data. In this thesis, two-part hybrid modeling methods consisting of autoencoders and gated linear networks are developed. More precisely, autoencoders are utilized to estimate missing values and gated linear networks are constructed to implement multi-local linear models or piecewise linear models by incorporating the gating mechanism, whose parameters are optimized by formulating SVR with quasi-linear kernel.

Though the quasi-linear kernel has been utilized in classification tasks, it is still challenging to exploit it by tackling regression issues, especially in the case of missing values. Therefore, according to the different solutions to the above three problems, we propose a series of hybrid models to solve nonlinear regression problems under the missing data scenario.

The work presented here aims to assess the performances of the hybrid models compared with traditional modeling methods based on different missing mechanisms. The thesis also shows the performance of proposed models even in dealing with large portions of missing values.

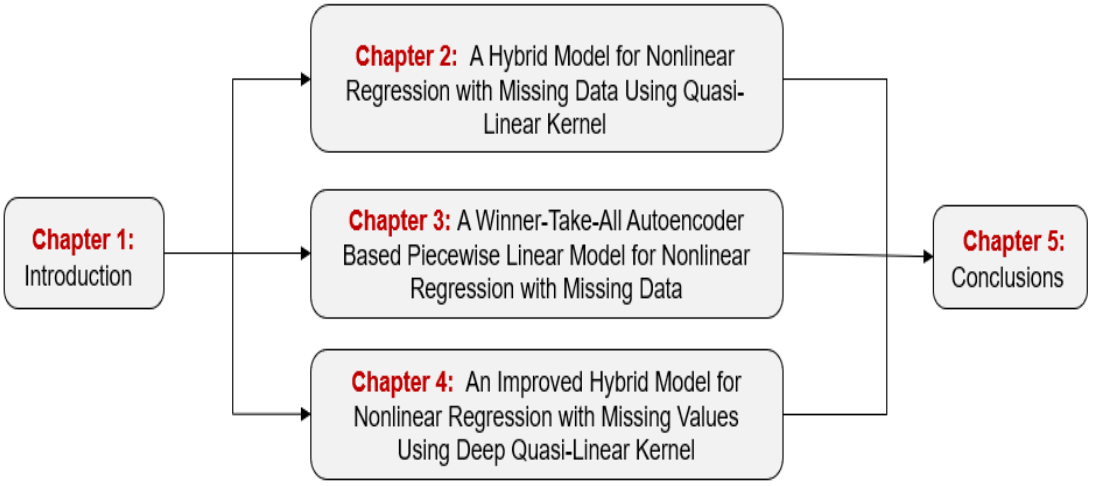


FIGURE 1.3: Flow diagram of this thesis

1.6 Thesis Outlines and Main Contributions

This thesis is divided into five chapters to introduce my work accumulated in my doctoral career. Chapter 1 introduces the background, related work, and an outline for the whole thesis. Chapter 2 proposes a hybrid modeling method consisting of the AP clustering step, an autoencoder, and a gated linear network for solving the regression problem under the missing value scenario. Chapter 3 proposes a winner-take-all (WTA) autoencoder-based piecewise linear model consisting of an overcomplete WTA autoencoder and a gated linear network to solve the nonlinear regression problem with missing data. Chapter 4 proposes an improved hybrid model based on previous work to solve the same problem. Chapter 5 makes the conclusion for the whole thesis and illustrates future research directions. The flow of the structure is illustrated in Fig.1.3.

This thesis summarizes the research on the nonlinear regression problem with missing data. The main contents and contributions are shown as follows:

Chapter 2 proposes a hybrid model consisting of an autoencoder and a gated linear network for solving the regression problem under missing value scenario. A sophisticated modeling and identifying algorithm is developed. Firstly, an extended affinity propagation (AP) clustering algorithm is applied to obtain a self-organized competitive net dividing the datasets into several clusters. Secondly, a

multiple imputation tool with top $p\%$ winner-take-all DAE is introduced to realize better predictions of missing values, in which rough estimates of missing values by using mean imputation and similarity method within the clusters are used as teacher signals of DAE. Finally, a gated linear network is designed to construct a piecewise linear regression model with interpolations in an exact same way as a SVR with a quasi-linear kernel composed using the cluster information obtained in the AP clustering step. Based on the experiments on five datasets, our proposed method demonstrates its effectiveness and robustness compared with other traditional kernels and state-of-the-art methods even on datasets with large percentage of missing values.

The main contributions related to this model are shown as follows:

- This paper is the first to present a hybrid model which combines winner-take-all DAE and a gated linear network to solve forecasting problem with missing data.
- To the best our knowledge, our studies on filling in missing data before training DAEs firstly use AP clustering algorithm for constructing the self-organized competitive net, which makes DAEs get more accurate and effective information during training, and as to the regression phase, we can also make full use of cluster information efficiently to build a local linear prediction model.

Chapter 3 proposes a WTA autoencoder-based piecewise linear model to solve the nonlinear regression problem under the missing value scenario, which consists of two parts: an overcomplete WTA autoencoder and a gated linear network. The overcomplete WTA autoencoder is a stacked denoising autoencoder (SDAE) designed to play two roles: 1) to estimate the missing values; 2) to realize a sophisticated partitioning by generating a broad set of binary gate control sequences. Besides, an iterative algorithm with renewed teacher signals is developed to train the SDAE. On the other hand, the gated linear network with the generated binary gate control sequences implements a flexible piecewise linear model for nonlinear regression. By composing a quasi-linear kernel based on the gate control

sequences, the piecewise linear model is then identified in the same way as a support vector regression.

The main contributions are that:

- By increasing the role of the denoising autoencoder we improve the overcomplete WTA autoencoder which plays two roles: 1) to estimate the missing values as a multiple imputation tool; 2) to realize a sophisticated partitioning by generating a broad set of binary gate control sequences using the feature layer of SDAEs.
- By using the binary gate control sequences, the gated linear network implements a flexible piecewise linear model for the nonlinear regression.

Chapter 4 proposes an improved hybrid model to solve the nonlinear regression problem under missing data scenarios, consisting of two parts: an overcomplete WTA autoencoder and a multilayer gated linear network. The WTA autoencoder is trained in an adversarial training process by taking advantage of gradually renewed teacher signals and the discrimination of missing values and observed values, and is designed to play two roles: 1) to impute missing components conditioned on observed samples; 2) to generate gate control sequences. On the other hand, the multilayer gated linear network with the generated gate control sequences implements a powerful piecewise linear regression model, whose parameters are optimized by formulating a SVR with a deep quasi-linear kernel. Experimental results based on different real-world datasets demonstrate the effectiveness of our proposed hybrid model.

The main contributions are that:

- By using the WTA autoencoder as a generator and introducing a discriminator, we can expect a better adversarial training of the WTA autoencoder by taking advantage of gradually renewed teacher signals and the discrimination of missing values and observed values.

- By using a multilayer gated linear network and the generated layered gate control sequences, we implement a more powerful piecewise linear regression model, whose parameters are then optimized by formulating a SVR with a deep quasi-linear kernel in a recursive form

Chapter 5 concludes this work, summarizes the thesis and gives suggestions for further research.

Chapter 2

A Hybrid Model for Nonlinear Regression with Missing Data Using Quasi-Linear Kernel

2.1 Introduction

[1] The objective of regression analysis in research field is to establish a model which can examine the relationship between variable of interest. Therefore, it is important to develop accurate and robust predicting models. Statistical methods like linear regression have been widely used in prediction domain. However, these linear models may not be able to achieve reliable prediction if the sequence is nonlinear or irregular. Moreover, the existence of noisy data which is a common phenomenon in scientific domain further reduces the prediction accuracy of these linear models. To overcome these shortcomings, AI methods, including ANN [64] and Hidden Markov Model [65] have been developed rapidly. However, many real-world applications like wind power prediction [66] are too complex to be modeled by the above single global model. In recent years, SVR [67] has been applied in nonlinear regression forecasting. It was proposed

[1] This chapter mainly extends the Journal paper: H. Zhu, Y. Tian, Y. Ren and J. Hu, "A Hybrid Model for Nonlinear Regression with Missing Data Using Quasi-Linear Kernel", *IEEJ Trans. on Electrical and Electronics Engineering*, Vol.15, No.12, pp.1791-1800, Dec 2020.

by the AT&T BELL Laboratories for classification, and then extended for the purpose of regression. But on its nonlinear expansion, choosing explicit kernel function for specific applications is challenging, which maps the sample from original space to another high-dimensional feature space [68].

Another inevitable problem becoming more popular is the missing data problem [4, 5], which has been a common phenomenon in prediction domain. Missing data is usually due to mechanical and human factors which includes unobserved samples and observation. Improper processing of the missing information will directly affect the accuracy of forecasting. The simplest method to solve missing data problem is deletion. However, it may discard a large amount of information in datasets. In particular, complete samples are too little to provide effective information especially when the proportion of missing values is large. Another approach is called imputation [19], which is a class of procedures that aims to replace the missing data with estimated values such as mean imputation and K-Nearest-Neighbors(KNN) imputation [69]. Moreover, clustering methods like K-POD [24] become more popular since similarities between samples can be found to estimate missing values. But it still has some disadvantages. Firstly, we have to set the number of clusters during K-means algorithm. Secondly, a simple and inexpensive fill-in step has been applied during update procedure, which may lead to inaccurate results. Besides, some advanced algorithms like EM algorithm [70] and multiple imputation [71] which is a statistical method which aims to allow for several representative imputation of the dataset have also been widely used for solving missing data problem.

In order to solve aforementioned problems, we propose a hybrid model consisting of two parts, to solve nonlinear regression problem under missing data scenario. As one part, a neural network is firstly proposed to solve missing data problem. In this way, an overcomplete winner-take-all DAE [72] is pre-trained as a multiple imputation tool to learn nonlinear relationships combined with a novel preprocessing method. Different from traditional autoencoders, DAEs allow corrupted versions as input data during training, where corruptions can be produced either by additive mechanisms or by missing data. In the context of DAEs, DAEs based imputation can naturally be combined with multiple imputation scenario, since DAEs draw several posterior predictive distributions through setting different random weights when initializing DAEs. Moreover,

the estimation of missing values cannot achieve the perfect 100% because of specific reasons in most real applications. In such case, the imperfect estimation result about missing data can be served as noisy data. Traditional regression models may be sensitive to noisy data due to overfitting problem. Therefore, as the other part, we build a gated linear network so as to realize a piecewise linear regression model. After reconstructing piecewise local linear model with interpolations, nonlinear regression problem can be transferred into linear-in-parameter problem, which can be finally solved by using SVR formulation [73, 74] in the renewed feature space.

However, training DAEs requires complete data at initialization, and we also need partition information during building gated linear network. Therefore, we extend a novel AP clustering algorithm [75] combined with two different updating methods during the preprocessing procedure named self-organized competitive net. By mean of this method, accurate teacher signals can be obtained during training phase and it can also provide parameters for further gated linear network.

Therefore, the general methodology is proposed for modeling and identifying the hybrid model. Firstly, an extended AP clustering algorithm with iterations is applied to obtain a self-organized competitive net. Secondly, a multiple imputation tool with top $p\%$ winner-take-all DAE is introduced to learn latent interactions among datasets so as to realize better predictions of missing values, in which rough estimates of missing values by using mean imputation and similarity method within the clusters are used as the initial values before training DAEs. Besides, multiple imputation can be adopted during training phase to provide several slightly different imputed values. Finally, a gated linear network is constructed to implement the piecewise linear regression model with interpolations in an exact same way as an SVR with a quasi-linear kernel composed using the cluster information obtained during the self.

Our main contributions in this chapter are as follows. Firstly, this chapter is the first to present a hybrid model which combines winner-take-all DAE and a gated linear network to solve forecasting problem with missing data. Secondly, to the best our knowledge, our studies on filling in missing data before training DAEs firstly use AP clustering algorithm for constructing the self-organized competitive net, which makes DAEs get more

accurate and effective information during training, and as to the regression phase, we can also make full use of cluster information efficiently to build a local linear prediction model.

The rest of the chapter is structured as follows: Section 2.2 formulates local linear model with missing data. Then, a novel AP clustering algorithm is introduced as a preprocessing method in Section 2.3. Section refsec3.4 presents an overcomplete winner-take-all DAE to realize better predictions of missing values. In section 2.5, a piecewise local linear regression is given, and followed by numerical experiments in Section 2.6. Finally, conclusions are summarized in Section 2.7.

2.2 Problem Formulation

Suppose a given dataset whose sample is $x \in R^d$ with missing values. The missing components are denoted by $\mathbf{N}_x \subseteq [d] = \{1, 2, ..., d\}$.

In consideration of nonlinear regression problem, we build a hybrid model consisting of an overcomplete DAE and a gated linear network. As shown in Fig.2.1(b), the DAE is defined by:

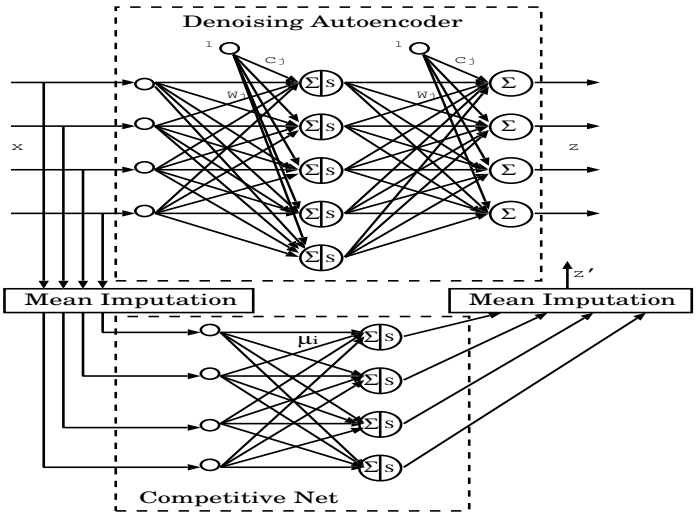
$$z = DAE(\mathbf{W}, c, x) \tag{2.1}$$

where $\{\mathbf{W}, c\}$ is parameter set, and the gated linear network is defined by:

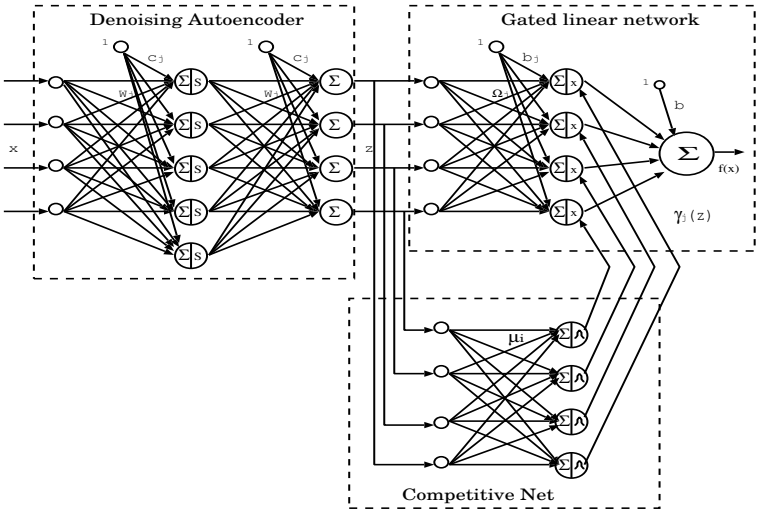
$$f(z) = \sum_{j=1}^{M} (\Omega_j^T z + b_j)\gamma_j(z) + b \tag{2.2}$$

where M is the number of linear node, and $\{\Omega_j, b_j, b\}$ is the parameter set, $\gamma_j(z)$ is the gate control signal generated by a self-organized competitive net.

The output of the DAE in Eq.(2.1), z, gives a prediction of x with the missing values estimated. The gated linear network in Eq.(3.2) performs the nonlinear regression as a piecewise linear regression model with interpolations, in which $\Omega_j^T z + b_j$ is local linear



(a) When training the denoising autoencoder



(b) After training the denoising autoencoder

FIGURE 2.1: The overall structure of the hybrid model

model and $\gamma_j(z)$ is a RBF interpolation function defined by:

$$\gamma_j(z) \quad = \quad \exp\left[-\frac{(\mathrm{z}-\mu_\mathrm{j})^2}{\lambda\sigma_\mathrm{j}^2}\right] \tag{2.3}$$

where μ_j and σ_j are the center and the width of the local clusters, and λ is an appropriate scale parameter.

In general, our essential problems are how to estimate the three sets of parameters $\{M, \mu_j, \sigma_j\}$, $\{\mathbf{W}, c\}$ and $\{\Omega_j,\ b_j,\ b\}$, by using the training data. We then propose a sophisticated modeling and identifying algorithm for the hybrid model, which consists of

three consecutive steps:

Step 1 By incorporating the mean imputation for filling missing data, AP clustering algorithm is extended to the case where there are missing data. With the extended AP clustering algorithm a self-organized competitive net is obtained to divide the dataset into M clusters. Then the centers and the sizes of clusters give the estimates of μ_j and σ_j. On the other hand, The mean imputation and the similarity method within clusters provides rough estimates of missing values.

Step 2 As shown in Fig.2.1(a), by using the rough estimates of missing values in the first step as the teacher signals, we train a winner-take-all DAE to obtain a prediction of x with missing values estimated.

Step 3 As shown in Fig.2.1(b), based on the output of DAE, z, a prediction of x with missing values estimated, the competitive net may be updated, if needed, to obtain the updated estimates of μ_i and σ_i. And then an SVR formulation is introduced to optimize the linear parameter set $\{\Omega_j, b_j, b\}$ in the reproduced feature space to provide final estimates.

2.3 An Extended Clustering for Partitioning

In this section, we will introduce the extended AP clustering algorithm aimed at obtaining both teacher signals and parameters $\{M,\mu_j,\sigma_j\}$ for constructing the gate constrol signal $\gamma_j(z)$.

2.3.1 AP Clustering Algorithm

An AP clustering Algorithm is a kind of fast clustering algorithm, which has several advantages compared with other traditional clustering methods. Firstly, AP clustering method does not need to specify parameters to decide the number of clusters. Secondly, all data points can be considered as potential centers simultaneously instead of creating

new clustering centers [76]. Thirdly, the results of executing multiple times are exactly the same, that is, there is no need to randomly select initial values.

Aimed at finding appropriate exemplars (clustering center), 'Responsibility' $R(i,k)$ and 'Availability' $A(i,k)$ are set as two evaluation criteria. $R(i,k)$ indicates the degree to which data point k is suitable as the clustering center of data point i, and $A(i,k)$ represents the suitability of data point i to select data point k as its exemplar. The larger $A(i,k)+R(i,k)$ is, the more likely data point k will be the final exemplar. In the iterative process, AP algorithm continuously updates $R(i,k)$ and $A(i,k)$ of each data point until it generates m accurate exemplars, and distributes the remaining data points to the corresponding clustering in the mean time.

In the first stage, we calculate an $n\times n$ similarity matrix S for n data points, update $R(i,k)$ of each point in the similarity matrix and calculate $A(i,k)$. In addition, $S(i,k)$ takes the negative value of the Euclidean distance between i and k, and we set $S(i,k)$ as the median of the entire matrix when $i=k$, which also affects the final number of clusters. Therefore, The main formulas in AP algorithm are:

$$R_{t+1}(i,k) = \begin{cases} S(i,k) - \max\limits_{j\neq k}\{A_t(i,j)+S(i,j)\} & i\neq k \\ S(i,k) - \max\limits_{j\neq k} S(i,j) & i=k \end{cases} \tag{2.4}$$

$$A_{t+1}(i,k) = \begin{cases} \min\{0, R_t(k,k) + \sum\limits_{j\neq i,k} \max\{R_t(j,k),0\}\} & i\neq k \\ \sum\limits_{j\neq k} \max\{0, R_t(j,k)\} & i=k \end{cases} \tag{2.5}$$

where $j\in\{1,2,...,n\}$. In Eq.(2.4), we only consider which point k is most likely to be the exemplar of point i, but neglect whether k could be seen as the clustering center of other data points, which may cause final number of clustering centers to be larger than the actual. As a result, Eq.(2.5) is proposed as the cumulative proof of selecting k as its clustering center.

Then in each iterative step, $R_{t+1}(i,k)$ and $A_{t+1}(i,k)$ are updated with the one in last iteration. The damping factor $\hat{\lambda}$ is used to eliminate oscillation and adjust convergence

speed, where $\hat{\lambda} \in [0,1]$ and default $\hat{\lambda} = 0.5$. The updating formulas are:

$$R_{t+1}(i,k) = \hat{\lambda} \times R_t(i,k) + (1-\hat{\lambda}) \times R_{t+1}(i,k) \tag{2.6}$$

$$A_{t+1}(i,k) = \hat{\lambda} \times A_t(i,k) + (1-\hat{\lambda}) \times A_{t+1}(i,k) \tag{2.7}$$

Finally, we calculate $A(i,k) + R(i,k)$ to test the decision of selecting the clustering center. If the clustering center remains constant after several iterations, or the maximum number of iterations is reached, then we terminate the overall procedure.

2.3.2 Fill-in Methods for Each Updating Iteration

Next, fill-in step should be discussed based on AP clustering algorithm to generate new dataset. The filling in is done per iteration until convergence.

Firstly, we use its conditional mean to replace the missing data point[77]. That is:

$$x_t^0 = \begin{cases} E[\mathbf{X}_t | x_{obs}] & \text{if } t \in \mathbf{N}_x \\ x_t & \text{otherwise} \end{cases} \tag{2.8}$$

where x_t^0 is the initial estimating version of x, and $\mathbf{X}_t$ is the tth co-variate. Then AP clustering algorithm is used to obtain initial cluster assignments and record its assigned cluster C. In the following, two methods can be used to update unobserved portion for each iteration. One is a kind of computationally inexpensive method which aims at determining the range of plausible values roughly and effectively. Based on it, missing values can be updated through conditional mean method based on clustering result. Thus:

$$x_t^n = \begin{cases} E[\mathbf{X}_{Ct} | x_{Cobs}] & \text{if } t \in \mathbf{N}_x \cup x \in C \\ x_t & \text{otherwise} \end{cases} \tag{2.9}$$

where x_t^n is the estimated value of nth iteration. Another updated method is to calculate the weight between the observed data in the same cluster. The detailed algorithm is shown as follows:

(1) Let $\dot{x}$ denotes another sample vector in the same cluster. Calculate the similarity between x and $\dot{x}$. This paper defines the similarity metric between data points in input space as:

$$s = \frac{1}{distance(x, \dot{x})} \tag{2.10}$$

where Euclidean distance is assigned to calculate the distance between samples.

(2) Calculate the weight between x and other samples in the same cluster in sequence. For instance, the weight of missing data sample x corresponding to $\dot{x}$ can be defined as:

$$w_j = \frac{s_j}{\sum_{j'=1}^{n} s_{j'}} \tag{2.11}$$

(3) Finally, we perform a weighted estimation of the missing values, which is calculated as:

$$x_t^n = \begin{cases} \sum_{j=1}^{n} w_j \mathbf{X}_{Ct} & \text{if } t \in \mathbf{N}_x \cup x \in C \\ x_t & \text{otherwise} \end{cases} \tag{2.12}$$

in which $\mathbf{X}_{Ct}$ means all the other samples in the same cluster and w_j is its corresponding weight.

More concretely, experimental results suggest that repeating 3 or 4 times of conditional mean method and then using the second weighted estimating method until convergence can be more efficient and delivers statistically higher performance. We summarize the whole procedure in Algorithm 1.

Therefore, the information about the number of clusters M, μ_i and σ_i of ith cluster can be acquired in the last iteration simultaneously, in which μ_i and σ_i are set as the center and radius of the ith data cluster. Moreover, rough estimates of missing values $z' = x_t^n$ can also be obtained simultaneously.

Algorithm 1 AP clustering algorithm for missing data during preprocessing procedure
Input: sample x with missing values
Output: rough estimates of missing values and cluster information about μ, σ and M
For each $i \in [1,3]$ **do**

1. Fill in unobserved entries through conditional mean method;
2. Update clustering result with AP clustering algorithm;

End For
Repeat

3. Fill in unobserved entries through weighted estimating method;
4. Update clustering result with AP clustering algorithm;

until Convergence

2.4 Denoising Autoencoders for Missing Data

2.4.1 Denoising Autoencoders

A traditional autoencoder is an unsupervised learning algorithm which is mainly used for feature extraction or dimensional reduction. For encoder phase, an autoencoder takes an input $x \in [0,1]^d$ and then maps it into a different representation $h \in [0,1]^{d'}$, where d' means another dimensional subspace. Then h can be mapped back into the decoder phase. Both encoders and decoders can be regarded as artificial neural networks. DAEs, which are natural unsupervised extensions to traditional autoencoders, are forced to map corrupted input data caused by missing mechanisms or distributional additive noise into hidden layers to learn latent features. Therefore, missing data problem can be seen as a special case that makes DAEs an effective candidate to recover missing patterns. As an example, we consider a 5-layer symmetrical autoencoder described as:

$$h(j) = a\left(\mathbf{W}_2^T\left(a(\mathbf{W}_1^T x + c_1)\right) + c_2\right),\ j = 1,2,... \tag{2.13}$$

$$z(i) = a\left(\mathbf{W}_1\left(a(\mathbf{W}_2 h + c_3)\right) + c_4\right),\ i = 1,2,... \tag{2.14}$$

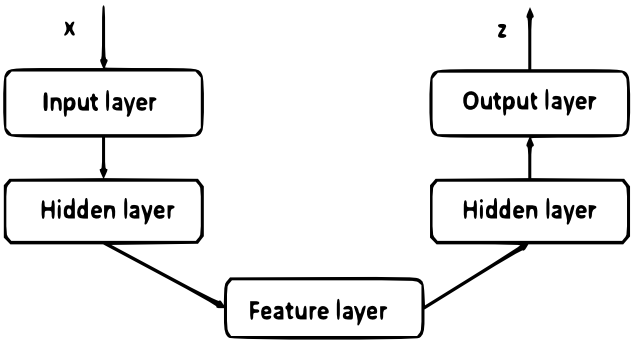


FIGURE 2.2: The basic structure of denoising autoencoders

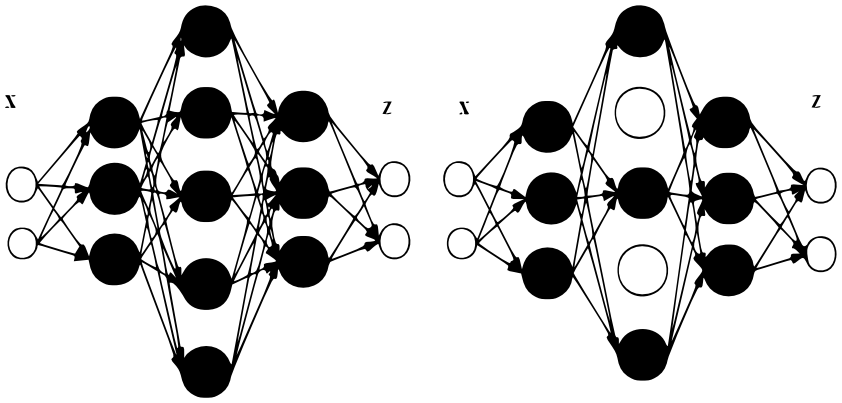


FIGURE 2.3: Comparison with basic DAE and winner-take-all DAE

where a is ReLU activation function, $h = [h(1), h(2), ...]$ is the feature representation, $z = [z(1), z(2), ...]$ is a prediction of x based on a reconstruction from the feature, the parameter set is $\{W_1, W_2, c_1, c_2, c_3, c_4\}$.

As shown in Fig.2.2 and Fig.2.3, in order to have an accurate reconstruction of missing data based on latent associations, it is built as an overcomplete DAE. That is, the number of nodes is increasing from the previous layer by a factor of $(1+\varphi)$ in encoder while decreasing in decoder, where φ=0.5 to 1.0 in this paper. On the other hand, to overcome overfitting problem, we apply a top $p\%$ winner-take-all [78] measure on the feature layer. Winner-take-all method accepts only top $p\%$ hidden layer units activated for each sample, while the others are set as 0. Let $\tilde{h}$ denote the input and consider ReLU activation function, the output of feature layer is

$$\begin{aligned} h(j) = a(\mathbf{W}_2^T \tilde{h} + c_2) &= \max\{0, \mathbf{W}_2^T \tilde{h} + c_2\} \\ &= (\mathbf{W}_2^T \tilde{h} + c_2) H(\mathbf{W}_2^T \tilde{h} + c_2) \end{aligned} \tag{2.15}$$

where $H(\cdot)$ is a step function. Then we rank all of values in hidden units on the feature layer and only top $p\%$ units are chosen to keep them active. Therefore, only a few units can be updated, which protects the model from overfitting. By defining a set $\Gamma = supp_p\{a(\mathbf{W}_2^T\tilde{h} + c_2)\}$ containing hidden units with top $p\%$ activation values, the representation $h(j)$ can finally be formulated as:

$$h(j) = \begin{cases} (\mathbf{W}_2^T\tilde{h} + c_2)H(\mathbf{W}_2^T\tilde{h} + c_2) & j \in \Gamma \\ 0 & j \notin \Gamma \end{cases} \tag{2.16}$$

where $j = 1, 2, ...n_l$ (the number of node in feature layer). p and φ are hyper-parameters. How to choose suitable p and φ affects the accuracy of calculating missing data. Either a larger p or φ may increase the risk of overfitting. However, smaller φ may also prevent the DAE from extracting enough information. $\varphi \in [0.5\ 1.0]$ and $p = 50$ in this paper.

The overcomplete DAE is trained by minimizing a loss function defined by

$$loss = \frac{1}{2}\sum_{i}^{N}(z_i' - z_i)^2 + L_2 \tag{2.17}$$

where z' is the teacher signals obtained using the extended AP clustering, L_2 is the L_2 regularization term.

2.4.2 Multiple Imputation for Missing Values

In order to provide diverse information for further inference, we apply multiple imputation to generate new replaced values for unobserved data, and subsequent regression can combine all imputed versions to acquire more accurate results. In short, multiple imputation takes the uncertainty of missing values into consider. Therefore, with the randomness of initial weights at each run, we sweep the complete model repetitively to generate P new datasets. The final result of DAEs can be defined through calculating an average respectively, that is:

$$z = \frac{1}{P}\sum_{i=1}^{P} z_i \tag{2.18}$$

2.5 SVR with Quasi-linear Kernel

In this section, we propose a SVR formulation to further optimize the parameter set of $\{\Omega_j, b_j, b\}$, which combines the advantages of SVR and piece-wise linear approximation.

After importing two vectors $\Phi(z)$ and Θ defined as:

$$\Phi(z) = [\gamma_1(z), z^T\gamma_1(z), ..., \gamma_M(z), z^T\gamma_M(z)] \tag{2.19}$$

$$\Theta = [b_1, \Omega_1^T, ..., b_M, \Omega_M^T]^T \tag{2.20}$$

Therefore, Eq.(2.2) can be expressed as a linear-in-parameter way as:

$$f(z) = \Theta^T\Phi(z) + b \tag{2.21}$$

As a result, the nonlinear regression problem is reduced to a linear regression model. $\Phi(z)$ is the regression vector and Θ is called linear parameter vector. In the following, we will focus on how to estimate linear parameters using SVR formulation. Based on the structural risk minimization principle as:

$$\begin{aligned} &\min_{\Theta,b,\xi_i,\xi_i^*} \frac{1}{2}\Theta^T\Theta + C\sum_{i=1}^{N}(\xi_i + \xi_i^*) \\ &\text{s.t.} \begin{cases} \Theta^T\Phi(z) + b - y_i \le \epsilon + \xi_i^* \\ y_i - \Theta^T\Phi(z) - b \le \epsilon + \xi_i \\ \xi_i, \xi_i^* \ge 0, \quad t = 1, 2, ..., N \end{cases} \end{aligned} \tag{2.22}$$

where y_i denoted the ideal output of z_i, C is a non-negative weight to determine the penalization of prediction errors, N is the number of observations, and ξ_t, ξ_t^* are slack variables. By introducing Lagrange multipliers $\mu \ge 0$, $\mu* \ge 0$, $\alpha \ge 0$, $\alpha* \ge 0$, we can

construct the Lagrange function as:

$$\begin{aligned} L(\Theta,\xi_t,\xi_t*,\alpha,\alpha*,\mu,\mu*) &= \frac{1}{2}\Theta^T\Theta + C\sum_{i=1}^{N}(\xi_i+\xi_i^*) \\ &+ \sum_{i=1}^{N}\alpha_t(f_z(z)-\Phi(z)\Theta-\epsilon-\xi_t) + \sum_{i=1}^{N}\alpha_t(-f_z(z) \\ &+ \Phi(z)\Theta-\epsilon-\xi_i^*) - \sum_{i=1}^{N}(\mu\xi_i+\mu^*\xi_i^*) \end{aligned} \tag{2.23}$$

Then it can be solved through getting the saddle point:

$$\begin{aligned} \frac{\partial L}{\partial\Theta} = 0 \rightarrow \Theta &= \sum_{i=1}^{N}(\alpha-\alpha^*)\Phi(z_i) \\ \frac{\partial L}{\partial\xi} = 0 \rightarrow C &= \alpha+\mu \\ \frac{\partial L}{\partial\xi^*} = 0 \rightarrow C &= \alpha^*+\mu^* \end{aligned} \tag{2.24}$$

After converting the Lagrange function into its dual problem, we can get:

$$\begin{aligned} \max\ W(\alpha,\alpha^*) = &-\frac{1}{2}\sum_{i,j=1}^{N}(\alpha_i-\alpha_i^*)(\alpha_j-\alpha_j^*)K(z_i,z_j) \\ &+ \sum_{i=1}^{N}(\alpha_i-\alpha_i^*)f(z) - \epsilon\sum_{i=1}^{N}(\alpha_i+\alpha_i^*) \\ \text{s.t.}\ \ &\sum_{i=1}^{N}(\alpha_i-\alpha_i^*) = 0. \quad \alpha,\alpha^*\in[0,C]. \end{aligned} \tag{2.25}$$

where $K(z_i,z_j)$ is a data-dependent composed kernel called quasi-linear kernel, defined as:

$$\begin{aligned} K(z_i,z_j) &= \Phi^T(z_i)\Phi(z_k) \\ &= (1+z_iz_j)\sum_{k=1}^{M}\gamma_k(z_i)\gamma_k(z_j) \end{aligned} \tag{2.26}$$

From the above, with the Lagrange multipliers α_i and α_i^* obtained, the regression model can finally be represented as:

$$f(z) = \sum_{i=1}^{N} (\alpha_i - \alpha_i^*) K(z, z_i) + b \tag{2.27}$$

Overall, by implementing the aforesaid procedures, we are able to solve the nonlinear regression problem with missing data. By comparison with the early studies, two major contributions have been made in our work. First, each part in our hybrid model is closely related. For instance, the result of AP clustering algorithm cannot only be used to provide teacher signals for DAEs, but also provide clustering information to construct a competitive net to generate gate control signal for the gated linear network. Second, a gated linear network is designed to build a piecewise linear regression model with interpolations, which is more suitable to solve the regression problem under missing data scenario because of its robustness.

2.6 Experiment Results and Discussions

2.6.1 Datasets

Five datasets without missing values are used in order to test the effectiveness of proposed method. The details of datasets are shown in Table 2.1. The last column of Table 2.1 shows the network size of 5-layer DAE. All datasets can be downloaded from UCI machine learning repository [79] and Tianchi crowd intelligence platform [80]. When dealing with missing values, only input samples are considered while outputs are left out.

Before processing missing data problem, it is necessary to understand the mechanism of missing data, which is divided into three main categories [81]:

- MCAR [9], which means the absence of missing data in the tth co-variate ($\mathbf{X}_t$) is independent of the value itself or any other attributes.

TABLE 2.1: Specification of the five tested regression datasets

Datasets	Attribute	Training data	Testing data	Net size of 5-layer DAE
Steam	38	2888	1444	38×57×86×57×38
Stocks	9	633	317	9×18×36×18×9
Tecator	122	160	80	122×208×353×208×122
Bank1	8	2999	1500	8×16×32×16×8
Bank2	16	2999	1500	16×24×36×24×16

- MAR, it represents that the probability of $\mathbf{X}_t$ is missing is only related to other observed attributes without regard to its own value. For instance, if men are more likely to tell their weights than women, The absence of weight values can be seen as MAR.

- MNAR, which is a mechanism that missingness generation depends on the value of unobserved attribute $\mathbf{X}_t$ itself. The only way to solve MNAR is to build the missingness model.

since our research mainly focuses on regression with missing data rather than analyzes missing data for some special reasons, so only MCAR mechanism is considered in this paper. Therefore, the real case scenario should be simulated by discarding some elements of the data randomly with a fixed probability during preprocessing procedure.Two-thirds of the whole datasets are chosen as the training part, and the remaining third is set as testing set. Then repeat this Monte-Carlo [82] preprocessing 10 times and calculate the average of Root Mean Square Error (RMSE) as the criterion for comparison. For each Monte-Carlo split, we also repeat the whole model 10 times and impute the mean standard deviation. The formula of RMSE is defined as:

$$RMSE = \sqrt{\frac{1}{L}\sum_{i=1}^{L}(m_i - n_i)^2} \tag{2.28}$$

where m_i and n_i are the predicted and observed values respectively, L is the total amount of the testing data. The value of RMSE closed to 0 indicates that the model is perfect.

2.6.2 Experiments and Results

To emphasize the effectiveness of DAEs, AP clustering algorithm combined with SVR with quasi-linear kernel (AP-QSVR) is applied to do the comparison. For further comparison, SVR with linear kernel (LinearSVR) and SVR with RBF kernel (RBF-SVR) are then applied to test respectively, and DAE combined with SVR with RBF kernel (DAE-SVR) is also used to verify the performance of AP clustering algorithm when dealing with missing data. The results of each datasets are presented for different percentages of missing data as 10%, 20%, 30%, 40%, 50%, 60%, and the same experiment procedures are done to evaluate the methods. Moreover, other two prediction methods with missing values including KSC clustering with MVI kernel method (KSC-MVI) devised in Ref.[83] and Gaussian mixture model (GMM) with extreme learning machine (ELM) (GMM-ELM) proposed in Ref.[84] are also utilized to compare with our method. Besides, this paper also tests GMM algorithm combined with SVR with quasi-linear kernel (GMM-QSVR) to verify the effectiveness of our proposed model, and the clustering result of GMM is then used to do further prediction. Since there may be missing values in the testing data, we use conditional mean method to fill in these missing values during the comparative experiments. Moreover, $P = 10$ is used for the multiple imputation scenario because of new datasets $\{\mathbf{z}_i\}_{i=1}^{P}$ generated from DAEs, and λ in Eq.(2.3) is estimated in experiments.

Take the steam data with 50% missing values for example, there are 4332 samples, 38 attributes and 1 output originally in the dataset. Firstly, $(4332\times38)\times50\% \approx 139308$ data points are deleted as missing values. Then for each Monte-Carlo split, 2888 samples are chosen randomly as training set and the other 1444 samples are set as testing part. Fig.2.4 illustrates the comparison result of steam data. x-axis means the percentage of the missing data in the training part from 10%-60%, and y-axis represents RMSE results.

Table.2.2 and Fig.2.4-2.8 are used to present the comparison results. For easy comparison, we highlight the first-rank model with boldface. Based on it, we can draw the following conclusion. Firstly, we make a comparison between our proposed method and SVR with conventional kernels. When the number of missing data is small, our

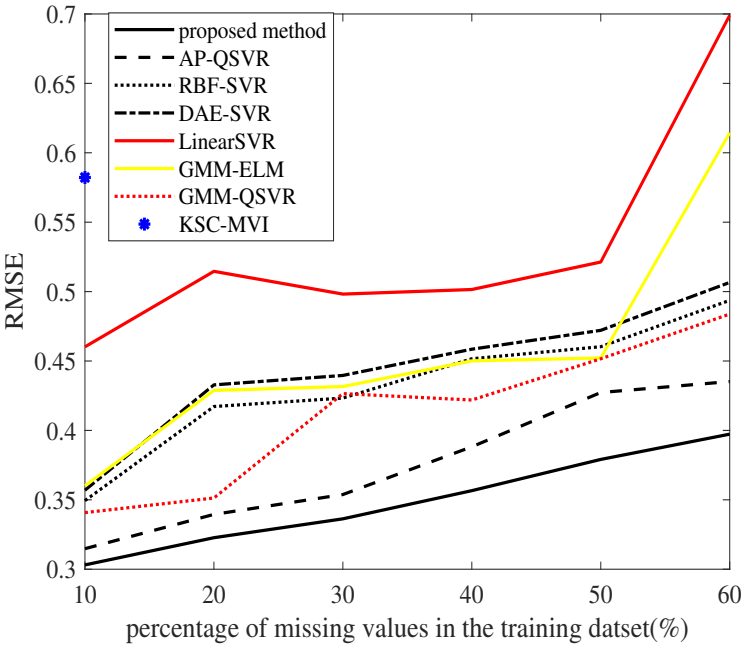


FIGURE 2.4: RMSE for steam data

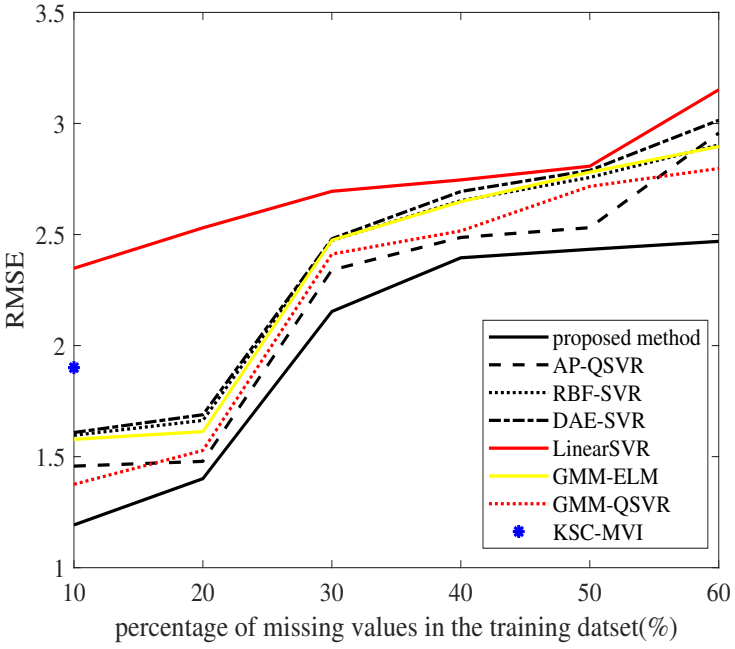


FIGURE 2.5: RMSE for stock data

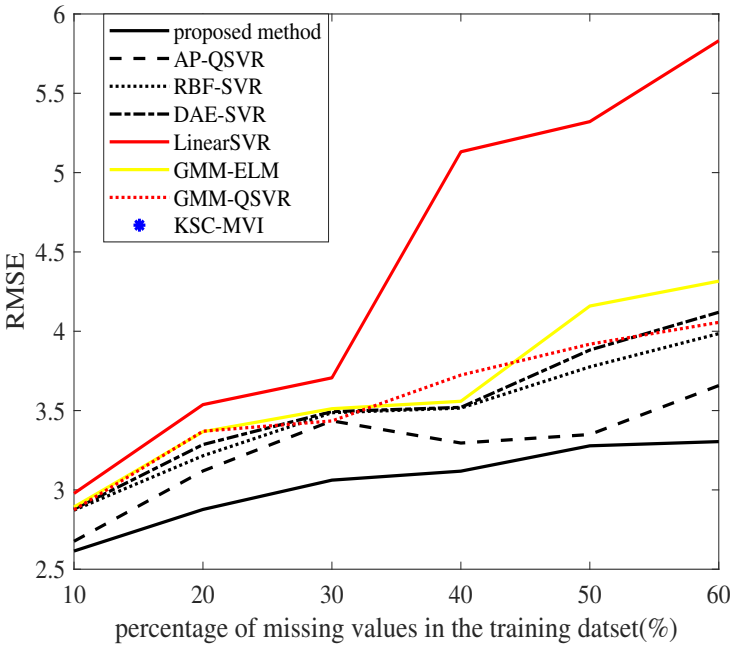


FIGURE 2.6: RMSE for tecator data

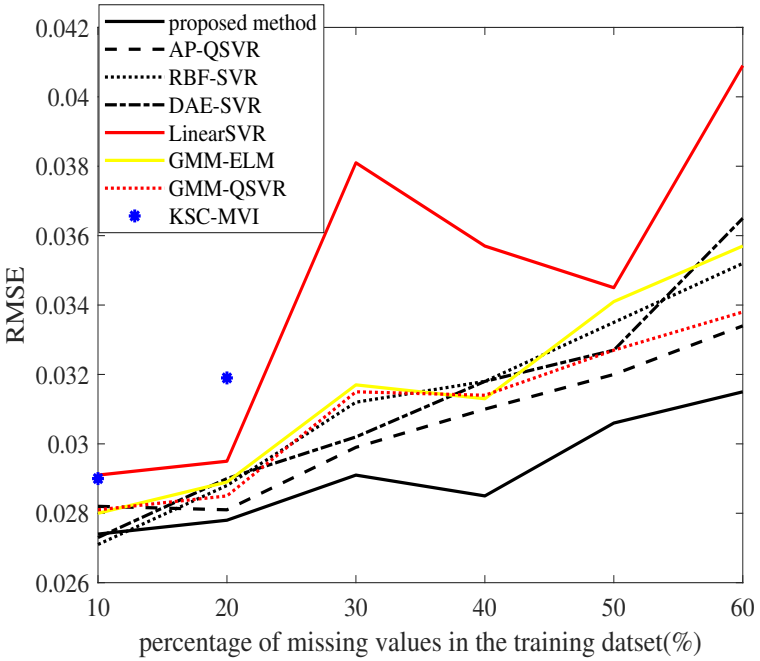


FIGURE 2.7: RMSE for bank1 data

TABLE 2.2: Prediction results for all five tested regression datasets

Datasets	Models	Missing					
		10%	20%	30%	40%	50%	60%
Steam	proposed method	**0.3031±0.0003**	**0.3227±0.0004**	**0.3363±0.0004**	**0.3566±0.0004**	**0.3791±0.0006**	**0.3973±0.0009**
	AP-QSVR	0.3148	0.3395	0.3538	0.3882	0.4273	0.4352
	RBF-SVR	0.3494±0.0003	0.4172±0.0004	0.4233±0.0006	0.4516±0.0006	0.4603±0.0009	0.4937±0.0011
	DAE-SVR	0.3569±0.0003	0.4328±0.0005	0.4396±0.0007	0.4585±0.0007	0.4721±0.0010	0.5066±0.0012
	LinearSVR	0.4602±0.0005	0.5146±0.0005	0.4982±0.0006	0.5015±0.0007	0.5213±0.0011	0.6991±0.0014
	GMM-ELM	0.3601	0.4288	0.4316	0.4501	0.4552	0.6144
	GMM-QSVR	0.3408	0.3513	0.4265	0.4219	0.4516	0.4838
	KSC-MVI	0.5822	-	-	-	-	-
Stocks	proposed method	**1.1924±0.0011**	**1.4008±0.0011**	**2.1534±0.0013**	**2.3952±0.0014**	**2.4335±0.0016**	**2.4692±0.0019**
	AP-QSVR	1.4573	1.4787	2.3403	2.4868	2.5309	2.9570
	RBF-SVR	1.5961±0.0012	1.6631±0.0013	2.4735±0.0016	2.6525±0.0017	2.7568±0.0020	2.9033±0.0023
	DAE-SVR	1.6085±0.0012	1.6887±0.0014	2.4809±0.0017	2.6938±0.0018	2.7885±0.0021	3.0146±0.0024
	LinearSVR	2.3480±0.0013	2.5301±0.0013	2.6945±0.0017	2.7459±0.0020	2.8076±0.0023	3.1518±0.0030
	GMM-ELM	1.5779	1.6126	2.4744	2.6478	2.7809	2.8962
	GMM-QSVR	1.3754	1.5280	2.4119	2.5163	2.7167	2.7973
	KSC-MVI	1.9007	-	-	-	-	
Tecator	proposed method	**2.6146±0.0027**	**2.8767±0.0027**	**3.0615±0.0029**	**3.1182±0.0030**	**3.2779±0.0033**	**3.3044±0.0036**
	AP-QSVR	2.6761	3.1194	3.4352	3.2954	3.3491	3.6577
	RBF-SVR	2.8721±0.0029	3.2152±0.0030	3.4867±0.0033	3.5158±0.0034	3.7755±0.0037	3.9851±0.0042
	DAE-SVR	2.8785±0.0029	3.2857±0.0030	3.4936±0.0034	3.5202±0.0036	3.8817±0.0039	4.1198±0.0044
	LinearSVR	2.9778±0.0030	3.5375±0.0033	3.7064±0.0037	5.1313±0.0040	5.3214±0.0042	5.8314±0.0046
	GMM-ELM	2.8922	3.3667	3.5118	3.5587	4.1590	4.3159
	GMM-QSVR	2.8744	3.3700	3.4347	3.7238	3.9187	4.0561
	KSC-MVI	-	-	-	-	-	-
Bank1	proposed method	0.0274±0.0001	**0.0278±0.0001**	**0.0291±0.0003**	**0.0285±0.0002**	**0.0306±0.0003**	**0.0315±0.0003**
	AP-QSVR	0.0282	0.0281	0.0299	0.0310	0.0320	0.0334
	RBF-SVR	**0.0271±0.0001**	0.0288±0.0001	0.0312±0.0003	0.0318±0.0003	0.0335±0.0004	0.0352±0.0004
	DAE-SVR	0.0273±0.0001	0.0290±0.0002	0.0302±0.0004	0.0318±0.0003	0.0327±0.0004	0.0365±0.0005
	LinearSVR	0.0291±0.0001	0.0295±0.0002	0.0381±0.0004	0.0357±0.0003	0.0345±0.0005	0.0409±0.0007
	GMM-ELM	0.0280	0.0289	0.0317	0.0313	0.0341	0.0357
	GMM-QSVR	0.0281	0.0285	0.0315	0.0314	0.0327	0.0338
	KSC-MVI	0.0290	0.0319	-	-	-	
Bank2	proposed method	**0.0841±0.0001**	**0.0847±0.0001**	**0.0856±0.0003**	**0.0863±0.0003**	**0.0865±0.0004**	**0.0876±0.0005**
	AP-QSVR	0.0848	0.0853	0.0864	0.0869	0.0884	0.0961
	RBF-SVR	0.0864±0.0001	0.0863±0.0002	0.0924±0.0004	0.1052±0.0004	0.1168±0.0006	0.1347±0.0006
	DAE-SVR	0.0865±0.0001	0.0891±0.0001	0.0993±0.0005	0.1167±0.0005	0.1275±0.0006	0.1431±0.0007
	LinearSVR	0.0895±0.0003	0.0892±0.0004	0.1038±0.0006	0.1322±0.0006	0.1495±0.0008	0.1883±0.0008
	GMM-ELM	0.0862	0.0866	0.0991	0.1147	0.1241	0.1437
	GMM-QSVR	0.0858	0.0861	0.0892	0.0907	0.0932	0.1204
	KSC-MVI	0.1002	-	-	-	-	-

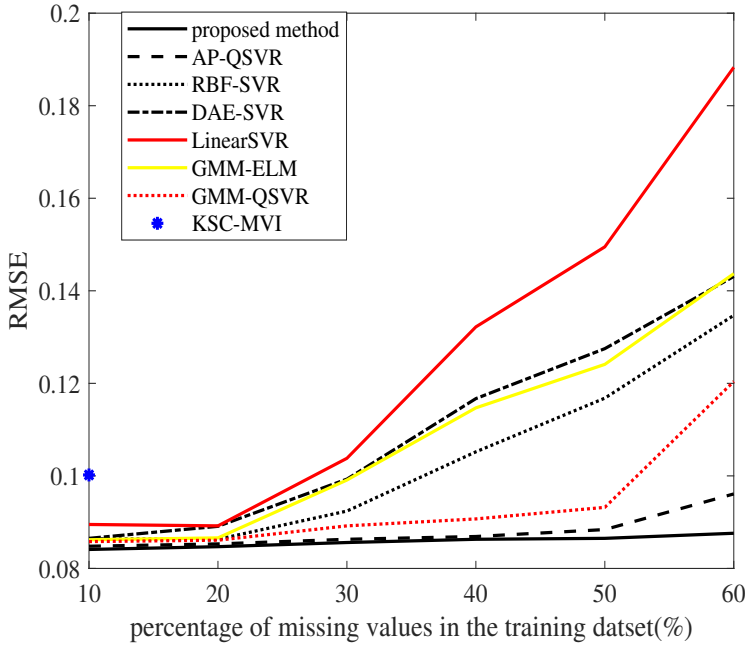


FIGURE 2.8: RMSE for bank2 data

proposed method has led to greater performance in most simulations other than bank1 data. We can see that in Fig.2.7, our proposed method cannot achieve the best result when the percentage of missing data is 10%. However, RBF-SVR method always gives complex parameters setting with case dependent. As the ratio of missing data increases, the advantage of our method becomes more obvious because of smooth rise of RMSE results, our method remains on a stable status even after adding 60% of the missing values. Secondly, it can be seen that deleting DAEs from the model reduces performance in all cases, except that it is inferior for tecator data with 20% of the missing data, which proves that DAEs can better leverage latent information. Thirdly, we also compare the performance of RBF-SVR and DAE-SVR to demonstrate one of the advantages of AP clustering algorithm, which is to provide the more accurate teacher signal for DAEs. Finally, our proposed method is then compared with two state-of-art algorithms. For KSC-MVI method, since it generates a set of constraints only based on the regular observed features and there are not enough samples left with the increasing amount of missing value, so it is only effective when the amount of missing data is small. Especially for tecator data, we can not use KSC-MVI method to do prediction due to its large attributes. Compared with GMM-ELM method, our proposed method has also achieved encouraging results in most cases, which proves that our algorithm is simple

but effective. Furthermore, we more concern the effectiveness of the proposed DAE, so GMM-QSVR method is also used to make a comparison. As to GMM-QSVR method, our proposed method has also achieved greater performance. Nevertheless, it is hard to estimate reliable parameters to fit GMM in high-dimensional data and execution time is also longer. As a whole, our proposed method have the lowest RMSE in most case which proves that our proposed model is much more robust and reliable.

2.6.3 Discussion

We in this paper introduce a novel hybrid model to solve regression problem with missing data. Experimental results prove that each component of the model is indispensable. Firstly, comparative experiments between the proposed method and AP-QSVR check the effectiveness of DAEs. Moreover, we also test the influence of layers of DAEs and show its effect when dealing with steam data as an example in Table.2.3.

TABLE 2.3: RMSE comparison across differen number of layers

Number of layers	Missing					
	10%	20%	30%	40%	50%	60%
3	0.3037	0.3235	0.3378	0.3592	0.3871	0.3984
5	**0.3031**	**0.3227**	**0.3363**	**0.3566**	**0.3791**	**0.3973**
7	0.3129	**0.3227**	0.3373	0.3569	0.3845	0.3992

Secondly, this local linear regression model inherits the advantages of SVR, and for quasi-linear kernel, it is physically meaningful and can be regarded as a composite and local linear kernel with interpolations, and this piecewise linear regression model is implemented by the cluster information of AP clustering algorithm. Experimental results among our proposed method and other traditional kernels have demonstrated the stability of quasi-linear kernel in the presence of noisy problem. Moreover, we implement the comparison between RBF-SVR and DAE-SVR to reveal that AP clustering algorithm is more suitable for combing with DAEs to solve missing data problem.

2.7 Conclusions

In this chapter, we have proposed an effective modeling method to realize the prediction task with missing data. Briefly speaking, AP clustering method with iterations is firstly presented as a preprocessing method aimed at providing teacher signals and cluster information to construct a competitive net. Then, we present a multiple imputation method based on winner-take-all DAEs. Finally, an advanced modification named SVR with quasi-linear kernel is used to design a gated linear network based on each partition.

Experiments of five datasets have shown the effectiveness and robustness of our proposed method. It has better performance compared with other advanced methods, especially when the percentage of missing data is very large.

Chapter 3

A Winner-Take-All Autoencoder Based Pieceswise Linear Model for Nonlinear Regression with Missing Data

3.1 Introduction

[1] As mentioned in Chapter 2, in most real applications, it is hard to find the reason why some data are missing. Therefore, it is vital to find a method that is suitable for all three missing mechanisms. In our previous work, a category of quasi-linear ARX models, which includes the quasi-linear kernel, was first applied to the nonlinear system identification [73], and then the multi-local linear model with interpolations has been proposed to solve nonlinear regression problems [74]. In Chapter 2, we proposed a hybrid model to solve the nonlinear regression problem under the missing data scenario, which consists of three parts: an extended AP clustering algorithm for partitioning, a denoising autoencoder for estimating the missing value, and a gated linear network implementing a multi-local linear model with interpolation. However, the performance

[1] This chapter mainly extends the Journal paper: H.Zhu,Y. Ren, Y. Tian and J. Hu, "A Winner-Take-All Autoencoder Based Piecewise Linear Model for Nonlinear Regression with Missing Data", *IEEJ Trans. on Electrical and Electronics Engineering*, Vol.16, No.12, pp.1618-1627, Dec 2021.

of the hybrid model depends heavily on the effectiveness of AP clustering for detecting local linear partitions that may be sensitive to the data distribution. In this chapter, by increasing the role of the denoising autoencoder we improve the hybrid model to consist of only two parts: an overcomplete WTA autoencoder [78] and a gated linear network [85]. The overcomplete WTA autoencoder is an SDAE designed to play two roles: 1) to estimate the missing values as a multiple imputation tool [71, 86]; 2) to realize a sophisticated partitioning by generating a broad set of binary gate control sequences using the feature layer of SDAEs [61]. When training the SDAE, an iterative algorithm is developed to improve the performance of the SDAE, which is, we firstly use the mean imputation method to acquire rough estimates of missing values, and then clustering information derived from the feature layer of SDAEs provides more accurate teacher signals for further training. On the other hand, by using the binary gate control sequences, the gated linear network implements a flexible piecewise linear model for the nonlinear regression. Moreover, by composing a quasi-linear kernel based on the gate control sequences, the piecewise linear model is identified in the same way as a SVR with the quasi-linear kernel.

The proposed hybrid model is applied to six real-world datasets with a wide range of missing data. Experimental results show that our proposed hybrid model has a better performance than state-of-the-art algorithms.

The rest of the chapter is organized as follows: Section 3.2 outlines the overall model, and then we introduce SDAEs for solving missing data and generating the gate mechanism in Section 3.3. The whole regression procedure is laid out in Section 3.4. Finally, experimental results on real-world datasets with a wide range of missing data are provided in Section 3.5, and conclusions are summarized in Section 3.6.

3.2 Structure of the Hybrid Model

Consider a d-dimensional dataset $S=\{(\mathrm{x}_1,\mathrm{y}_1),...,(\mathrm{x}_d,\mathrm{y}_d)\}$ with missing values, where $x_i \in R^d$ is the i-th input feature vector and $y_i \in R$ is the target output of i-th sample.

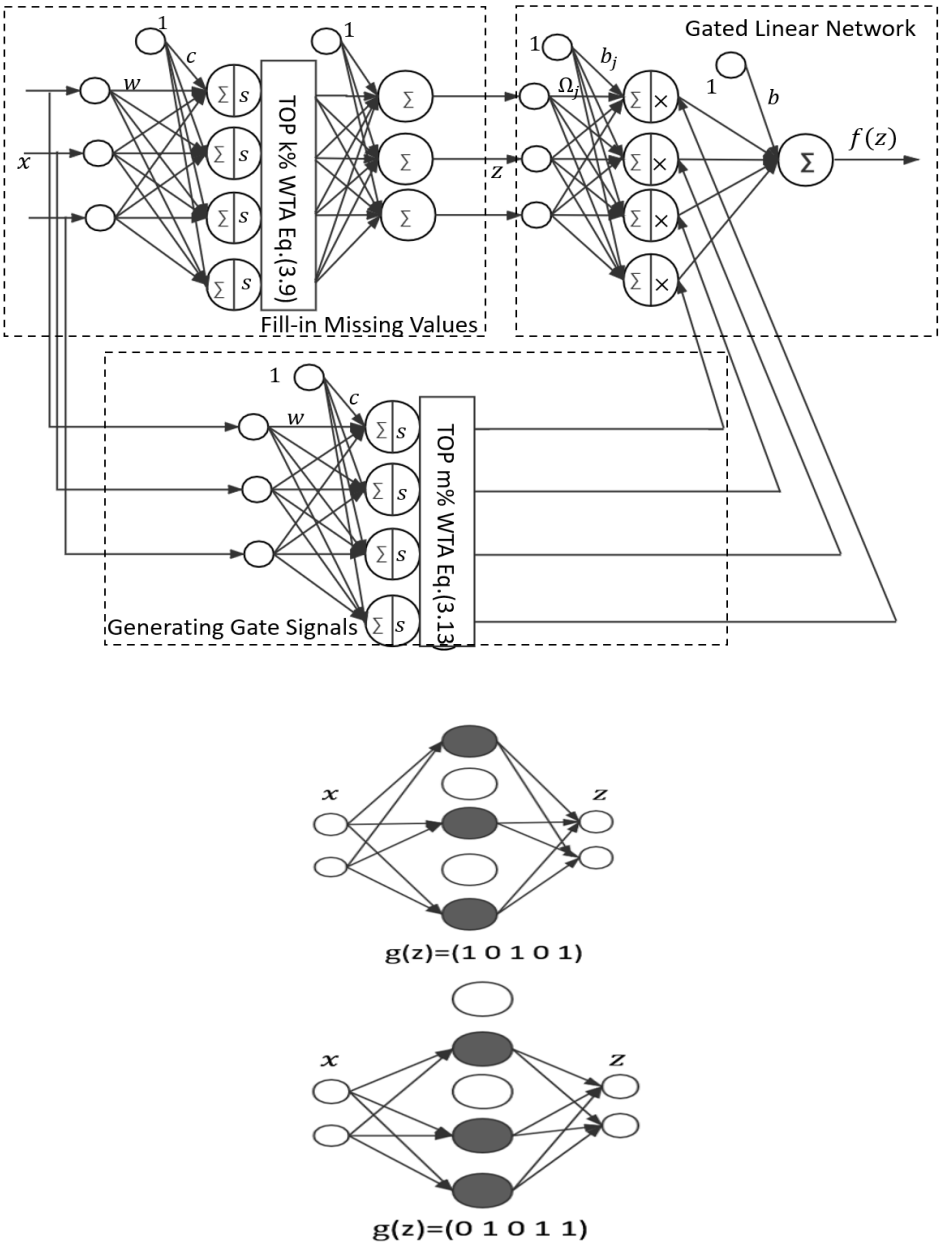


FIGURE 3.1: (a)The overall structure of the autoencoder based piecewise linear model; (b) An image of different gating signals by using different sequence of $g(z)$

When solving nonlinear regression problems with missing values, we consider an efficient hybrid model consisting of an overcomplete WTA SDAE and a gated linear network. As shown in Fig.3.1 (a), the WTA SDAE is defined by:

$$z = \text{SDAE}(\mathbf{w}, c, x) \tag{3.1}$$

where $\{\mathbf{w}, c\}$ is the parameter set, and then the gated linear network realizing a piecewise linear model, is defined by:

$$f(z) = \sum_{j=1}^{M} (\Omega_j^T z + b_j) g_j(z) + b \tag{3.2}$$

where M is the number of linear base models $\Omega_j^T z + b_j$, $\{\Omega_j, b_j, b\}$ is the parameter set, and $g_j(z) \in \{0, 1\}$ is a gate signal controlling whether the j-th base model works. A gating mechanism for generating gate control sequences $\mathbf{g}(z) = [g_1(z), g_2(z), ..., g_M(z)]^T$ is built by using the information of feature layers, defined by:

$$\mathbf{g}(z) = \text{SDAE}_{Flayer}(\mathbf{w}, c, x). \tag{3.3}$$

Note that as shown in Fig.3.1, the encoder parts of the WTA autoencoder used for filling in missing values and for generating gate signals are the same. Fig.3.1(b) shows an image of calculating z and $g(z)$ from an overcomplete WTA autoencoder and the details of SDAE$(\mathbf{w}, c, x)$ and SDAE$_{Flayer}(\mathbf{w}, c, x)$ will be discussed in Section 3.3.

In general, the overall proposed model consists of two parts as illustrated in Fig.3.1: an overcomplete WTA autoencoder for estimating missing values and generating gate control sequences, and a gated linear network for implementing a piecewise linear model for nonlinear regression. We will discuss the details of two parts in Sections 3.3 and 3.4 respectively.

3.3 Overcomplete Winner-take-all Autoencoder

The overcomplete WTA SDAE is designed to perform two functions: to estimate missing values and to generate gate control signals, as shown in Fig.3.1. Due to the missing values, accurate teacher signals are not available for training the SDAE. To solve the problem, we will introduce a sophisticated training algorithm consisting of two steps. In the first step, rough teacher signals are first generated by filling the missing values using the mean imputation method, then the weights of each encoder layer are pre-trained

through DAEs in a layer-by-layer way. In the second step, a fine-tuning of the whole autoencoder including the top $k\%$ WTA layer is performed. By applying a normal WTA strategy to the pre-trained encoder, we realize a clustering to better estimate the missing data by using clustered mean imputation method. In this way, the fine-tuning part is repeated by updating the teacher signals. Finally, all decoder parts are symmetrically scaled back to their original dimensions in our model.

The whole procedure is summarized in Algorithm 2.

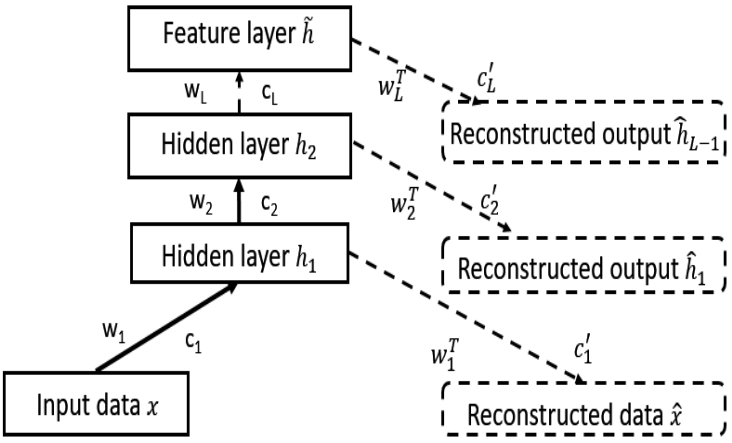


FIGURE 3.2: The network architecture of the full encoder part

Algorithm 2 Overcomplete WTA SDAEs for solving missing data and generating gate signals

Input: Dataset x with missing data

1. Pre-training the weights of all layers in the encoder, and mean imputation is used for generating rough teacher signals;

2. Fine-tuning of the full autoencoder;

Repeat

3. Updating teacher signals through clustering information provided by the feature layer;

4. Repeating the fine-tuning part of the full autoencoder;

Until Convergence

5. Generating gate signals through information on the feature layer;

Return Complete dataset z and gate singals $\mathbf{g}(z)$ for the gated linear network.

3.3.1 Pre-training Step of the Encoder

The pre-training step consists of stacking simple autoencoders using the DAE technique. Fig.3.2 shows the network architecture of the full encoder part. The encoder phase of the first DAE maps the corrupted input x caused by additive noise or discarding some data to the hidden representation h through any nonlinear functions, and then h is mirrored back to the reconstruction as the decoder. It is constructed as overcomplete, which means the number of nodes increases from the previous layer in the encoder. Therefore, the missing data problem being a typical case makes the DAE an ideal candidate for recovering missing variables. The transformation of the first DAE is shown as follows:

$$h_1 = a(\mathbf{w}_1 x + c_1) \tag{3.4}$$

$$\bar{x} = a\left(\mathbf{w}_1^T h_1 + c_1'\right) \tag{3.5}$$

where a represents rectified linear (ReLU) activation function in our model, $\{w_1, c_1, c_1'\}$ is the parameter set. To pre-train the first DAE, we minimize a loss function defined by:

$$E_1 = \frac{1}{N}\sum_{i=1}^{N}\sum_{j=1}^{d}\alpha_{ij} \parallel \hat{x}_{ij} - \bar{x}_{ij} \parallel^2 + L_2 \tag{3.6}$$

where L_2 represents the L_2 regularization term. N is the number of samples in the d-dimensional dataset, $\bar{x}_{ij}$ is rough teacher signals constructed by replacing missing values by the mean imputation and $\hat{x}_{ij}$ represents the reconstructed jth component of the ith data sample, and α_{ij} is a coefficient defined by:

$$\alpha_{ij} = \begin{cases} \alpha & \text{when } x_{ij} \text{ is missing component} \\ 1 & \text{otherwise} \end{cases}$$

where $\alpha < 1$ is a parameter to reduce the effect of the uncertainty of teacher signals related to missing values. α is set as a smaller value at the beginning of the training so that the deep neural network can better learn latent relationships among observed samples, and then we increase the value of α to retrain the autoencoder.

Similar to the previous step, the second layer of the encoder uses h_1 as the input to train the second simple autoencoder by optimizing the loss function defined as:

$$E_2 = \frac{1}{N}\sum_{n=1}^{N} \| h_{1n} - \hat{h}_{1n} \|^2 + L_2 \tag{3.7}$$

where $\hat{h}_1$ is the reconstructed output given by the second autoencoder. Therefore, the remaining hidden layers can be pre-trained by repeating the aforementioned procedures.

3.3.2 Fine-tuning of the Full Network

During the fine-tuning procedure, we train the whole network through minimizing the reconstruction error like:

$$E_f = \frac{1}{N}\sum_{i=1}^{N}\sum_{j=1}^{d} \alpha_{ij} \| \hat{z}_{ij} - \bar{x}_{ij} \|^2 + L_2 \tag{3.8}$$

where $\hat{z}_{ij}$ is the reconstructed jth component of the ith data sample. On the other hand, the overcomplete SDAE is an overparameterized model so as to have the capability of estimating missing values by learning the latent relationships among samples. When training even with the L_2 regularization term, it is easy to be overfitting.

To account for the overfitting problem using an aggressive sparsity technique, we add a top $k\%$ WTA strategy in the last hidden layer of the encoder named feature layer h_F. Set the last hidden layer $\tilde{h}$ as the the input and $\{\tilde{\mathbf{w}}, \tilde{c}\}$ as the parameter set of the feature layer, by defining a set $\Gamma = supp_k\{a(\tilde{\mathbf{w}}^T\tilde{h} + \tilde{c})\}$ containing hidden units with top $k\%$ activation values, the representation h_F can be finally defined by:

$$h_F(p) = \begin{cases} a(\tilde{\mathbf{w}}^T\tilde{h} + \tilde{c}) & p \in \Gamma \\ 0 & p \notin \Gamma \end{cases} \tag{3.9}$$

where $p = 1, 2, ..., M$ and M is the number of nodes of the feature layer.

3.3.3 Updating Teacher Signals

By applying a standard WTA strategy to the pre-trained encoder, we cluster the data and obtain better estimates of missing values by imputing conditional mean for each cluster [87]. After updating teacher signals using better estimates of missing values, the aforementioned fine-tuning part can be trained repeatedly by using renewed teacher signals x' as the input. Finally, the full autoencoder is accomplished by minimizing the loss function E_{fin}:

$$E_{fin} = \frac{1}{N}\sum_{i=1}^{N}\sum_{j=1}^{d}\alpha_{ij} \parallel \hat{z}_{ij} - x'_{ij} \parallel^2 + L_2 \tag{3.10}$$

where z_{ij} represents generated new dataset. This procedure will be repeated. Our experiments show that repeating the fine-tuning by updating teacher signals for 1 or 2 times can result in better and more effective results. Therefore, the final decoded result can be regarded as the generating new dataset:

$$z \quad = \quad a(\bar{\mathbf{w}}\bar{h} + \bar{c}') \tag{3.11}$$

where $\{\bar{\mathbf{w}},\bar{c}'\}$ is the parameter set of the last decoder layer, and $\bar{h}$ is the representation of the previous layer.

3.3.4 Generation of Gated Signals

Generally, the basic strategy should be considered to build a 0-1-sequence. Therefore, an assumption is made to decide whether $g(\cdot)$ equals 0 or 1. In order to make full use of information about the feature layer efficiently to generate gate signals for the linear base model, we duplicate and redefine the feature layer when $p \in \Gamma$ as:

$$\begin{aligned} h_F = a(\tilde{\mathbf{w}}^T\tilde{h} + \tilde{c}) &= \max\{0, \tilde{\mathbf{w}}^T\tilde{h} + \tilde{c}\} \\ &= (\tilde{\mathbf{w}}^T\tilde{h} + \tilde{c})H(\tilde{\mathbf{w}}^T\tilde{h} + \tilde{c}) \end{aligned} \tag{3.12}$$

where $H(\cdot)$ is formulated as the step function. When estimating the missing values, the WTA strategy has already been applied to prevent overfitting, where top $k\% = 50\%$

has been applied to the hidden units in the feature layer, which satisfies the sparsity requirement. However, $k\% = 50\%$ results in the largest diversity since the number of partition is $C_M^{M\times k\%}$, which may not be suitable for generating gate control sequences. Therefore, we introduce another WTA of top $m\%(m < k)$ ones satisfying the assumption defined by $\zeta = supp_m\{a(\tilde{w_p}^T \hat{x} + \tilde{c_p})\}$ for generating gate control sequences. We finally define $g(p)$ as follows:

$$g(p) = \begin{cases} H(\tilde{w_p}^T \tilde{h} + \tilde{c_p}) & p \in \zeta \\ 0 & p \notin \zeta \end{cases} \tag{3.13}$$

It is an arbitrary choice for choosing $m\%$ and M, both of which can be regarded as tuning hyper-parameters. How to choose effective m and M affects the accuracy of solving both missing values and the nonlinear regression. For missing data problems, increasing the number M leads to the tendency of overfitting and too much training time. However, fewer M may also prevent the SDAE from extracting enough information. For the regression stage, a smaller $m\%$ and M is preferable to decrease the risk of overfitting and prevent too many partitions. Detailed numbers of hyper-parameters based on different datasets are illustrated in Section 3.5.

3.4 Gated Linear Network for Regression

By generating a set of gate control sequences $\mathbf{g}(z)$ from the SDAE, the gated linear network implements a piecewise linear regression model, which can be identified as a SVR with a quasi-linear kernel. Moreover, the multiple imputation tool, which results in unbiased and useful estimates for missing values, is also taken into account.

3.4.1 SVR with Quasi-linear Kernel

By importing two vectors Φ(z) and Θ defined as:

$$\Phi(z) = [g_1(z), z^T g_1(z), ..., g_M(z), z^T g_M(z)] \tag{3.14}$$

$$\Theta = [b_1, \Omega_1^T, ..., b_M, \Omega_M^T]^T \tag{3.15}$$

Eq.(3.2) can be expressed as a linear-in-parameter form as:

$$f(z) = \Theta^T \Phi(z) + b \tag{3.16}$$

where Φ(z) is the regression vector and Θ is called linear parameter vector. In the following, we concentrate on how to estimate linear parameters using SVR formulation. Based on the structural risk minimization principle as:

$$\begin{aligned} &\min_{\Theta, b, \xi_i, \xi_i^*} \frac{1}{2}\Theta^T\Theta + C\sum_{i=1}^{N}(\xi_i + \xi_i^*) \\ &\text{s.t.} \begin{cases} \Theta^T\Phi(z) + b - y_i \le \epsilon + \xi_i^* \\ y_i - \Theta^T\Phi(z) - b \le \epsilon + \xi_i \\ \xi_i, \xi_i^* \ge 0, \quad t = 1, 2, ..., N \end{cases} \end{aligned} \tag{3.17}$$

where y_i denoted the ideal output of z_i, C is a non-negative weight to determine the penalization of prediction errors, N is the number of observations, and ξ_t, ξ_t^* are slack variables. By applying the Lagrange function through introducing Lagrange multipliers $\mu \ge 0$, $\mu* \ge 0$, $\alpha \ge 0$, $\alpha* \ge 0$, we can construct the Lagrange function as:

$$\begin{aligned} L(\Theta, \xi_t, \xi_t*, \alpha, \alpha*, \mu, \mu*) &= \frac{1}{2}\Theta^T\Theta + C\sum_{i=1}^{N}(\xi_i + \xi_i^*) \\ &+ \sum_{i=1}^{N}\alpha_t(f_z(z) - \Phi(z)\Theta - \epsilon - \xi_t) + \sum_{i=1}^{N}\alpha_t(-f_z(z) \\ &+ \Phi(z)\Theta - \epsilon - \xi_i^*) - \sum_{i=1}^{N}(\mu\xi_i + \mu^*\xi_i^*) \end{aligned} \tag{3.18}$$

Then it can be solved by getting the saddle point:

$$\begin{aligned}
&\frac{\partial L}{\partial \Theta} = 0 \rightarrow \Theta = \sum_{i=1}^{N} (\alpha - \alpha^*)\Phi(z_i) \\
&\frac{\partial L}{\partial \xi} = 0 \rightarrow C = \alpha + \mu \\
&\frac{\partial L}{\partial \xi^*} = 0 \rightarrow C = \alpha^* + \mu^*
\end{aligned} \tag{3.19}$$

After converting the Lagrange function into its dual problem, we can get:

$$\begin{aligned}
\max\ W(\alpha, \alpha^*) = &-\frac{1}{2} \sum_{i,j=1}^{N} (\alpha_i - \alpha_i^*)(\alpha_j - \alpha_j^*) K(z_i, z_j) \\
&+ \sum_{i=1}^{N} (\alpha_i - \alpha_i^*) f(z) - \epsilon \sum_{i=1}^{N} (\alpha_i + \alpha_i^*) \\
\text{s.t.}\ \sum_{i,j=1}^{N} &(\alpha_i - \alpha_i^*) = 0. \quad \alpha, \alpha^* \in [0, C].
\end{aligned} \tag{3.20}$$

where $K(z_i, z_j)$ is a data-dependent composed kernel called quasi-linear kernel, defined as:

$$\begin{aligned}
K(z_i, z_j) &= \Phi^T(z_i)\Phi(z_k) \\
&= (1 + z_i z_j) \sum_{k=1}^{M} g_k(z_i) g_k(z_j)
\end{aligned} \tag{3.21}$$

From the above, with the Lagrange multipliers α_i and α_i^* obtained, the regression model can finally be represented as:

$$f(z) = \sum_{i=1}^{N} (\alpha_i - \alpha_i^*) K(z, z_i) + b. \tag{3.22}$$

3.4.2 Multiple Imputation for Missing Data

Since initialized weights of the SDAE are random at each run, kinds of new datasets can be generated as posterior predictive distributions of missing values. The fill-in

TABLE 3.1: Sizes and features of all datasets and network size

Datasets	Attribute	Samples	Net size of the SDAE	$m\%$
Steam	38	4332	38×60×90×140×90×60×38	30%
CO	9	7343	9×20×50×20×9	20%
NO_2	9	6609	9×20×40×20×9	30%
NO_x	9	6000	9×20×40×20×9	30%
Bank	16	4499	8×20×50×20×8	20%
PM2.5(TT)	10	28541	10×20×40×20×10	30%

procedure is repeated for prespecified number of times N to generate different new datasets $\{z^n\}_{n=1}^N$. We combine the piecewise regression model for each new dataset to make the prediction and then calculate an average of N models $Model_N$.

$$f(z) = \frac{1}{N}\sum_{n=1}^{N} Model_N(z^n) \tag{3.23}$$

To summarise, estimations of $f(z)$ appropriately reflect sampling diversity on account of incomplete values.

3.5 Experimental Results

3.5.1 The General Setup

Six real-world datasets that are complete downloaded from UCI machine learning repository [79] and Tianchi crowd intelligence platform [80] are taken to convince our hybrid model's effectiveness. In the experiment, all the inputs are normalized to a range of [0, 1]. Moreover,we split the dataset into 6 folds and for each imputation, 4-fold is used for training and the other 2-fold is set as the testing set. Therefore, 15 train-test splits are obtained to impute the results and mean standard deviation.

All of hyper-parameters are chosen through a 4-fold cross-validation method on the training set. The optimal learning rate of different datasets is chosen in the range of $\{1e\text{-}1, 1e\text{-}2, 1e\text{-}3\}$. For L_2 regularization term, weight decay is chosen within the grid $\{1e\text{-}3, 3e\text{-}3, 1e\text{-}2, 3e\text{-}2, 1e\text{-}1\}$. For parameters related with winner-take-all strategy, the

sparsity level k in the feature layer is selected from $\{40, 45, 50\}$, and another sparsity level m is chosen within a grid $\{15, 20, 25, 30, 35\}$. The optimal parameter α in loss function is searched in the grid of $\{0.1, 0.2, 0.3\}$ at the beginning, and then α is chosen from $\{0.8, 0.9, 1.0\}$. For the training of kernel SVR, the penalty parameter C is searched in the grid of $\{1e\text{-}2, 1e\text{-}1, 1e0, 1e1, 1e2\}$. Referring to Ref.[88], we set the number of hidden layers as 5 or 7 in our experiments. Besides, the number of the latter layer nodes is chosen with the 1.5-2.5 times of the dimensions of the former layer during the encoding stage. Table.3.1 shows the properties of datasets, network size, and the percentage of m used in generating gate signals, we also set $N = 20$ regarding the multiple imputation scenario and then calculate the standard deviation. In order to evaluate the performance, RMSE is used as the criterion defined by:

$$RMSE = \sqrt{\frac{1}{L}\sum_{i=1}^{L}(y_i' - y_i)^2} \tag{3.24}$$

where y_i' is the predicted data of corresponding the observed data y_i, L is the total number of the testing values.

Considering three missing mechanisms, two different experiments are conducted to analyze the effect of our proposed method. Firstly, missing data generation is executed by inserting missing data at six missing rates randomly (10%, 20%, 30%, 40%, 50%, 60%). To confirm that the effectiveness of the proposed model does not depend on the percentages of missing values, results without missing values are also considered. Afterward, we consider mixing two missing mechanisms MCAR and MNAR to test our proposed method, which is also satisfied in real life. That is, firstly, two attributes x_1 and x_2 are randomly selected from the dataset and their medians are calculated as m_1 and m_2, and then we set it to have missing data where $x_1 < m_1$ and $x_2 > m_2$. Besides, we remove datasets randomly with fixed missingness proportions of 20% and 60% respectively.

3.5.2 Numerical Experiments Procedure

To verify the validity of the proposed model, we compare it from two aspects. Firstly, SVR with radial basis function kernel (RBF-SVR) and MLP are combined with SDAEs

TABLE 3.2: Prediction results for all six tested regression datasets

Datasets	Models	Missing						
		0%	10%	20%	30%	40%	50%	60%
CO	Proposed Method	**0.8526±0.0002**	**0.8534±0.0003**	0.8960±0.0008	**0.9682±0.0012**	**1.1521±0.0013**	**1.4005±0.0018**	**1.9723±0.0029**
	RBF-SVR	0.9400±0.0003	0.9406± 0.0004	1.0121± 0.0011	1.1541± 0.0016	1.4774±0.0025	1.8589±0.0026	2.4590±0.0033
	MLP	0.8916±0.0011	0.8930± 0.0017	0.9689±0.0024	1.0595±0.0024	1.3608±0.0029	1.7474±0.0037	2.3187±0.0045
	MICE-RBF	1.0291±0.0008	1.0298±0.0012	1.1542± 0.0017	1.2702±0.0021	1.5342± 0.0024	2.1029±0.0028	2.6253±0.0037
	MICE-MLP	0.9214± 0.0011	0.9223± 0.0016	1.0268± 0.0027	1.1239± 0.0032	1.4553 ±0.0039	1.9061±0.0041	2.5717± 0.0045
	AP-DAE-QSVR	0.8564± 0.0003	0.8577±0.0003	**0.8956±0.0007**	1.0921±0.0013	1.1817±0.0016	1.5893±0.0022	2.0601±0.0031
	KSC-MVI	1.1093	1.1108	1.2911	-	-	-	-
NO_2	Proposed Method	**38.0492±0.0351**	**38.4571±0.0356**	**39.0593±0.0376**	**40.6838±0.0453**	**43.2971±0.0568**	**47.9019±0.0597**	**51.2146±0.0612**
	RBF-SVR	46.5059±0.0302	46.9153±0.0367	48.3417±0.0432	52.1103± 0.0491	56.4660 ±0.0587	63.4085±0.0642	70.2598± 0.0674
	MLP	41.9386±0.0417	42.9840±0.0485	48.1532± 0.0549	51.5351±0.0596	51.2869±0.0641	60.0843±0.0685	71.3112±0.0704
	MICE-RBF	47.8294±0.0313	48.7218±0.0408	51.4086± 0.0482	55.1611±0.0508	60.3695±0.0602	68.3817± 0.0684	74.7798±0.0713
	MICE-MLP	43.8791±0.0475	44.7362±0.0481	50.8569± 0.0576	53.1183±0.0601	55.4670±0.0685	64.2916±0.0711	74.9248±0.0720
	AP-DAE-QSVR	39.0812±0.0329	39.7571±0.0355	40.6103±0.0390	41.1295±0.0507	44.1148±0.0602	50.4604±0.0523	53.1101±0.0627
	KSC-MVI	46.9382	48.1729	53.1221	-	-	-	-
NO_x	Proposed Method	**100.3120±0.1539**	**101.0274±0.1990**	**103.8153±0.1689**	**106.2911±0.2005**	**111.1936±0.2884**	**118.9967±0.3610**	**126.1032±0.4221**
	RBF-SVR	111.0975±0.1993	112.1302±0.1991	115.4068±0.2032	121.9897±0.2420	128.1862±0.2941	143.0125±0.4007	160.9272±0.4189
	MLP	108.9130±0.2002	109.2414±0.2042	109.7991±0.2407	115.2098±0.3397	122.4933±0.3938	130.6917±0.4508	155.4991±0.4593
	MICE-RBF	113.4177±0.1320	114.4801±0.1982	117.7192±0.2185	122.2007±0.3978	130.9027±0.3816	145.2834±0.4890	163.5229±0.4706
	MICE-MLP	110.1592±0.1896	110.9957±0.2179	111.3653±0.3830	116.4387±0.3301	123.1080± 0.3642	132.1356 ±0.4500	157.4287±0.5313
	AP-DAE-QSVR	102.6092±0.1433	103.1454±0.1237	106.9235±0.1982	109.2568±0.2311	114.7809±0.2947	122.7451±0.3086	127.9067±0.3990
	KSC-MVI	109.3542	111.9490	121.7851	-	-	-	-
Steam	Proposed Method	**0.2907±0.0003**	**0.2993±0.0003**	**0.3052±0.0004**	**0.3170±0.0005**	**0.3391±0.0005**	**0.3587±0.0007**	**0.3679±0.0011**
	RBF-SVR	0.3122±0.0003	0.3203±0.0003	0.4017±0.0006	0.4095±0.0007	0.4570±0.0007	0.4741±0.0010	0.5227±0.0014
	MLP	0.3024±0.0004	0.3105±0.0005	0.3859±0.0009	0.3931±0.0009	0.4486±0.0013	0.4827±0.0013	0.5140±0.0018
	MICE-RBF	0.3372±0.0003	0.3496±0.0004	0.4173±0.0005	0.4236±0.0007	0.4522±0.0009	0.4608±0.0010	0.4941±0.0014
	MICE-MLP	0.3183±0.0004	0.3247±0.0004	0.3926±0.0007	0.4155±0.0010	0.4683±0.0015	0.4761±0.0016	0.5160±0.0018
	AP-DAE-QSVR	0.3018±0.0003	0.3031±0.0003	0.3230±0.0005	0.3363±0.0005	0.3572±0.0006	0.3799±0.0008	0.3982±0.0010
	KSC-MVI	0.4969	0.5821	-	-	-	-	-
Bank	Proposed Method	0.0269±0.0002	0.0272±0.0003	**0.0273±0.0003**	**0.0285±0.0004**	**0.0277±0.0005**	**0.0292±0.0004**	**0.0309±0.0006**
	RBF-SVR	**0.0267±0.0001**	**0.0271±0.0001**	0.0288±0.0001	0.0313±0.0003	0.0316±0.0004	0.0335±0.0004	0.0353±0.0005
	MLP	0.0274±0.0003	0.0277±0.0003	0.0283±0.0004	0.0303±0.0004	0.0315±0.0007	0.0328±0.0009	0.0351±0.0011
	MICE-RBF	0.0272±0.0003	0.0276±0.0003	0.0293±0.0005	0.0321±0.0005	0.0330±0.0009	0.0344±0.0011	0.0371±0.0014
	MICE-MLP	0.0277±0.0003	0.0280±0.0004	0.0287±0.0006	0.0305±0.0008	0.0329±0.0009	0.0335±0.0012	0.0368±0.0017
	AP-DAE-QSVR	0.0270±0.0001	0.0274±0.0002	0.0279±0.0003	0.0295±0.0003	0.0287±0.0004	0.0302±0.0005	0.0318±0.0004
	KSC-MVI	0.0282	0.0289	0.0321	-	-	-	-
PM2.5(TT)	Proposed Method	**35.9801±0.0531**	**36.3723±0.0644**	**37.8388±0.0879**	**39.9609±0.0897**	**43.1198±0.1005**	**47.5060±0.1142**	**52.0104±0.1272**
	RBF-SVR	38.9252±0.0527	40.1470±0.0686	43.6241±0.0853	47.1182±0.0912	52.3550±0.1004	58.6259±0.1189	70.2563±0.1299
	MLP	37.2941±0.0691	37.9748±0.0748	40.2535±0.0903	43.8859±0.1032	47.2821±0.1164	54.5000±0.1377	61.4231±0.1531
	MICE-RBF	39.0379±0.0578	41.6125±0.0773	44.9273±0.0948	49.2534±0.0005	55.0517±0.1096	60.9227±0.1206	72.0133±0.1320
	MICE-MLP	38.3471±0.0727	40.8069±0.0916	42.6520±0.0981	45.6359±0.1105	50.8408±0.1183	56.3541±0.1256	67.8093±0.1454
	AP-DAE-QSVR	36.0792±0.0398	36.9993±0.0579	38.3412±0.0637	41.6369±0.0725	45.7286±0.0906	49.2351±0.1050	55.8742±0.1186
	KSC-MVI	39.8023	46.2486	50.9811	-	-	-	-

TABLE 3.3: Prediction results of mixed missing data

Datasets	Models	Missing	
		20%	60%
CO	Proposed Method	**0.9027±0.0009**	**1.9917±0.0030**
	RBF-SVR	1.0477± 0.0015	2.4903±0.0035
	MLP	0.9902±0.0027	2.4691±0.0049
	MICE-RBF	1.1847±0.0017	2.6909±0.0042
	MICE-MLP	1.0580±0.0030	2.6233±0.0051
	AP-DAE-QSVR	0.9052±0.0009	2.0819±0.0032
	KSC-MVI	1.3469	-
NO_2	Proposed Method	**39.4761±0.0381**	**52.4034±0.0610**
	RBF-SVR	49.1037±0.0441	72.6330±0.0681
	MLP	49.1421±0.0538	73.7772±0.0728
	MICE-RBF	52.1699±0.0491	76.0714±0.0720
	AP-DAE-QSVR	40.9463±0.0387	54.7729±0.0669
	KSC-MVI	55.2581	-
NO_x	Proposed Method	**104.7605±0.1793**	**128.0278±0.4017**
	RBF-SVR	117.0897±0.2077	164.0531±0.4129
	MLP	111.4266± 0.2580	157.8426±0.4690
	MICE-RBF	119.6138±0.2591	166.2327±0.4793
	MICE-MLP	121.1152±0.3117	177.8541±0.5697
	AP-DAE-QSVR	105.9787±0.1948	129.7165±0.4544
	KSC-MVI	125.0198	-
Steam	Proposed Method	**0.3098±0.0004**	**0.3708±0.0012**
	RBF-SVR	0.4061±0.0007	0.5269±0.0015
	MLP	0.3927±0.0009	0.5201±0.0027
	MICE-RBF	0.4217±0.0006	0.5053±0.0016
	MICE-MLP	0.4063±0.0007	0.5211±0.0022
	AP-DAE-QSVR	0.3291±0.0005	0.4045±0.0013
	KSC-MVI	-	-
Bank	Proposed Method	**0.0278±0.0004**	**0.0321±0.0007**
	RBF-SVR	0.0298±0.0001	0.0376±0.0006
	MLP	0.0297±0.0004	0.0367±0.0012
	MICE-RBF	0.0302±0.0005	0.0391±0.0014
	MICE-MLP	0.0295±0.0005	0.0383±0.0017
	AP-DAE-QSVR	0.0286±0.0003	0.0337±0.0006
	KSC-MVI	0.0329	-
PM2.5(TT)	Proposed Method	**38.0172±0.0891**	**52.5730±0.1296**
	RBF-SVR	44.2480±0.0886	70.9757±0.1374
	MLP	40.9723±0.0935	62.0813±0.1629
	MICE-RBF	45.5362±0.0962	72.9448±0.1448
	MICE-MLP	43.4899±0.0997	68.7432±0.1562
	AP-DAE-QSVR	39.0210±0.0641	56.6105±0.1299
	KSC-MVI	52.4877	-

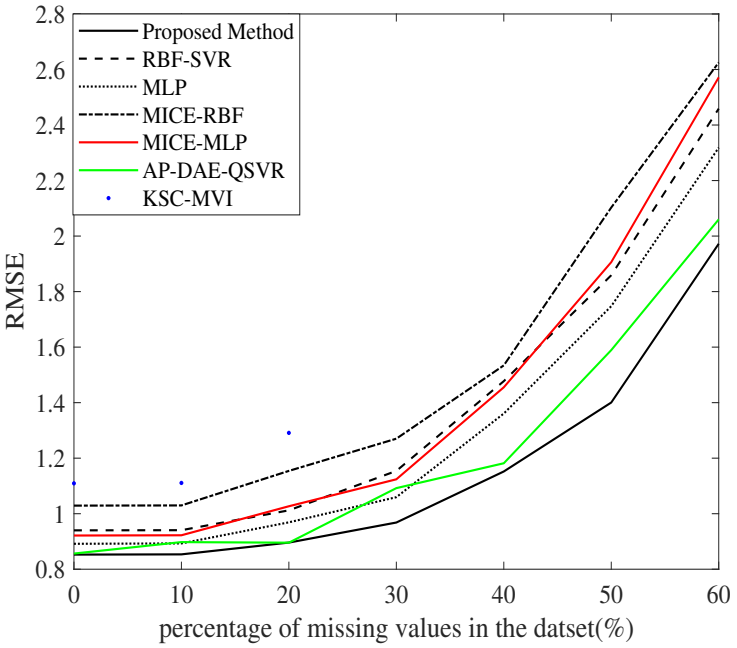


FIGURE 3.3: RMSE for CO data

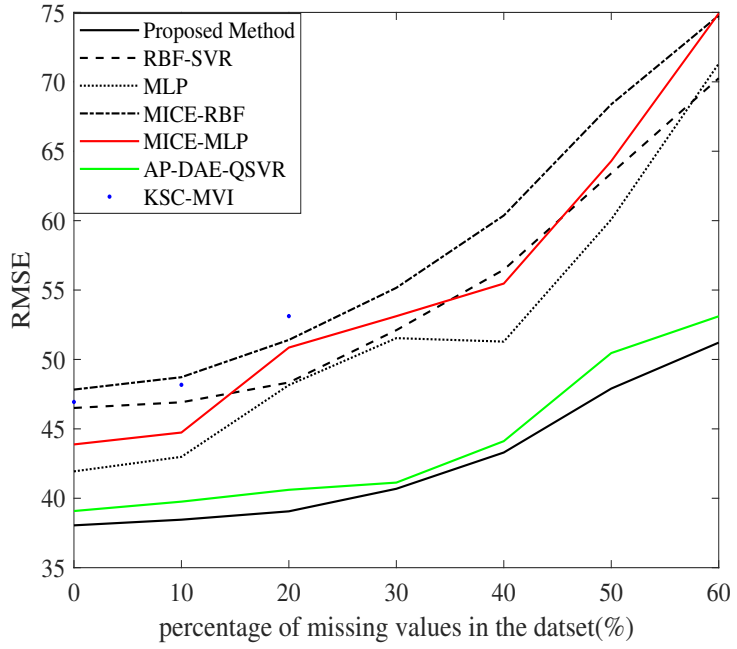


FIGURE 3.4: RMSE for NO_2 data

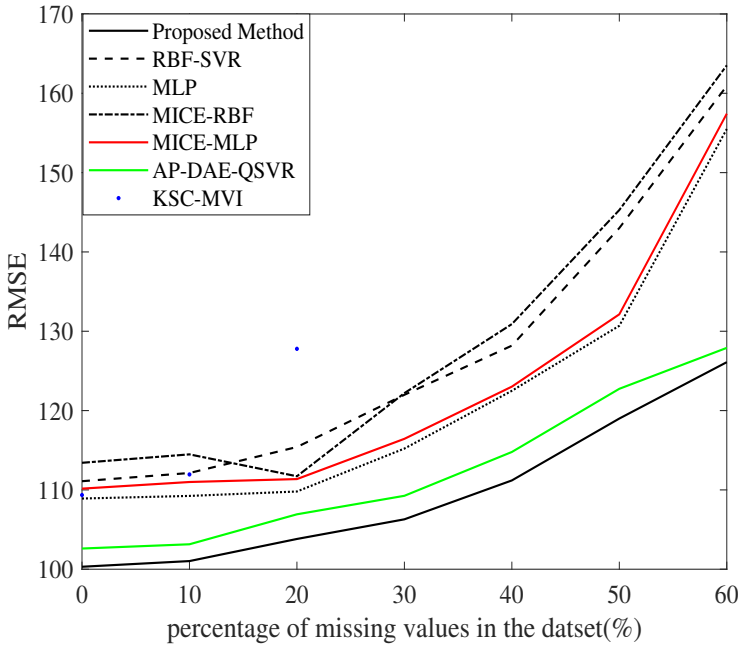


FIGURE 3.5: RMSE for NO_x data

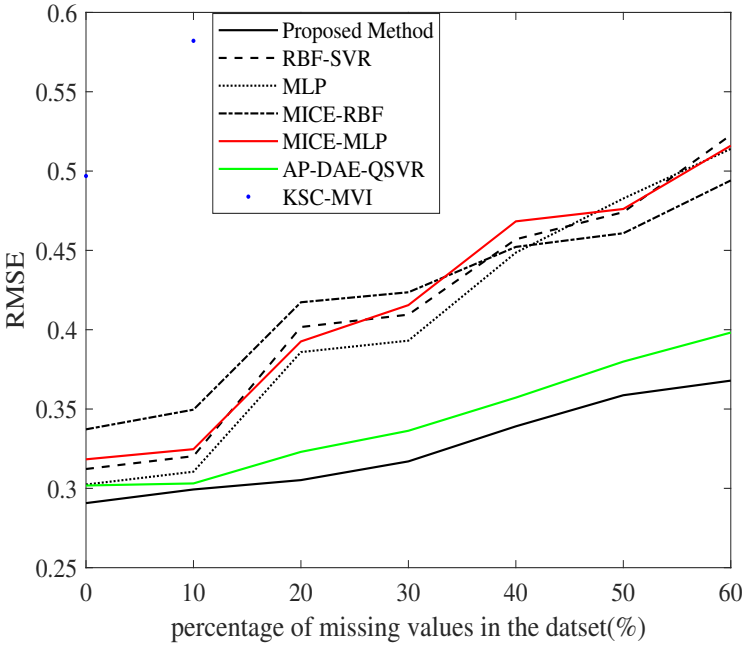


FIGURE 3.6: RMSE for steam data

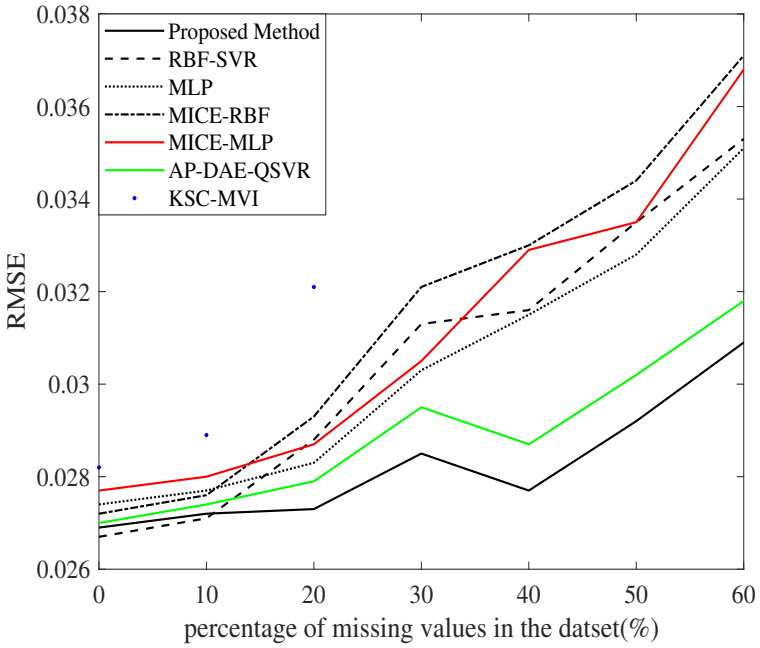


FIGURE 3.7: RMSE for bank data

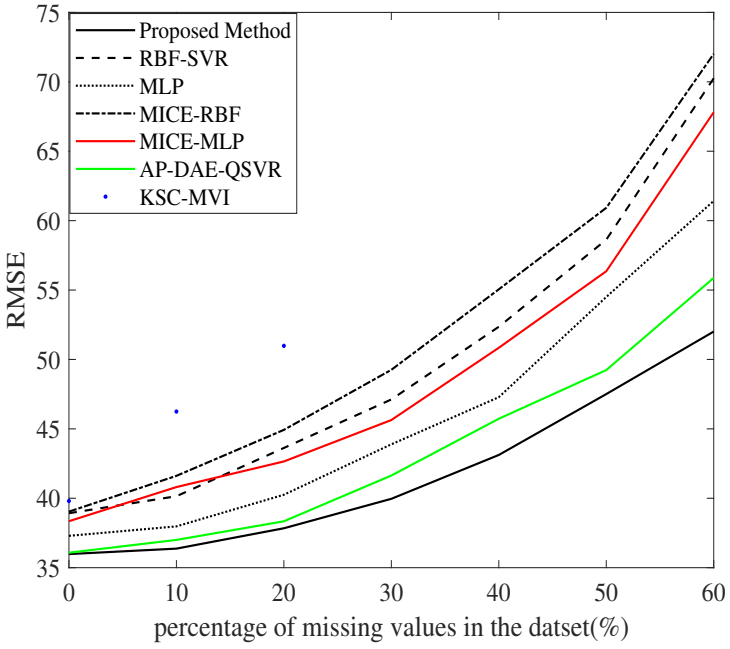


FIGURE 3.8: RMSE for $PM_{2.5}$(TT) data

respectively to calculate the prediction accuracy. Moreover, we also examine the performance of SDAEs, that is, the well known multiple imputation method multivariate imputation by chained equations (MICE) [89] in combination with SVR with RBF kernel (MICE-RBF) and MLP (MICE-MLP) are all conducted in the experiments. Besides, two prevailing prediction methods with missing values including KSC clustering with MVI kernel method (KSC-MVI) proposed in Ref.[83] and affinity propagation (AP) clustering combined with DAE and the hybrid model (AP-DAE-QSVR) developed by us in Ref.[90] are also tested for comparison. The results of all the experiments are shown in Table.3.2 and Table.3.3, respectively, and the numbers highlighted with boldface mean that the corresponding model achieves the first-rank for easy comparison.

From Table 3.2 and Fig.3.3-3.8, we can conclude several observations. Firstly, our proposed model achieves better performance than traditional regression methods when the amount of missing data is small, except for bank data shown in Fig.3.7. RBF-SVR is higher than our proposed method's RMSE value for bank data with 10% missing values. However, RBF-SVR has to set complex parameters with case dependent. Secondly, the advantage of our proposed method increases as the percentage of missing data rises. We can see that the growth of RMSE value for our proposed model is the slowest with the increasing number of missing data. Thirdly, the SDAE method is superior to the MICE method for solving missing data problems. Finally, for KSC-MVI model, it is risky when the amount of missing values is large since unobserved samples are removed during the preprocessing procedure. Like the steam data, there are not enough complete training samples to run while the percentage of missing values is greater than 10%. Compared with AP-DAE-QSVR method, the proposed method gives a competitive results with lower RMSE values except CO dataset with 20% missing values. The proposed model is improved version of the AP-DAE-QSVR model with a more flexible gating mechanism and does not need to detect local linear partitions and set the number of local linear models. Therefore, it leads to the conclusion that the proposed model is more stable and effective at dealing with the regression problem with missing data.

Experimental results of mixed missing data mechanisms are shown in Table.3.3. We can see that the performance of our proposed methods exceeds other state-of-the-art methods for CO data. It learns to the conclusion that our proposed model achieves the

best performance for every case. The possible reason is that the SDAE has the capability to learn latent associations among datasets.

3.6 Conclusions

In this chapter, a hybrid modeling method is presented in dealing with the regression task under the missing data scenario. The complete model composes of two steps. WTA SDAEs are employed to tackle missing data problems, and then a gated linear network is designed to construct a piecewise linear model. In this way, SVR with a data-dependent quasi-linear kernel can be used for prediction tasks.

Two comparative experiments are conducted based on different missing data mechanisms. The accuracy and robustness of the proposed model has been verified by both experimental results of five real-world datasets. Even for a large fraction of missing data, the role of our proposed model is also apparent.

Chapter 4

An Improved Hybrid Model for Nonlinear Regression with Missing Values Using Deep Quasi-Linear Kernel

4.1 Introduction

[1] The objective of regression analysis in the research field is to establish a model which can examine the relationship between a response variable and one or more independent variables. Among these, nonlinear regression is a kind of regression analysis and has been widely used in many practical applications, such as health forecasting [3]. Moreover, missing values are ubiquitous and inevitable in the nonlinear regression field, which often complicate analysis and cause trouble for the further research. Traditional methods to deal with missing values include mean interpolation, KNN [91] and Expectation Maximization [92]. Currently, imputation methods based on deep neural networks such as DAE [33], SDAE [93] and generative adversarial nets (GAN) [36]

[1]This chapter mainly extends the Journal paper: H.Zhu and J. Hu, "An Improved Hybrid Model for Nonlinear Regression with Missing Values Using Deep Quasi-Linear Kernel", *IEEJ Trans. on Electrical and Electronic Engineering*, Vol.17, No.10, PP.1-9, 2022.

are rousing attention due to learning latent relationships among datasets. On the other hand, no filling-in algorithms can achieve 100% accuracy, and noised features are inevitable. Some classical solutions like linear regression [94] are difficult to unearth and model such complex nonlinear relationships among samples. It is highly motivated to develop a more robout, powerful piecewise linear regression model to solve problems as mentioned above.

In our previous work, hybrid models have been proposed to solve nonlinear regression problems under missing data scenarios [90, 95], which consists of two parts: 1) an overcomplete WTA autoencoder [96] to impute missing components conditioned on observed samples and generate a set of gate control signals, 2) a gated linear network with generated gate control signal implementing flexible multi-local linear models. In this paper, the hybrid model is improved in two aspects. On the one hand, an adversarial training process is introduced to train the WTA autoencoder. Under missing data scenarios, a key difficulty is the lack of an accurate teacher signal for the training of WTA autoencoder, and the uncertainty of the teacher signals limits the performance of the WTA autoencoder. In Ref. [95], we use a clustered imputation of missing values as the updated teacher signals. By using the WTA autoencoder as a generator and introducing a discriminator, we can expect a better adversarial training of the WTA autoencoder by taking advantage of gradually renewed teacher signals and the discrimination of missing values and observed values. In addition to the missing values estimation, the WTA autoencoder is trained to realize sophisticated partitioning by generating a set of layered binary gate control sequences. On the other hand, by using a multilayer gated linear network and the generated layered gate control sequences, we implement a more powerful piecewise linear regression model, whose parameters are then optimized by formulating a SVR with a deep quasi-linear kernel in a recursive form [62, 63]. Experimental results on several real-world datasets show that the improved hybrid model is effective.

The rest of this chapter is organized as follows. The structure of the hybrid model is presented in Section 4.2. Section 4.3 explains how to solve missing data problems and generate layered gate control sequences. Section 4.4 introduces the multilayer gated linear network for the nonlinear regression problem. In Section 4.5, the experimental

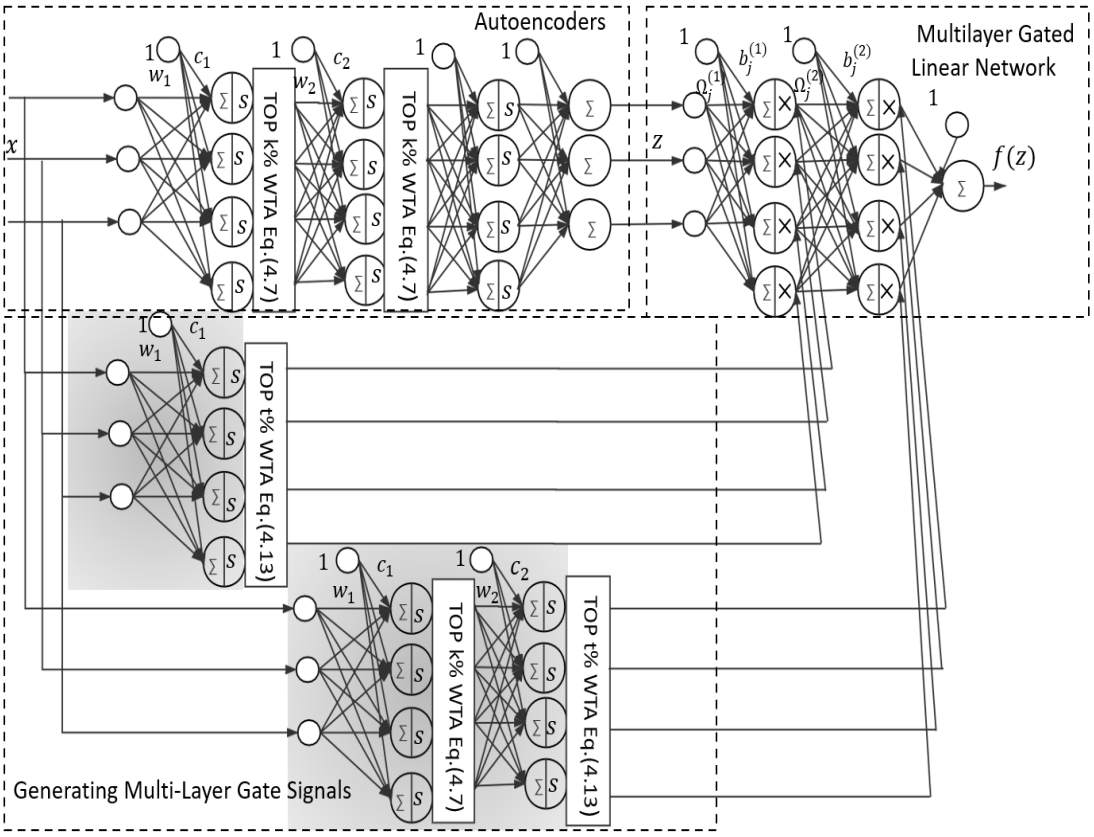


FIGURE 4.1: A hybrid prediction model consisting of a WTA autoencoder and a multilayer gated linear network. The encoder parts of WTA autoencoder used for filling in missing values and for generating gate signals are the same.

results are shown to validate the effectiveness of the proposed model. Finally, conclusions are given in Section 4.6.

4.2 Model Structure

In the case of missing data problem, we firstly consider a d-dimensional training set $(\mathbf{x}_1,\mathrm{y}_1)$, $(\mathbf{x}_2,\mathrm{y}_2)$,..., $(\mathbf{x}_n,\mathrm{y}_n)$. $\mathbf{x}_n=[\mathrm{x}_n^1, \mathrm{x}_n^2,..., \mathrm{x}_n^d]\in\mathrm{R}^d$ represents the n-th input and $y_n\in R$ is the correspondent n-th target output.

Considering the nonlinear regression problem under missing data scenarios, we establish a hybrid model composed of an overcomplete WTA autoencoder and an multilayer gated linear network. As shown in Fig.4.1, the WTA autoencoder will fill-in the missing

values and generate complete datasets z defined by:

$$z = \text{WTA\_AE}(\mathbf{w}, \mathbf{c}, x) \tag{4.1}$$

where $\{\mathbf{w}, \mathbf{c}\}$ is the parameter set, and the multilayer gated linear network realizing a powerful piecewise linear model is defined by [62, 63]:

$$f(z) = \sum_{j=1}^{M_S} \left(\Omega_j^{(S)T} a_{S-1}(z) + b_j^{(S)}\right) g_j^{(S)}(z) + b \tag{4.2}$$

$$a_i(z) = \sum_{j=1}^{M_i} \left(\Omega_j^{(i)T} a_{i-1}(z) + b_j^{(i)}\right) g_j^{(i)}(z) \tag{4.3}$$
$$i = 1, 2, ..., S-1; \quad a_0(z) = z$$

where S is the number of layers in the multilayer gated linear network, M_i is the number of linear base models $\Omega_j^{(i)T} a_{i-1}(z) + b_j^{(i)}$ in the i-th layer, $\{\Omega_j^{(i)}, b_j^{(i)}, b\}$ is the parameter set, and $g_j^{(i)}(z) \in \{0, 1\}$ is a gate signal controlling whether the j-th base model in the i-th layer works.

A gating mechanism for generating gate control sequences $\mathbf{g}^{(i)}(z) = [g_1^{(i)}(z), g_2^{(i)}(z), ..., g_{M_i}^{(i)}(z)]^T$ $(i = 1, 2, ..., S)$ is built by using the encoder part of WTA autoencoder, defined by:

$$\mathbf{g}^{(i)}(z) = \text{WTA\_E}_i(\mathbf{w}, \mathbf{c}, x). \tag{4.4}$$

Note that as shown in Fig.4.1, the encoder parts of the WTA autoencoder used for filling in missing values and for generating gate signals are the same. The details of $\text{WTA\_AE}(\mathbf{w}, \mathbf{c}, x)$ and $\text{WTA\_E}_i(\mathbf{w}, \mathbf{c}, x)$ will be discussed in Section 4.4 and 4.5, respectively.

In this section, the adversarial training process is introduced in the autoencoder part to solve the missing data problem and generate layered gate control sequences.

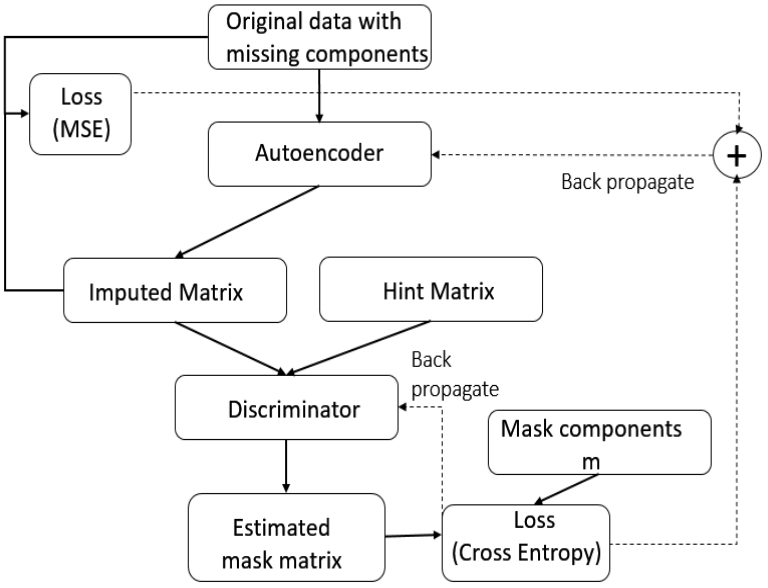


FIGURE 4.2: The architecture of our proposed adversarial training process

4.3 Autoencoders for Filling-in Missing Values

The traditional autoencoder is an unsupervised learning algorithm, which is mainly used for feature extraction or dimensional reduction. For the encoder phase, an autoencoder takes an input $x \in [0,1]^d$ and then maps it into a different representation $h \in [0,1]^{d'}$, where we set $d' > d$ since the autoencoder is overcomplete. Then h can be mapped back into the decoder phase. As an example, we consider a 5-layer symmetrical autoencoder described as:

$$h_F = a(\mathbf{w}_2(a(\mathbf{w}_1 x + c_1)) + c_2) \tag{4.5}$$

$$z = a\Big(\mathbf{w}_1^T\Big(a(\mathbf{w}_2^T h_F + c_3)\Big) + c_4\Big) \tag{4.6}$$

where z is a prediction of x based on a reconstruction from the feature, h_F represents the final layer of the encoder, the parameter set is $\{\mathbf{w}_1, \mathbf{w}_2, c_1, c_2, c_3, c_4\}$, and $a(\cdot)$ is ReLU activation function in our model.

In this chapter, we use DAEs to solve missing data problems, which are natural unsupervised extensions of traditional autoencoders. DAEs are forced to map corrupted input data caused by missing mechanisms or distributional additive noise into hidden layers to

learn latent features. Therefore, the missing data problem can be seen as a special case that allows the DAE to effectively recover the missing schema. Moreover, to solve the overfitting problem, we add a top $k\%$ WTA strategy to each hidden layer of the encoder to avoid the overfitting problem. Set upper layer of the i-th hidden layer $\tilde{h}$ as the input ($\tilde{h}$ can also be equal to x if i=1), S as the number of hidden layers of the encoder and $\{\tilde{\mathbf{w}}, \tilde{c}\}$ as the parameter set, by defining a set $\Gamma = supp_k\{a(\tilde{\mathbf{w}}^T\tilde{h} + \tilde{c})\}$ containing hidden units with top $k\%$ activation values, the representation of each hidden layer $h^{(i)}$ can be finally defined by:

$$h^{(i)}(p) = \begin{cases} a(\tilde{\mathbf{w}}^T\tilde{h} + \tilde{c}) & p \in \Gamma \\ 0 & p \notin \Gamma \end{cases} \tag{4.7}$$

where $p = 1, 2, ..., M_i$ and M_i is the number of nodes in the i-th layer, $i = 1, 2, ..., S$.

4.3.1 Adversarial Training Process

The adversarial training process is then applied, and the aforementioned DAE is set as the generator. Fig.4.2 depicts its architecture. Considering discriminator part, it is constructed as fully connected neural nets. To ensure that the discriminator forces the autoencoder to learn the desired distributions, we introduce a hint variable $\mathbf{H}_n = \{H_n^1, ..., H_n^d\}$ for the n-th sample as additional input to distinguish between observed and imputed values. By controlling the amount of information contained in $\mathbf{H}_n$ about missing values, we define $\mathbf{H}_n$ for the n-th input by first sampling υ from $\{1, ..., d\}$ (d is the dimension of each sample) at random as:

$$H_n^i = \begin{cases} 1 & if\ the\ component\ can\ be\ observed \\ 0.5 & if\ i = \upsilon \\ 0 & if\ the\ component\ is\ missing \end{cases} \qquad (i = 1, ..., d) \tag{4.8}$$

The matrix $\mathbf{H}$ (where $\mathbf{H} = \{\mathbf{H}_1, ..., \mathbf{H}_N\}$ and N is the number of samples) can also be seen as the hint mechanism since we can define $\mathbf{H}$ in different ways to control the amount

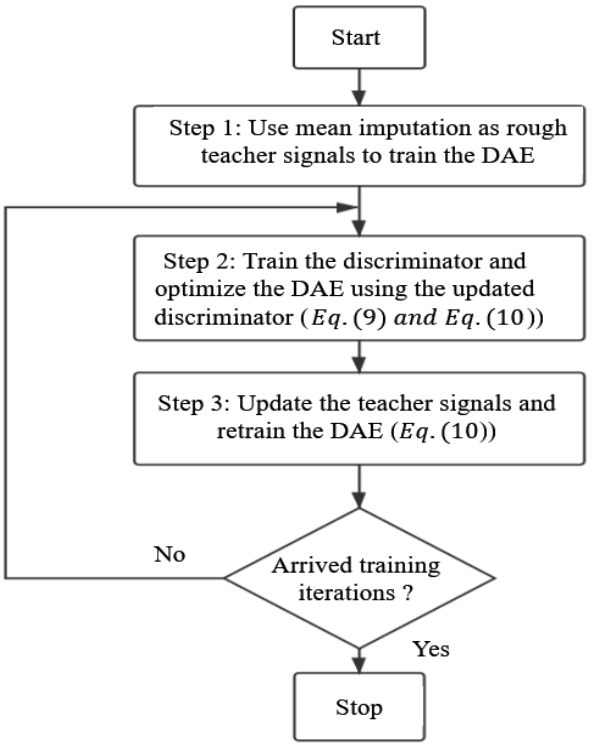


FIGURE 4.3: Flowchart of training process in the autoencoder part

of information contained in $\mathbf{H}$ to the discriminator. For instance, if we cannot provide information about the missing data, there are several distributions that the autoencoder could be reproduced to be optimal for the discriminator. Therefore, the discriminator becomes a function : $\mathbf{x} \times \mathbf{H} \rightarrow [0,1]^d$. To optimize the discriminator, we compare it with the mask components m, where $m = 0$ means that the component of $\mathbf{x}$ is missing and $m = 1$ corresponds to the probability that the component is observed. Hence, the loss function of D is finally defined as:

$$L_D = \sum_{i=1}^{N} \sum_{j:H_i^j=0.5}^{d} [m_i^j \log(\hat{m}_i^j) + (1 - m_i^j) \log(1 - \hat{m}_i^j)] \tag{4.9}$$

where $\hat{m}$ is the output of the discriminator. In the loss function, we only consider the ones corresponding to $H_i^j = 0.5$ to account for overfitting problem.

Then, the generator is optimized by using the newly discriminator. Since we ensure that

the imputed values for missing components (m=0) can fool the discriminator successfully. Therefore, the loss function of the DAE can be updated as:

$$\begin{aligned} L_G = & \frac{1}{N}\sum_{i=1}^{N}\sum_{j=1}^{d}\zeta_i^j \parallel \hat{z}_i^j - \bar{x}_i^j \parallel^2 \\ & - \sum_{i=1}^{N}\sum_{j:H_i^j=0.5}^{d}(1-m_i^j)\log\hat{m}_i^j + \tau\|\mathbf{w}\|_2{}^2 \end{aligned} \tag{4.10}$$

where $\mathbf{w} = \{\mathbf{w}_1, \mathbf{w}_2\}$ represents the weights of the antoencoder, and $\tau\|\mathbf{w}\|_2{}^2$ represents the L_2 regularization term. N is the number of samples in the d-dimensional dataset, $\hat{z}_i^j$ is the reconstructed j-th component of the i-th data sample, and $[-\sum_{i=1}^{N}\sum_{j:H_i^j=0.5}^{d}(1-m_i^j)\log\hat{m}_i^j]$ is smaller when $\hat{m}_i^j$ is closer to 1 when $m_i^j = 0$. In other words, the effect we want to achieve is that D is less able to recognize the imputed samples as being imputed. $\bar{x}_i^j$ is rough teacher signals constructed by replacing missing values by the mean imputation, and ζ_i^j is a coefficient defined by:

$$\zeta_i^j = \begin{cases} \zeta & \text{when } x_i^j \text{ is missing component} \\ 1 & \text{otherwise} \end{cases} \tag{4.11}$$

where $\zeta < 1$ is a parameter to reduce the effect of the uncertainty of teacher signals related to missing values. ζ is set as a smaller value at the beginning of the training so that the autoencoder can better learn latent relationships among observed samples, and then we increase the value of ζ to retrain the autoencoder.

4.3.2 Updating Teacher Signals

Next, more effective teacher signals need to be obtained to improve the accuracy of data filling and guide the convergence. Given the trained last hidden layer of the encoder, we only select the top one hidden layer unit activated for each data. Therefore, samples with the same activated unit are seen as the same cluster. Then, we impute conditional mean for each cluster to replace the missing data points. After updating teacher signals,

$\bar{x}$ in Eq.(4.10) can be changed into renewed teacher signals x' as the input. Fig.4.3 shows the flowchart of the overall training procedure.

4.3.3 Generation of Gate Control Signals

Next, we introduce a basic strategy for establishing a 0-1 sequence. In order to make full use of the information of hidden layers in autoencoders, we express the i-th hidden layer as:

$$\begin{aligned} h^{(i)} = a(\tilde{\mathbf{w}}^T \tilde{h} + \tilde{c}) &= \max\{0, \tilde{\mathbf{w}}^T \tilde{h} + \tilde{c}\} \\ &= (\tilde{\mathbf{w}}^T \tilde{h} + \tilde{c}) F(\tilde{\mathbf{w}}^T \tilde{h} + \tilde{c}) \end{aligned} \tag{4.12}$$

where $F(\cdot)$ is formulated as the step function, and $i = 1, 2, ..., S$.

When estimating missing values, the WTA strategy has been applied to prevent over fitting, where top $k\% = 50\%$ has been applied to each hidden layers that meet the sparsity requirements. However, $k\% = 50\%$ results in maximum diversity because the number of partitions is $C_{M_i}^{M_i \times k\%}$ in the i-th layer, which is not suitable for partitioning based on previous work [95, 97] since too many partitions may increase the risk of overfitting for the piecewise linear model. Therefore, we introduce another WTA strategy with top $t\%(t < k)$ ones satisfying the assumption defined by $\varpi = supp_t\{a(\tilde{w_p}^T \hat{x} + \tilde{c_p})\}$ for generating multi-layer gate control sequences $g^{(i)}(.)$. We finally define the i-th $g^{(i)}(.)$ as follows:

$$g^{(i)}(p) = \begin{cases} F(\tilde{w_p}^T \tilde{h} + \tilde{c_p}) & p \in \varpi \\ 0 & p \notin \varpi \end{cases} \tag{4.13}$$

4.4 Multilayer Gated Linear Network

In this section, a deep quasi-linear kernel is derived by applying an SVR formula to the multilayer gated linear network in a recursive form. When the gate signals are given, the multi-layer gated linear network can reduce to a linear model. By denoting $\Phi_0 = z$

and $\Theta_0 = []$, we import two vectors $\Phi_S(z)$ and Θ_S defined as:

$$\begin{aligned}\Phi_i(z) &= [g_1^{(i)}(z), \Phi_{i-1}^T(z) g_1^{(i)}(z), ..., g_{M_i}^{(i)}(z), \Phi_{i-1}^T(z) g_{M_i}^{(i)}(z)] \\ &= [\mathbf{g}^{(i)T}(z) \otimes [1 \;\; \Phi_{i-1}^T(z)]]^T \qquad (4.14) \\ \Theta_i &= [\mathbf{\Omega}^{(i)T} \otimes [1 \;\; \Theta_{(i-1)}^T]]^T \quad (i = 1, 2, ..., S) \qquad (4.15)\end{aligned}$$

where $\mathbf{g}(z) = [g_1(z), ..., g_M(z)]^T$, and $\otimes$ represents Kronecker production, $\mathbf{\Omega}^i = [b_1^{(i)}, \Omega_1^{(i)T}, ..., b_{M_i}^{(i)}, \Omega_{M_i}^{(i)T})]^T$ in which $b_j^i (j = 1, ..., M_i)$ is the bias of i-th layer in the encoder.

$\Phi_S(z)$ gives a multi-linear mapping shown in the front part of Fig.4.1 since for each given $\mathbf{g}^{(i)}(z)$ it is a linear mapping. Therefore, the S-layer gated linear network can be compactly expressed as a linear-in-parameter form as:

$$f(z) = \Theta_S^T \Phi_S(z) + b \qquad (4.16)$$

To estimate the parameters Θ_i (i=1, 2, ..., S), the SVR formulation is used to solve the regression form $f(z)$ in Eq.(4.16). Same as the process of basic SVR, by applying the structural risk minimization principle, we concentrate on quadratic programming (QP) optimization problem described by:

$$\begin{aligned}&\min_{\Theta_S, b, \xi_l, \xi_l^*} \frac{1}{2} {\Theta_S}^T \Theta_S + C \sum_{l=1}^{N} (\xi_l + \xi_l^*) \\ &\text{s.t.} \begin{cases} {\Theta_S}^T \Phi_S(z_l) + b - y_l \le \epsilon + \xi_l^* \\ y_l - {\Theta_S}^T \Phi_S(z_l) - b \le \epsilon + \xi_l \\ \xi_l, \xi_l^* \ge 0, \;\; l = 1, 2, ..., N \end{cases}\end{aligned} \qquad (4.17)$$

where y_l denoted the ideal output of z_l, C is a non-negative weight to determine the penalization of prediction errors, N is the number of observations, and ξ_l, ξ_l^* are slack variables. By applying the Lagrange function through introducing Lagrange multipliers

$\mu_l \geq 0$, $\mu_l^* \geq 0$, $\alpha_l \geq 0$, $\alpha_l^* \geq 0$, the Lagrange function can be constructed as:

$$\begin{aligned} L(\Theta_S,\xi,\xi^*,\alpha,\alpha^*,\mu,\mu^*) &= \frac{1}{2}{\Theta_S}^T\Theta_S + C\sum_{l=1}^{N}(\xi_l+\xi_l^*) \\ &+\sum_{l=1}^{N}\alpha_l(f(z_l)-y_l-\epsilon-\xi_l)+\sum_{l=1}^{N}\alpha_l^*(-f(z_l)+y_l- \\ &\epsilon-\xi_l^*)-\sum_{l=1}^{N}(\mu_l\xi_l+\mu_l^*\xi_l^*) \end{aligned} \tag{4.18}$$

Then it can be solved by getting the saddle point:

$$\begin{aligned} \frac{\partial L}{\partial \Theta_S} = 0 \rightarrow \Theta_S &= \sum_{l=1}^{N}(\alpha_l-\alpha_l^*)\Phi_S(z_l) \\ \frac{\partial L}{\partial \xi_l} = 0 \rightarrow C &= \alpha_l+\mu_l \\ \frac{\partial L}{\partial \xi_l^*} = 0 \rightarrow C &= \alpha_l^*+\mu_l^* \end{aligned} \tag{4.19}$$

After transforming the Lagrangian function into its dual problem, we can get:

$$\begin{aligned} \max\ W(\alpha,\alpha^*) = &-\frac{1}{2}\sum_{l,j=1}^{N}(\alpha_l-\alpha_l^*)(\alpha_j-\alpha_j^*)K_S(z_l,z_j) \\ &+\sum_{l=1}^{N}(\alpha_l-\alpha_l^*)f(z)-\epsilon\sum_{l=1}^{N}(\alpha_l+\alpha_l^*) \\ \text{s.t.}\ \ &\sum_{l=1}^{N}(\alpha_l-\alpha_l^*)=0.\quad \alpha_l,\alpha_l^*\in[0,C]. \end{aligned} \tag{4.20}$$

where $K_S(z_l,z_j)$ is a data-dependent composed kernel called deep quasi-linear kernel, defined in a recursive form as:

$$\begin{aligned} K_i(z_l,z_j) &= {\Theta_i}^T(z_l)\Theta_i(z_j) \\ &= (1+K_{i-1}(z_l,z_j)){\mathbf{g}^{(i)}}^T(z_l)\mathbf{g}^{(i)}(z_j) \\ (i=1,...,S) & \end{aligned} \tag{4.21}$$

where $K_0(z_l,z_j)=z_l^Tz_j$.

Therefore, through substitute Eq.(4.19) and Eq.(4.21) into Eq.(4.16), the regression model can finally be represented as:

$$f(z) = \sum_{l=1}^{N} (\alpha_l - \alpha_l^*) K_S(z, z_l) + b \tag{4.22}$$

4.5 Experiments and Results

4.5.1 Experimental Setup

In this section, we use the air quality datasets and bank datasets to verify the effectiveness of the proposed model from two aspects. Firstly, we use air quality datasets [79] that originally have missing values to predict PM2.5. Except for original missing values contained in the datasets, we also analyze data to find outliers with values greater or less than three times the Interquartile Range (3IQR) [98], which should be removed from the dataset. Thus, the detailed description of features and the final condition of datasets are summarized in Table.4.1 and Table.4.2 respectively. The air quality datasets are recorded hourly and cover the year 2013-2017 from four stations of Beijing: Shunyi (SY), Huairou (HR), Changping (CP) and Tiantan (TT). The forecasting horizons are from T_1 to T_6, where T_n represents the n-th hour after the initial time. In the experiment, the PM2.5 prediction is regarded as the original prediction problem rather than time-series problem to verify the generality of our proposed model. Secondly, to verify the robustness of our proposed model with different range of missing values, another experiment based on Tiantan air quality dataset and two bank datasets [99] which have 8 attributes and 16 attributes respectively are conducted. Missing data generation is executed by inserting missing data at six missing rates randomly (10%, 20%, 30%, 40%, 50%, 60%).

For both experiments, 70% of the samples for each dataset are split into the training set, and the rest 30% belongs to the testing set. All the inputs are normalized to a range of

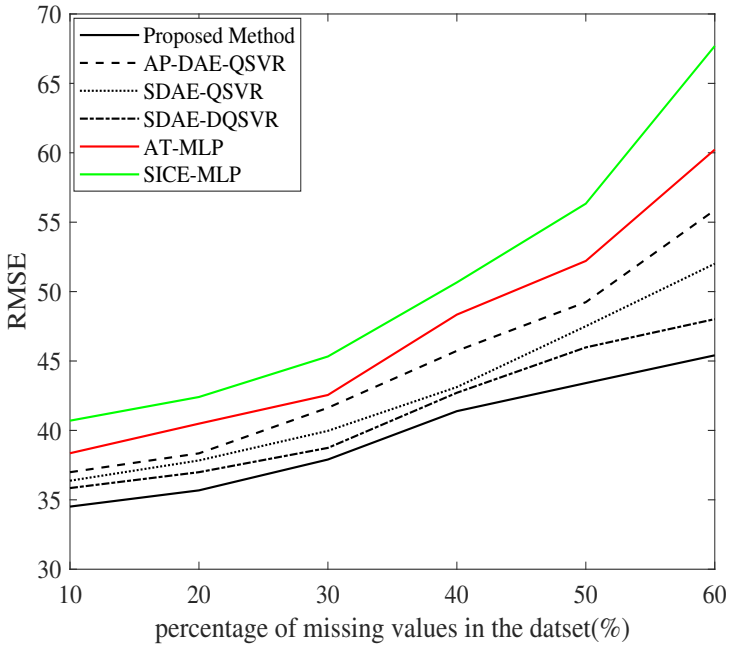


FIGURE 4.4: RMSE results for Tiantan dataset

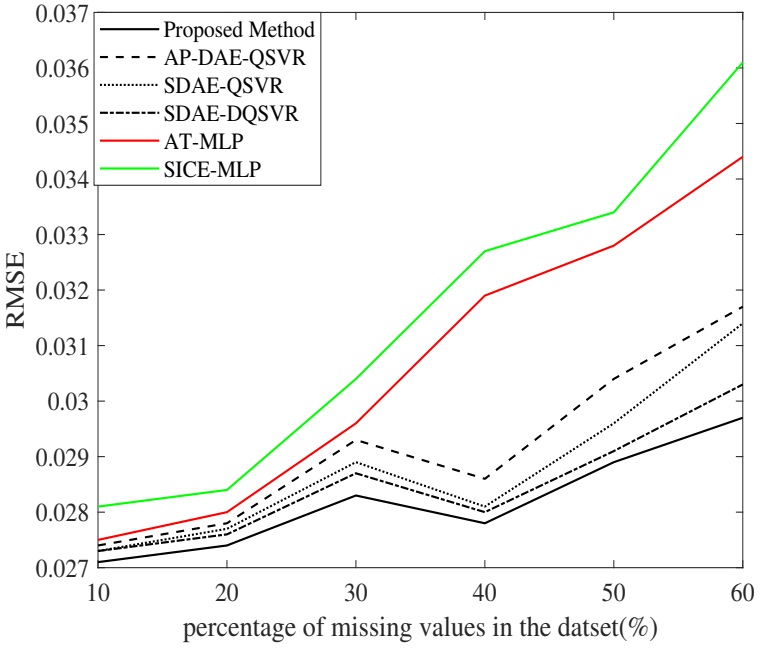


FIGURE 4.5: RMSE results for bank1 dataset

TABLE 4.1: The detailed description of features in air quality datasets

Features	Description
PM10	concentration ($\mu g/m^3$)
PM2.5	concentration ($\mu g/m^3$)
SO_2	concentration ($\mu g/m^3$)
NO_2	concentration ($\mu g/m^3$)
CO	concentration ($\mu g/m^3$)
0_3	concentration ($\mu g/m^3$)
TEMP	Temperature(degree Celsius)
PRES	Pressure (hPa)
DEWP	dew point temperature (degree Celsius)
RAIN	precipitation (mm)
WSPM	wind speed (m/s)

TABLE 4.2: Details of all four air quality datasets

Station	SY			HR			CP			TT		
Features	Miss	Mean	SD	Miss	Mean	SD	Miss	Mean	SD	Miss	Mean	SD
PM10	1435	90.372	71.759	820	84.408	68.442	1091	85.487	65.081	1196	95.757	68.546
SO_2	3678	9.287	10.318	3110	7.521	8.076	2687	10.269	10.864	3259	9.620	9.438
NO_2	1857	42.365	28.407	1637	30.618	23.082	1030	41.976	25.964	1104	50.643	27.917
CO	3877	988.768	729.554	2161	893.152	600.437	3645	900.688	591.180	2610	1080.016	720.589
0_3	1489	55.201	54.873	486	59.619	54.531	184	57.494	53.880	357	55.863	59.678
TEMP	51	13.388	11.483	51	12.337	11.749	53	13.585	11.376	20	13.667	11.493
PRES	51	1013.062	10.177	53	1007.677	10.007	50	1007.809	10.240	20	1012.547	10.246
DEWP	54	2.465	13.726	53	2.200	14.091	53	1.414	13.844	20	2.486	13.801
RAIN	51	0.061	0.762	55	0.068	0.854	51	0.061	0.761	20	0.064	0.788
WSPM	44	1.808	1.288	48	1.646	1.196	43	1.851	1.310	21	1.857	1.278

[0, 1], and the scaling formula is:

$$\bar{a}_i = \frac{a_{max} - a_i}{a_{max} - a_{min}} \tag{4.23}$$

where $\bar{a}_i$ represents the corresponding scaled value, a_i is the value of the i-th point, a_{min} and a_{max} is the minimum and maximum values of the dataset, respectively. Extra hyper-parameters are selected by a 5-fold cross-validation on the training set. The optimal learning rate of different datasets is chosen in the range of $\{1e\text{-}1, 1e\text{-}2, 1e\text{-}3\}$. For L_2 regularization term, weight decay is chosen within the grid $\{1e\text{-}3, 3e\text{-}3, 1e\text{-}2, 3e\text{-}2, 1e\text{-}1\}$. υ related with hint mechanism is chosen within $\{1, 2, 3\}$. For parameters related to the winner-take-all strategy, the sparsity level k in the feature layer is selected from $\{40, 45, 50\}$, and another sparsity level t is chosen within a grid $\{15, 20, 25, 30, 35\}$. Based on Section 4.2, the optimal parameter ζ in the loss function formula Eq.(4.10) is

TABLE 4.3: Prediction results for all four tested regression datasets

Datasets	Models	T_1	T_2	T_3	T_4	T_5	T_6
SY	Proposed Model	**29.026**±0.062	**29.717**±0.069	**30.734**±0.075	**31.276**±0.070	**32.899**±0.062	**34.026**±0.058
	AP-DAE-QSVR	30.957± 0.050	31.824±0.064	33.457±0.053	34.830±0.066	38.037±0.058	40.914±0.061
	SDAE-QSVR	30.475±0.069	31.095±0.071	32.909±0.077	33.914±0.074	36.752±0.069	38.423±0.079
	SDAE-DQSVR	30.128±0.071	30.893±0.083	31.788±0.077	32.668±0.080	34.318±0.085	36.972±0.090
	AT-MLP	31.203±0.139	33.652±0.150	35.485±0.162	37.813±0.134	41.691±0.140	46.111±0.146
	SICE-MLP	31.871±0.102	34.534±0.134	36.215±0.127	38.203±0.149	44.970±0.121	48.316±0.128
HR	Proposed Model	**21.520**±0.043	**21.849**±0.056	**23.335**±0.038	**25.891**±0.041	**27.985**±0.052	**30.696**±0.044
	AP-DAE-QSVR	23.075±0.037	24.531±0.058	27.098±0.044	30.421±0.040	36.637±0.059	40.016±0.048
	SDAE-QSVR	22.764±0.059	23.346±0.060	25.894±0.067	28.654±0.069	34.629±0.077	37.469±0.072
	SDAE-DQSVR	22.441±0.061	22.702±0.052	24.734±0.057	26.817±0.073	31.348±0.078	35.329±0.060
	AT-MLP	27.742±0.072	30.057±0.098	33.512±0.102	36.322±0.083	40.589±0.093	44.081±0.104
	SICE-MLP	25.871±0.061	32.713±0.080	35.340±0.093	38.119±0.095	42.405±0.120	47.108±0.114
CP	Proposed Model	**20.120**±0.027	**21.226**±0.030	**22.176**±0.035	**24.015**±0.042	**26.903**±0.037	**29.710**±0.038
	AP-DAE-QSVR	23.138±0.021	24.147±0.018	25.920±0.031	29.201±0.026	31.354 ±0.041	35.349±0.038
	SDAE-QSVR	21.615±0.037	22.556±0.055	24.398±0.048	27.211±0.040	29.898±0.039	32.073±0.052
	SDAE-DQSVR	21.041±0.040	22.487±0.056	24.014±0.052	26.851±0.055	28.234±0.043	30.328 ±0.061
	AT-MLP	25.857±0.069	27.469±0.097	30.875 ±0.086	33.519±0.094	37.058±0.082	41.554±0.086
	SICE-MLP	26.954±0.077	29.327±0.082	32.536±0.089	35.289±0.079	39.980±0.099	43.421±0.096
TT	Proposed Model	**20.906**±0.021	**21.352**±0.029	**22.978**±0.042	**25.257**±0.037	**28.043**±0.022	**31.218**±0.035
	AP-DAE-QSVR	22.312±0.020	23.804±0.048	25.981±0.037	31.117±0.024	33.178±0.037	38.299±0.031
	SDAE-QSVR	21.484±0.031	22.825±0.036	24.978±0.038	29.129±0.050	31.734±0.044	35.531±0.053
	SDAE-DQSVR	21.053±0.039	22.099±0.050	24.209±0.043	27.141±0.026	29.558±0.053	32.958±0.040
	AT-MLP	25.410±0.062	27.621±0.071	31.793±0.089	36.041±0.081	41.269±0.098	47.020±0.103
	SICE-MLP	27.048±0.073	29.712±0.074	33.357±0.087	39.628±0.080	43.534±0.097	49.416±0.105

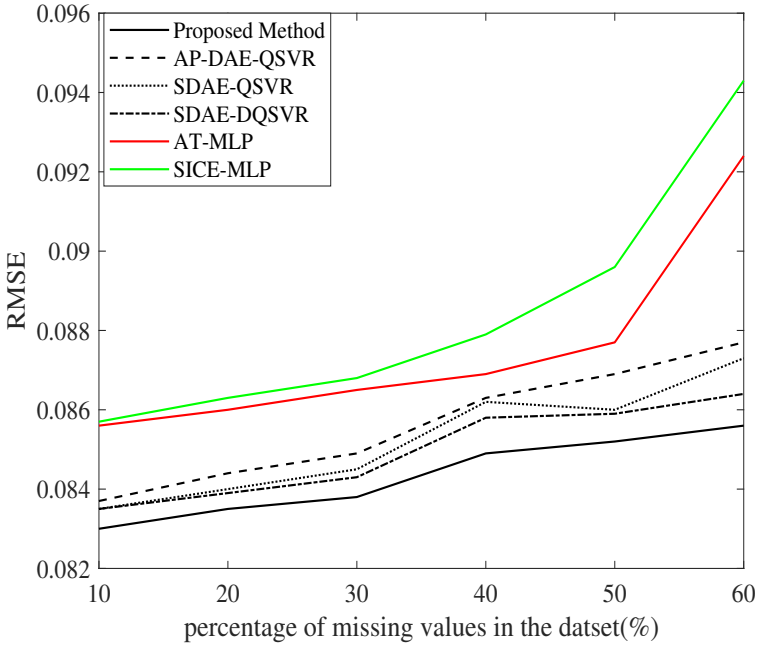


FIGURE 4.6: RMSE results for bank2 dataset

searched in the grid of $\{0.1, 0.2, 0.3\}$ at the beginning, and then ζ is chosen from $\{0.8, 0.9, 1.0\}$. In our experiments, we set the network size of the autoencoder as five layers, and the network size of the discriminator is set as one hidden layer. We use the pytorch, which is a python-based open source machine learning framework and our experiments are conducted with a $2.8GHz$ CPU.

In this paper, we apply RMSE to evaluate the performance of the proposed model, which is formulated as:

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i' - y_i)^2} \tag{4.24}$$

where y_i' is the predicted data of corresponding y_i, n is the number of the elements in testing part. The value of RMSE close to 0 means the superiority of the prediction model.

Since initialized weights of the generative model are random at each run, various new datasets can be generated as posterior predictive distributions of missing values. In the experiment, we repeat the hybrid model for the prespecified number of times $D = 20$ and then calculate the standard deviation to verify the stability of the model.

TABLE 4.4: Prediction results based different percentages of missing values

Datasets	Models	Missing					
		10%	20%	30%	40%	50%	60%
TT	proposed method	**34.512**±0.070	**35.681**±0.081	**37.903**±0.087	**41.387**±0.099	**43.410**±0.116	**45.409**±0.125
	AP-DAE-QSVR	36.990±0.060	38.347±0.063	41.629±0.075	45.733 ±0.091	49.239±0.102	55.871±0.121
	SDAE-QSVR	36.377±0.067	37.836±0.085	39.969±0.093	43.122±0.101	47.507±0.116	52.008±0.122
	SDAE-DQSVR	35.847±0.067	36.991±0.086	38.741±0.092	42.711±0.100	45.981±0.119	48.017±0.130
	AT-MLP	38.356±0.088	40.487±0.097	42.552±0.107	48.342±0.113	52.208±0.120	60.230±0.128
	SICE-MLP	40.702±0.091	42.407±0.093	45.326±0.110	50.661±0.114	56.337±0.122	67.694±0.129
Bank1	Proposed Method	**0.0271**±0.0002	**0.0274**±0.0003	**0.0283**±0.0004	**0.0278**±0.0004	**0.0289**±0.0005	**0.0297**±0.0005
	AP-DAE-QSVR	0.0274±0.0001	0.0278±0.0002	0.0293±0.0003	0.0286±0.0003	0.0304±0.0004	0.0317±0.0004
	SDAE-QSVR	0.0273±0.0002	0.0277±0.0003	0.0289±0.0003	0.0281±0.0004	0.0296±0.0005	0.0314±0.0005
	SDAE-DQSVR	0.0273±0.0003	0.0276±0.0004	0.0287±0.0005	0.0280±0.0005	0.0291±0.0006	0.0303±0.0007
	AT-MLP	0.0275±0.0004	0.0280±0.0004	0.0296± 0.0006	0.0319±0.0007	0.0328±0.0009	0.0344±0.0011
	SICE-MLP	0.0281±0.0004	0.0284±0.0004	0.0304±0.0006	0.0327±0.0009	0.0334±0.0010	0.0361±0.0013
Bank2	Proposed Method	**0.0830**±0.0002	**0.0835**±0.0002	**0.0838**±0.0003	**0.0849**±0.0005	**0.0852**±0.0005	**0.0856**±0.0006
	AP-DAE-QSVR	0.0837±0.0001	0.0844±0.0001	0.0849±0.0003	0.0863±0.0004	0.0869±0.0004	0.0877±0.0005
	SDAE-QSVR	0.0835±0.0001	0.0840±0.0002	0.0845±0.0004	0.0862±0.0004	0.0860±0.0006	0.0873±0.0007
	SDAE-DQSVR	0.0835±0.0003	0.0839±0.0003	0.0843±0.0005	0.0858±0.0006	0.0859±0.0006	0.0864±0.0008
	AT-MLP	0.0856±0.0004	0.0860±0.0005	0.0865±0.0007	0.0869±0.0008	0.0877±0.0009	0.0924±0.0009
	SICE-MLP	0.0857±0.0005	0.0863±0.0005	0.0868±0.0006	0.0879±0.0008	0.0896±0.0011	0.0943±0.0014

In this paper, we compare our proposed model with persistence methods from three aspects. Since in the literature, many researchers regard the nonlinear regression problem with missing values as two separate problems, so we first combine the adversarial training method with multi-layer perception (AT-MLP) which is still a popular method for solving the nonlinear regression problem [100] to address the effectiveness of SVR with deep quasi-linear kernel. Secondly, the performance of the adversarial training method is examined, the improved version of prevalent multiple imputation method MICE named Single Center Imputation from Multiple Chained Equation(SICE) [101] with MLP (SICE-MLP) are conducted in the experiments. Thirdly, we also compare the proposed model with our two previously published models as references. The first AP-DAE-QSVR model was developed in Ref.[90] by combing affinity propagation clustering algorithm, denoising autoencoders, and a piecewise linear regression model with interpolations. The second SDAE-QSVR model was proposed in Ref.[95], which consists of two parts: an overcomplete WTA autoencoder for estimating missing values and generating gate control sequences, and a binary gated linear network for implementing a piecewise linear model. Besides, the novel SDAE method proposed in Ref.[95] combined with SVR with deep quasi-linear kernel (SDAE-DQSVR) is also tested to convince the effectivenss of deep quasi-linear kernel.

4.5.2 Performance Evaluation

The experimental results of PM2.5 prediction are shown in Table.4.3. For easy comparison, the first-rank model is highlighted with boldface. From Table.4.3, we can draw the following five conclusions. Firstly, we can clearly see that the proposed model achieves the best performance in each case. To specify, with the increase of prediction interval, the advantages of our proposed model are becoming more obvious. Secondly, comparing the results of AT-MLP method, we can conclude that SVR with the deep quasi-linear kernel outperforms prevalent MLP method. Thirdly, we make a comparison between the proposed model and SICE-MLP method to convince that the adversarial training method has led to the greater performance for tackling missing data problem. Finally,

compared SDAE-QSVR method with SDAE-DQSVR method from the T_1 to T_6 moments, DQSVR is more stable and robust in dealing with nonlinear prediction problems. For instance, the growth rate between T_1 and T_6 is 26.1% for the SDAE-QSVR method and 22.7% for the SDAE-DQSVR method in the SY dataset.

Furthermore, as Table.4.4 and Fig.4.4-4.6 show, our proposed model yields the best prediction accuracy with a wide range of missing values. Fig.4.4-4.6 show that with the increase of missing data, the RMSE value of our proposed model grows the slowest. Therefore, it leads to the conclusion that the proposed model is more stable and effective in dealing with the regression problem of missing data.

4.6 Conclusions

In this chapter, we propose a hybrid model to solve the severe problem: nonlinear regression under missing data scenarios. This modelling method consists of two parts: the overcomplete WTA autoencoder, which is trained in an adversarial training process for imputing missing components conditioned on observed samples and designing layered gated control sequences, and the multilayer gated linear network with generated gate control sequences for implementing the piecewise linear regression model. In this way, we can implicitly optimize the parameters of the piecewise linear model by applying an SVR formulation with the deep quasi-linear kernel. Various experiments have shown that our proposed model is effective and robust even in dealing with large proportion of missing data.

Chapter 5

Conclusions

5.1 Summary

This final chapter concludes the overall thesis. In this thesis, three different hybrid modeling methods are proposed to solve nonlinear regression problems under the missing data. All of these models fully leverage the information of datasets during the filling-in missing values step and parameterize the information as prior knowledge to generate local-linear models or piecewise linear models. Therefore, our proposed models are more effective and robust even in dealing with large proportions of missing values.

chapter 2 proposes a hybrid model consisting of an autoencoder and a gated linear network. AP clustering method with iterations is firstly presented as a preprocessing method aimed at providing teacher signals and cluster information to construct a competitive net. Then, we present a multiple imputation method based on winner-take-all DAEs. Finally, an advanced modification named SVR with the quasi-linear kernel is used to design a gated linear network based on each partition. The following are the main conclusions of this chapter:

- It is the first to present a hybrid model which combines winner-take-all DAE and a gated linear network to solve forecasting problems with missing data.

- To the best of our knowledge, our studies on filling in missing data before training DAEs firstly use the AP clustering algorithm for constructing the self-organized competitive net, which makes DAEs get more accurate and effective information during training, and as to the regression phase, we can also make full use of cluster information efficiently to build a local linear prediction model.

Chapter 3 proposes a winner-take-all (WTA) autoencoder-based piecewise linear model, which consists of two steps. WTA SDAEs are employed to tackle missing data problems, and then a gated linear network is designed to construct a piecewise linear model. In this way, SVR with a data-dependent quasi-linear kernel can be used for prediction tasks. The main contributions of this chapter are shown as follows:

- By increasing the role of the denoising autoencoder, we use the SDAE to fill in the missing values. Morevoer, an iterative algorithm is developed to improve the performance of the SDAE.
- SDAEs can realize a sophisticated partitioning instead of the AP clustering method by generating a broad set of binary gate control sequences using the feature laye. By using the binary gate control sequences, the gated linear network implements a flexible piecewise linear model for the nonlinear regression.

Chapter 4 develops a hybrid model to solve nonlinear regression under missing data scenarios. This modelling method consists of two parts: the overcomplete WTA autoencoder, which is trained in an adversarial training process for imputing missing components conditioned on observed samples and designing layered gated control sequences, and the multilayer gated linear network with generated gate control sequences for implementing the piecewise linear regression model. In this way, we can implicitly optimize the parameters of the piecewise linear model by applying an SVR formulation with the deep quasi-linear kernel. The main contributions of this chapter are shown as follows:

- We use adversarial training process to improve the accuracy of filling in missing values.

- A piecewise linear regression model is built through the multi-layer gated linear network.

5.2 Future Research of Topics

Though much progress has been made in this thesis, there are still two main fields that need to be further investigated. Firstly, since the accuracy of teacher signals which are the input of autoencoders plays an very important role in filling in missing data, we should consider methods to improve the accuracy of teacher signals. Secondly, more applications should be considered in the future like automatic control systems.

Bibliography

[1] P. H. Sherrod, "Nonlinear regression analysis program," *Nashville, TN, USA*, 2005.

[2] H.-H. Huang, C. K. Hsiao, S.-Y. Huang, P. Peterson, E. Baker, and B. McGaw, "Nonlinear regression analysis," *International encyclopedia of education*, pp. 339–346, 2010.

[3] I. N. Soyiri and D. D. Reidpath, "An overview of health forecasting," *Environmental health and preventive medicine*, vol. 18, no. 1, pp. 1–9, 2013.

[4] A. R. T. Donders, G. J. Van Der Heijden, T. Stijnen, and K. G. Moons, "A gentle introduction to imputation of missing values," *Journal of clinical epidemiology*, vol. 59, no. 10, pp. 1087–1091, 2006.

[5] A. N. Baraldi and C. K. Enders, "An introduction to modern missing data analyses," *Journal of school psychology*, vol. 48, no. 1, pp. 5–37, 2010.

[6] J. L. Schafer and J. W. Graham, "Missing data: our view of the state of the art." *Psychological methods*, vol. 7, no. 2, p. 147, 2002.

[7] M. S. Santos, R. C. Pereira, A. F. Costa, J. P. Soares, J. Santos, and P. H. Abreu, "Generating synthetic missing data: A review by missing mechanism," *IEEE Access*, vol. 7, pp. 11 651–11 667, 2019.

[8] S. P. Caro, S. V. Schaper, R. A. Hut, G. F. Ball, and M. E. Visser, "The case of the missing mechanism: how does temperature influence seasonal timing in endotherms?" *PLoS Biology*, vol. 11, no. 4, p. e1001517, 2013.

[9] D. F. Heitjan and S. Basu, "Distinguishing "missing at random" and "missing completely at random"," *The American Statistician*, vol. 50, no. 3, pp. 207–213, 1996.

[10] S. Seaman, J. Galati, D. Jackson, and J. Carlin, "What is meant by "missing at random"?" *Statistical Science*, vol. 28, no. 2, pp. 257–268, 2013.

[11] C. K. Enders, "Missing not at random models for latent growth curve analyses." *Psychological methods*, vol. 16, no. 1, p. 1, 2011.

[12] P. E. McKnight, K. M. McKnight, S. Sidani, and A. J. Figueredo, *Missing data: A gentle introduction*. Guilford Press, 2007.

[13] H. Zhu, S.-Y. Lee, B.-C. Wei, and J. Zhou, "Case-deletion measures for models with incomplete data," *Biometrika*, vol. 88, no. 3, pp. 727–737, 2001.

[14] M. Soley-Bori, "Dealing with missing data: Key assumptions and methods for applied analysis," *Boston University*, vol. 23, p. 20, 2013.

[15] Z. Zhang, "Missing data imputation: focusing on single imputation," *Annals of translational medicine*, vol. 4, no. 1, 2016.

[16] C. D. Newgard and R. J. Lewis, "Missing data: how to best account for what is not known," *Jama*, vol. 314, no. 9, pp. 940–941, 2015.

[17] P. J. García-Laencina, J.-L. Sancho-Gómez, A. R. Figueiras-Vidal, and M. Verleysen, "K nearest neighbours with mutual information for simultaneous classification and missing data imputation," *Neurocomputing*, vol. 72, no. 7-9, pp. 1483–1493, 2009.

[18] R. R. Andridge and R. J. Little, "A review of hot deck imputation for survey non-response," *International statistical review*, vol. 78, no. 1, pp. 40–64, 2010.

[19] D. W. Joenssen and U. Bankhofer, "Hot deck methods for imputing missing data," in *International workshop on machine learning and data mining in pattern recognition*. Springer, 2012, pp. 63–75.

[20] D. Sullivan and R. Andridge, "A hot deck imputation procedure for multiply imputing nonignorable missing data: The proxy pattern-mixture hot deck," *Computational Statistics & Data Analysis*, vol. 82, pp. 173–185, 2015.

[21] S. Z. Christopher, T. Siswantining, D. Sarwinda, and A. Bustaman, "Missing value analysis of numerical data using fractional hot deck imputation," in *2019 3rd International Conference on Informatics and Computational Sciences (ICICoS)*. IEEE, 2019, pp. 1–6.

[22] R. Khincha, U. Sarawgi, W. Zulfikar, and P. Maes, "Robustness to missing features using hierarchical clustering with split neural networks," *arXiv preprint arXiv:2011.09596*, 2020.

[23] D. Li, J. Deogun, W. Spaulding, and B. Shuart, "Towards missing data imputation: a study of fuzzy k-means clustering method," in *International conference on rough sets and current trends in computing*. Springer, 2004, pp. 573–579.

[24] J. T. Chi, E. C. Chi, and R. G. Baraniuk, "k-pod: A method for k-means clustering of missing data," *The American Statistician*, vol. 70, no. 1, pp. 91–99, 2016.

[25] J. L. Besay Montesdeoca, J. Maillo, D. Garcıa-Gil, S. Garcıa, and F. Herrera, "A first approach on big data missing values imputation," *Conference: 4th International Conference on Internet of Things, Big Data and Security*, 2019.

[26] Z. Zhang, H. Fang, and H. Wang, "Multiple imputation based clustering validation (miv) for big longitudinal trial data with missing values in ehealth," *Journal of medical systems*, vol. 40, no. 6, p. 146, 2016.

[27] L. Malan, C. M. Smuts, J. Baumgartner, and C. Ricci, "Missing data imputation via the expectation-maximization algorithm can improve principal component analysis aimed at deriving biomarker profiles and dietary patterns," *Nutrition Research*, vol. 75, pp. 67–76, 2020.

[28] W.-C. Lin and C.-F. Tsai, "Missing value imputation: a review and analysis of the literature (2006–2017)," *Artificial Intelligence Review*, vol. 53, no. 2, pp. 1487–1509, 2020.

[29] L. H. Rubin, K. Witkiewitz, J. S. Andre, and S. Reilly, "Methods for handling missing data in the behavioral neurosciences: Don't throw the baby rat out with the bath water," *Journal of Undergraduate Neuroscience Education*, vol. 5, no. 2, p. A71, 2007.

[30] O. Delalleau, A. Courville, and Y. Bengio, "Efficient EM training of gaussian mixtures with missing data," *arXiv preprint arXiv:1209.0521*, 2012.

[31] G. Madhu, B. L. Bharadwaj, G. Nagachandrika, and K. S. Vardhan, "A novel algorithm for missing data imputation on machine learning," in *2019 International Conference on Smart Systems and Inventive Technology (ICSSIT)*. IEEE, 2019, pp. 173–177.

[32] X. Chai, H. Gu, F. Li, H. Duan, X. Hu, and K. Lin, "Deep learning for irregularly and regularly missing data reconstruction," *Scientific reports*, vol. 10, no. 1, pp. 1–18, 2020.

[33] A. F. Costa, M. S. Santos, J. P. Soares, and P. H. Abreu, "Missing data imputation via denoising autoencoders: the untold story," in *International Symposium on Intelligent Data Analysis*. Springer, 2018, pp. 87–98.

[34] Q. Ma, W.-C. Lee, T.-Y. Fu, Y. Gu, and G. Yu, "Midia: exploring denoising autoencoders for missing data imputation," *Data Mining and Knowledge Discovery*, vol. 34, no. 6, pp. 1859–1897, 2020.

[35] S. Ryu, M. Kim, and H. Kim, "Denoising autoencoder-based missing value imputation for smart meters," *IEEE Access*, vol. 8, pp. 40 656–40 666, 2020.

[36] J. Yoon, J. Jordon, and M. Schaar, "Gain: Missing data imputation using generative adversarial nets," in *International Conference on Machine Learning*. PMLR, 2018, pp. 5689–5698.

[37] T. E. Raghunathan, J. M. Lepkowski, J. Van Hoewyk, P. Solenberger *et al.*, "A multivariate technique for multiply imputing missing values using a sequence of regression models," *Survey methodology*, vol. 27, no. 1, pp. 85–96, 2001.

[38] S. Van Buuren and K. Groothuis-Oudshoorn, "mice: Multivariate imputation by chained equations in R," *Journal of statistical software*, vol. 45, pp. 1–67, 2011.

[39] E. A. Stuart, M. Azur, C. Frangakis, and P. Leaf, "Multiple imputation with large data sets: a case study of the children's mental health initiative," *American journal of epidemiology*, vol. 169, no. 9, pp. 1133–1139, 2009.

[40] Y. He, A. M. Zaslavsky, M. Landrum, D. Harrington, and P. Catalano, "Multiple imputation in a large-scale complex survey: a practical guide," *Statistical methods in medical research*, vol. 19, no. 6, pp. 653–670, 2010.

[41] O. I. Abiodun, A. Jantan, A. E. Omolara, K. V. Dada, N. A. Mohamed, and H. Arshad, "State-of-the-art in artificial neural network applications: A survey," *Heliyon*, vol. 4, no. 11, p. e00938, 2018.

[42] S. Palani, S.-Y. Liong, and P. Tkalich, "An ANN application for water quality forecasting," *Marine pollution bulletin*, vol. 56, no. 9, pp. 1586–1597, 2008.

[43] A. R. Pazikadin, D. Rifai, K. Ali, M. Z. Malik, A. N. Abdalla, and M. A. Faraj, "Solar irradiance measurement instrumentation and power solar generation forecasting based on Artificial Neural Networks (ANN): A review of five years research trend," *Science of The Total Environment*, vol. 715, p. 136848, 2020.

[44] D. Voukantsis, K. Karatzas, J. Kukkonen, T. Räsänen, A. Karppinen, and M. Kolehmainen, "Intercomparison of air quality data using principal component analysis, and forecasting of PM10 and PM2.5 concentrations using artificial neural networks, in thessaloniki and helsinki," *Science of the Total Environment*, vol. 409, no. 7, pp. 1266–1276, 2011.

[45] M. M. H. Khan, N. S. Muhammad, and A. El-Shafie, "Wavelet-ANN versus ANN-based model for hydrometeorological drought forecasting," *Water*, vol. 10, no. 8, p. 998, 2018.

[46] Y. Bengio, *Learning deep architectures for AI*. Now Publishers Inc, 2009.

[47] A.-r. Mohamed, T. N. Sainath, G. Dahl, B. Ramabhadran, G. E. Hinton, and M. A. Picheny, "Deep belief networks using discriminative features for phone

recognition," in *2011 IEEE international conference on acoustics, speech and signal processing (ICASSP)*. IEEE, 2011, pp. 5060–5063.

[48] L. Zhang, Z. Shi, M.-M. Cheng, Y. Liu, J.-W. Bian, J. T. Zhou, G. Zheng, and Z. Zeng, "Nonlinear regression via deep negative correlation learning," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 43, no. 3, pp. 982–998, 2021.

[49] J. Fox and S. Weisberg, "Nonlinear regression and nonlinear least squares," 2002.

[50] G. Ferrari-Trecate and M. Muselli, "A new learning method for piecewise linear regression," in *International conference on artificial neural networks*. Springer, 2002, pp. 444–449.

[51] J. Cervantes, F. Garcia-Lamont, L. Rodríguez-Mazahua, and A. Lopez, "A comprehensive survey on support vector machine classification: Applications, challenges and trends," *Neurocomputing*, vol. 408, pp. 189–215, 2020.

[52] A. J. Smola and B. Schölkopf, "A tutorial on support vector regression," *Statistics and computing*, vol. 14, no. 3, pp. 199–222, 2004.

[53] J. A. Suykens, T. Van Gestel, J. De Brabanter, B. De Moor, and J. P. Vandewalle, *Least squares support vector machines*. World scientific, 2002.

[54] M. Welling, "Kernel ridge regression," *Max Welling's classnotes in machine learning*, pp. 1–3, 2013.

[55] V. Gosasang, W. Chandraprakaikul, and S. Kiattisin, "An application of neural networks for forecasting container throughput at bangkok port," in *Proceedings of the world congress on engineering*, vol. 1, 2010, pp. 2078–0958.

[56] A. Sherstinsky, "Fundamentals of recurrent neural network (RNN) and long short-term memory (LSTM) network," *Physica D: Nonlinear Phenomena*, vol. 404, p. 132306, 2020.

[57] X. Wang, K. Wang, J. Ding, X. Chen, Y. Li, and W. Zhang, "Predicting water quality during urbanization based on a causality-based input variable selection

method modified back-propagation neural network," *Environmental Science and Pollution Research*, vol. 28, no. 1, pp. 960–973, 2021.

[58] B.-C. Kuo, H.-H. Ho, C.-H. Li, C.-C. Hung, and J.-S. Taur, "A kernel-based feature selection method for SVM with RBF kernel for hyperspectral image classification," *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol. 7, no. 1, pp. 317–326, 2013.

[59] F. Javed, G. S. Chan, A. V. Savkin, P. M. Middleton, P. Malouf, E. Steel, J. Mackie, and N. H. Lovell, "Rbf kernel based support vector regression to estimate the blood volume and heart rate responses during hemodialysis," in *2009 annual international conference of the IEEE engineering in medicine and biology society*. IEEE, 2009, pp. 4352–4355.

[60] B. Zhou, B. Chen, and J. Hu, "Quasi-linear support vector machine for nonlinear classification," *IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences*, vol. 97, no. 7, pp. 1587–1594, 2014.

[61] W. Li, P. Liang, and J. Hu, "An autoencoder-based piecewise linear model for nonlinear classification using quasilinear support vector machines," *IEEJ Transactions on Electrical and Electronic Engineering*, vol. 14, no. 8, pp. 1236–1243, 2019.

[62] W. Li, B. Zhou, B. Chen, and J. Hu, "A deep neural network based quasi-linear kernel for support vector machines," *IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences*, vol. 99, no. 12, pp. 2558–2565, 2016.

[63] P. Liang, W. Li, and J. Hu, "Oversampling the minority class in a multi-linear feature space for imbalanced data classification," *IEEJ Transactions on Electrical and Electronic Engineering*, vol. 13, no. 10, pp. 1483–1491, 2018.

[64] I. Yilmaz and O. Kaynar, "Multiple regression, ANN (RBF, MLP) and ANFIS models for prediction of swell potential of clayey soils," *Expert systems with applications*, vol. 38, no. 5, pp. 5958–5966, 2011.

[65] A. Krogh, B. Larsson, G. Von Heijne, and E. L. Sonnhammer, "Predicting transmembrane protein topology with a hidden markov model: application to complete genomes," *Journal of molecular biology*, vol. 305, no. 3, pp. 567–580, 2001.

[66] T. Liu, H. Wei, and K. Zhang, "Wind power prediction with missing data using gaussian process regression and multiple imputation," *Applied Soft Computing*, vol. 71, pp. 905–916, 2018.

[67] Y. Xiang, L. Gou, L. He, S. Xia, and W. Wang, "A SVR–ANN combined model based on ensemble EMD for rainfall prediction," *Applied Soft Computing*, vol. 73, pp. 874–883, 2018.

[68] W. Wang, Z. Xu, W. Lu, and X. Zhang, "Determination of the spread parameter in the gaussian kernel for classification and regression," *Neurocomputing*, vol. 55, no. 3-4, pp. 643–663, 2003.

[69] J. Van Hulse and T. M. Khoshgoftaar, "Incomplete-case nearest neighbor imputation in software measurement data," *Information Sciences*, vol. 259, pp. 596–610, 2014.

[70] E. Eirola, A. Lendasse, V. Vandewalle, and C. Biernacki, "Mixture of Gaussians for distance estimation with missing data," *Neurocomputing*, vol. 131, pp. 32–42, 2014.

[71] F. Scheuren, "Multiple imputation: How it began and continues," *The American Statistician*, vol. 59, no. 4, pp. 315–319, 2005.

[72] Y. Bengio, L. Yao, G. Alain, and P. Vincent, "Generalized denoising auto-encoders as generative models," in *Advances in neural information processing systems*, 2013, pp. 899–907.

[73] J. Hu, K. Kumamaru, and K. Hirasawa, "A Quasi-ARMAX approach to modelling of non-linear systems," *International Journal of Control*, vol. 74, no. 18, pp. 1754–1766, 2001.

[74] Y. Cheng, L. Wang, and J. Hu, "Identification of quasi-ARX neurofuzzy model with an SVR and GA approach," *IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences*, vol. 95, no. 5, pp. 876–883, 2012.

[75] K. Wang, J. Zhang, D. Li, X. Zhang, and T. Guo, "Adaptive affinity propagation clustering," *arXiv preprint arXiv:0805.1096*, 2008.

[76] B. J. Frey and D. Dueck, "Clustering by passing messages between data points," *science*, vol. 315, no. 5814, pp. 972–976, 2007.

[77] Q. Yu, Y. Miche, E. Eirola, M. Van Heeswijk, E. SéVerin, and A. Lendasse, "Regularized extreme learning machine for regression with missing data," *Neurocomputing*, vol. 102, pp. 45–51, 2013.

[78] A. Makhzani and B. J. Frey, "Winner-take-all autoencoders," in *Advances in neural information processing systems*, 2015, pp. 2791–2799.

[79] http://www.archive.ics.uci.edu/ml/datasets.html.

[80] https://tianchi.aliyun.com/home/.

[81] R. J. Little and D. B. Rubin, *Statistical analysis with missing data*. John Wiley & Sons, 2019, vol. 793.

[82] C. Robert and G. Casella, *Monte Carlo statistical methods*. Springer Science & Business Media, 2013.

[83] L. Wang, D. Fu, Q. Li, and Z. Mu, "Modelling method with missing values based on clustering and support vector regression," *Journal of Systems Engineering and Electronics*, vol. 21, no. 1, pp. 142–147, 2010.

[84] D. Sovilj, E. Eirola, Y. Miche, K.-M. Björk, R. Nian, A. Akusok, and A. Lendasse, "Extreme learning machine for missing data using multiple imputations," *Neurocomputing*, vol. 174, pp. 220–231, 2016.

[85] P. Liang, W. Li, Y. Wang, and J. Hu, "One-class classification using quasi-linear support vector machine," in *2018 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*. IEEE, 2018, pp. 662–667.

[86] J. L. Schafer and M. K. Olsen, "Multiple imputation for multivariate missing-data problems: A data analyst's perspective," *Multivariate behavioral research*, vol. 33, no. 4, pp. 545–571, 1998.

[87] X. Dong, L. Lin, R. Zhang, Y. Zhao, D. C. Christiani, Y. Wei, and F. Chen, "TOBMI: trans-omics block missing data imputation using a k-nearest neighbor weighted approach," *Bioinformatics*, vol. 35, no. 8, pp. 1278–1283, 2019.

[88] N. Abiri, B. Linse, P. Edén, and M. Ohlsson, "Establishing strong imputation performance of a denoising autoencoder in a wide range of missing data problems," *Neurocomputing*, vol. 365, pp. 137–146, 2019.

[89] M. J. Azur, E. A. Stuart, C. Frangakis, and P. J. Leaf, "Multiple imputation by chained equations: what is it and how does it work?" *International journal of methods in psychiatric research*, vol. 20, no. 1, pp. 40–49, 2011.

[90] H. Zhu, Y. Tian, Y. Ren, and J. Hu, "A hybrid model for nonlinear regression with missing data using quasi-linear kernel," *IEEJ Transactions on Electrical and Electronic Engineering*, vol. 15, no. 12, pp. 1791–1800, 2020.

[91] M. R. Malarvizhi and A. S. Thanamani, "K-nearest neighbor in missing data imputation," *International Journal of Engineering Research and Development*, vol. 5, no. 1, pp. 5–7, 2012.

[92] S. Hua-Yan, L. Ye-Li, Z. Yun-Fei, and H. Xu, "Accelerating EM missing data filling algorithm based on the k-means," in *2018 4th Annual International Conference on Network and Information Systems for Computers (ICNISC)*. IEEE, 2018, pp. 401–406.

[93] P. Vincent, H. Larochelle, I. Lajoie, Y. Bengio, P.-A. Manzagol, and L. Bottou, "Stacked denoising autoencoders: Learning useful representations in a deep network with a local denoising criterion." *Journal of machine learning research*, vol. 11, no. 12, 2010.

[94] K. P. Singh, S. Gupta, A. Kumar, and S. P. Shukla, "Linear and nonlinear modeling approaches for urban air quality prediction," *Science of the Total Environment*, vol. 426, pp. 244–255, 2012.

[95] H. Zhu, Y. Ren, Y. Tian, and J. Hu, "A winner-take-all autoencoder based pieceswise linear model for nonlinear regression with missing data," *IEEJ Transactions on Electrical and Electronic Engineering*, 2021.

[96] A. Makhzani and B. Frey, "Winner-take-all autoencoders," *arXiv preprint arXiv:1409.2752*, 2014.

[97] P. Liang, W. Li, H. Tian, and J. Hu, "One-class classification using a support vector machine with a quasi-linear kernel," *IEEJ Transactions on Electrical and Electronic Engineering*, vol. 14, no. 3, pp. 449–456, 2019.

[98] P. Sakul-Ung, P. Ruchanawet, N. Thammabunwarit, A. Vatcharaphrueksadee, C. Triperm, and M. Sodanil, "PM2. 5 prediction based weather forecast information and missingness challenges: A case study industrial and metropolis areas," in *2019 Research, Invention, and Innovation Congress (RI2C)*. IEEE, 2019, pp. 1–5.

[99] ⟨https://www.dcc.fc.up.pt/~ltorgo/Regression/DataSets.html⟩.

[100] Z. Car, S. Baressi Šegota, N. Anelić, I. Lorencin, and V. Mrzljak, "Modeling the spread of COVID-19 infection using a multilayer perceptron," *Computational and mathematical methods in medicine*, vol. 2020, 2020.

[101] S. I. Khan and A. S. M. L. Hoque, "SICE: an improved missing data imputation technique," *Journal of big data*, vol. 7, no. 1, pp. 1–21, 2020.

Publication List

Journal Papers

J1. H.Zhu and J. Hu, "An Improved Hybrid Model for Nonlinear Regression with Missing Values Using Deep Quasi-Linear Kernel", *IEEJ Trans. on Electrical and Electronic Engineering*, Vol.17, No.10, PP.1-9, 2022.

J2. Y. Ren, H. Jiang, H.Zhu, Y. Tian and J. Hu, "A Semi-Supervised Classification Method of Parasites Using Contrastive Learning", *IEEJ Trans. on Electrical and Electronic Engineering*, Vol.17, No.3, pp.445-453, March 2022.

J3. H.Zhu,Y. Ren, Y. Tian and J. Hu, "A Winner-Take-All Autoencoder Based Piecewise Linear Model for Nonlinear Regression with Missing Data", *IEEJ Trans. on Electrical and Electronics Engineering*, Vol.16, No.12, pp.1618-1627, Dec 2021.

J4. Y. Ren, H.Zhu, Y. Tian and J. Hu, "A Laplacian SVM Based Semi-Supervised Classification Using Multi-Local Linear Model", *IEEJ Trans. on Electrical and Electronic Engineering*, Vol.16, No.3, pp.455-463, March 2021.

J5. H. Zhu, Y. Tian, Y. Ren and J. Hu, "A Hybrid Model for Nonlinear Regression with Missing Data Using Quasi-Linear Kernel", *IEEJ Trans. on Electrical and Electronics Engineering*, Vol.15, No.12, pp.1791-1800, Dec 2020.

Proceeding Papers

P1. H. Zhu, Y. Ren and J. Hu, "Establishing a Hybrid Piecewise Linear Model for Air Quality Prediction Based Missingness Challenges", in *Proc. of 2021 IEEE International Conference on System, Man, and Cybernetics* (SMC 2021) (Melbourne), pp.1705-1710, Oct 2021.

P2. Y. Ren, H. Deng, H. Jiang, H. Zhu and J. Hu, "A Semi-Supervised Classification Method of Apicomplexan Parasites and Host Cell Using Contrastive Learning Strategy", in *Proc. of 2021 IEEE International Conference on System, Man, and Cybernetics* (SMC 2021) (Melbourne), pp.2973-2978, Oct 2021.

P3. H. Zhu and J. Hu, “Air Quality Forecasting using SVR with Quasi-Linear Kernel”, in *Proc. of the 2019 International Conference on Computer*, Information and Telecommunication Systems (CITS 2019) (Bejing), pp.1-5, August 2019.