An Integrated Data Preparation Scheme for Neural Network Data Analysis

Lean Yu, Shouyang Wang, and K.K. Lai

Abstract—Data preparation is an important and critical step in neural network modeling for complex data analysis and it has a huge impact on the success of a wide variety of complex data analysis tasks, such as data mining and knowledge discovery. Although data preparation in neural network data analysis is important, some existing literature about the neural network data preparation are scattered, and there is no systematic study about data preparation for neural network data analysis. In this study, we first propose an integrated data preparation scheme as a systematic study for neural network data analysis. In the integrated scheme, a survey of data preparation, focusing on problems with the data and corresponding processing techniques, is then provided. Meantime, some intelligent data preparation solution to some important issues and dilemmas with the integrated scheme are discussed in detail. Subsequently, a cost-benefit analysis framework for this integrated scheme is presented to analyze the effect of data preparation on complex data analysis. Finally, a typical example of complex data analysis from the financial domain is provided in order to show the application of data preparation techniques and to demonstrate the impact of data preparation on complex data analysis.

Index Terms—Data preparation, neural networks, complex data analysis, cost-benefit analysis.

1 Introduction

Preparing data is an important and critical step in neural network modeling for complex data analysis and it has an immense impact on the success of a wide variety of complex data analysis, such as data mining and knowledge discovery [1]. The main reason is that the quality of the input data into neural network models may strongly influence the results of the data analysis [2]. As Lou [3] stated, the effect on the neural network's performance can be significant if important input data are missing or distorted. In general, properly prepared data are easy to handle, which makes the data analysis task simple. On the other hand, improperly prepared data may make data analysis difficult, if not impossible. Furthermore, data from different sources and growing amounts of data produced by modern data acquisition techniques have made data preparation a time-consuming task. It has been claimed that 50-70 percent of the time and effort in data analysis projects is required for data preparation [2], [4]. Therefore, data preparation involves enhancing the data in an attempt to improve complex data analysis.

In past decades, artificial neural networks (ANNs), as a class of typical intelligent data analysis tool, have been studied extensively in many fields of knowledge, from science (e.g., [5]) to engineering (e.g., [6]) and from management (e.g., [7]) to control (e.g., [8]), and many software products, such as *NeuroShell* (http://www. neuroshell.com), *BrainMaker* (http://www.calsci.com), and *Neural Network Toolbox of Matlab* (http://www.mathworks. com), have been applied successfully in many practical projects. However, most studies and commercial systems focus almost exclusively on the design and implementation of neural models. Data preparation in neural network modeling has received scant recognition. In almost all theoretical and practical researches about neural networks [5], [6], [7], [8], [9], [10], [11], [12], [13], [14], [15], [16], [17], [18], neural network data preparations concentrate on data normalization for transformation and data division for training. Some studies even utilize neural networks for modeling without any data preparation procedure. In these studies, there is an implicit assumption that all the data are prepared in advance and the data can be used directly in modeling. In practice, data are not always prepared beforehand for specific data analysis tasks. Although there are some data for a specific project, the quality and completeness of that data is limited. As a result, the complex data analysis process cannot succeed without a serious effort to prepare the data. Strong evidence [18], [19], [20] reveals that data quality has a significant effect on neural network models. In addition, various interpretations have been given to the role and the need for data preparation. Zhang et al. [21] revealed that data preparation could generate smaller magnitude and higher quality data, which can significantly improve the efficiency of complex data analysis. In the case of neural network learning, data preparation would enable users to decide how to represent the data, which concepts to learn, and how to present the results of data analysis so that it is easier to explain them in the real world [22]. Data preparation is therefore crucial in neural network data analysis for guaranteeing data quality and completeness.

Although data preparation is useful for any kind of analysis, neural networks, as a class of important intelligent data analysis tool, have some special requirements for data preparation. First of all, neural networks are a kind of novel learning paradigm, but it is a time-consuming data analysis tool, which is different from any other data analysis tool. To

- *L. Yu and S. Wang are with the Institute of Systems Science, Academy of Mathematics and Systems Science, Chinese Academy of Sciences, Beijing, 100080, China. E-mail: {yulean, sywang}@amss.ac.cn.*
- *K.K. Lai is with the Department of Management Sciences, City University of Hong Kong, Hong Kong and College of Business Administration, Hunan University, China. E-mail: mskklai@city.edu.hk.*

Manuscript received 20 Nov. 2004; revised 30 Mar. 2005; accepted 14 July 2005; published online 19 Dec. 2005.
For information on obtaining reprints of this article, please send e-mail to: tkde@computer.org, and reference IEEECS Log Number TKDESI-0460-1104.

speed up the analysis process, data preparation is very important and necessary for complex data analysis tasks. Second, data preparation can largely reduce model complexity for neural network modeling, which is important for complex data analysis tasks relative to other data analysis tools. Third, effective data preparation can increase generalization ability of data analysis, especially for neural networks. Basically, the main objective of complex data analysis is to discovery knowledge that will be used to solve problems or make decisions, but problems about the data may prevent this. In most cases, imperfections with the data are not noticed until the data analysis starts. This is the case, especially for the neural networks. Therefore, data preparation in neural networks is more important than that of other data analysis tool.

However, in many neural network data preparation studies [23], [24], [25], [26], [27], [28], [29], [30], [31], data preparation is restricted to data cleaning, normalization, and division. In their studies, most authors considered that data preprocessing is equivalent to data preparation. Actually, the two terms are different. First of all, they have different significances. According to the American Heritage Dictionary [32], "preparation" is defined as "a preliminary measure that serves to make ready for something," while "preprocessing" is "to perform conversion, formatting, or other functions on (data) before further processing." Second, basic contents covered are different. Data preparation covers more contents than data preprocessing. Generally, data preprocessing only includes data transformation and data formatting, while data preparation contains more contents, such as data collection, data selection, data integration, and data validation in addition to data preprocessing. In our study, data preparation is expanded into an integrated scheme with three phases from a systematic perspective. This integrated scheme is comprised of a data preanalysis phase, including data collection, data selection and data integration, and a data postanalysis phase, such as data validation and data readjustment, as well as a data preprocessing phase. In this sense, our study goes well beyond previous studies [23], [24], [25], [26], [27], [28], [29], [30], [31].

In the overview of this topic, we found that there are three main problems in neural network data preparation. The first is that there is no universal scheme or methodology for neural network data analysis (Problem I). That is, there is no perfect and systematic data preparation framework or architecture for neural network data analysis. Although some related studies are presented, a systematic work about neural network data preparation has not been formulated so far. Most existing studies focused on the data preprocessing. Furthermore, these researchers often confused the difference between data preparation and data preprocessing. Therefore, it is necessary to construct a universal data preparation scheme for neural network data analysis. Second, in preparing data for neural network data analysis, some important issues and dilemmas are often faced and are hard to handle (Problem II). For example, skilled experts may find a good solution, but may also find it difficult to judge whether the preparation chosen is appropriate. Third, data preparation requires extra effort, raising the question of cost versus benefits (Problem III). If they see data preparation as having little impact on final neural network data analysis results, decision-makers may be unwilling to invest in data preparation.

In light of the three problems outlined above, the main motivations of this study are four-fold:

1. to propose an integrated data preparation framework for neural network data analysis,
2. to provide some intelligent solutions to some important issues and dilemmas in the data preparation framework,
3. to analyze and confirm the effects of the proposed data preparation scheme on a neural network data analysis, and
4. to survey the literature about data preparation of neural network data analysis.

The study first proposes an integrated data preparation scheme for neural network modeling to contribute to the solution of the first main problem and then presents, in detail, alternative solutions to the dilemmas of the data preparation framework. For the third problem, a cost-benefit analysis framework for neural network data preparation is proposed. However, empirical evidence of the impact of data preparation on complex data analysis is critical. Without loss of generality, we explored the effects of data preparation on data analysis within a specific problem domain—the business financial risk classification area. This is a vital research field with a vast number and variety of important data sets.

The remainder of the study is organized as follows: A brief description of neural network models for complex data analysis is presented in Section 2. In view of the neural network data analysis framework, an integrated data preparation scheme for neural networks is proposed to fill up the gap in the literature. Meanwhile, the steps in every phase, as well as important issues of the integrated scheme, are described and discussed in detail. Accordingly, some intelligent solutions to some important issues and dilemmas are provided in Section 3. A comprehensive cost-benefit analysis framework for analyzing the effect of the proposed integrated data preparation scheme on neural network data analysis is proposed in Section 4. To verify the effects of data preparation on neural network data analysis, an empirical example is presented in Section 5. Finally, the paper ends with concluding remarks and future directions in Section 6.

2 NEURAL NETWORKS FOR COMPLEX DATA ANALYSIS

The foundation of the artificial neural networks (ANNs) paradigm was laid in the 1950s. Since then, ANNs have earned significant attention because of the development of more powerful hardware and neural algorithms [9]. ANNs have been studied and explored by many researchers and been applied and manipulated in almost every field, examples include system identification and modeling [10] and prediction and classification [11], [12], [13], [14], [15], [16], [17] Generally, ANNs can be used as an effective intelligent data analysis tool for their unique learning

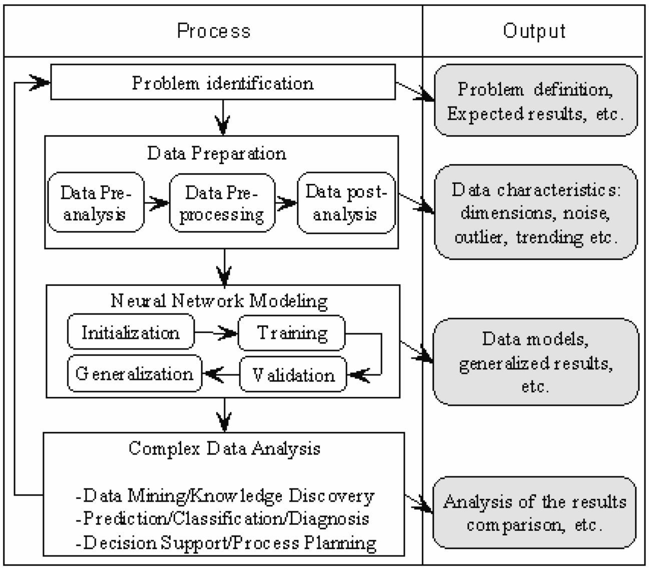
Process
Output
Problem identification
Problem definition, Expected results, etc.
Data Preparation
Data Pre-analysis
Data Pre-processing
Data post-analysis
Data characteristics: dimensions, noise, outlier, trending etc.
Neural Network Modeling
Initialization
Training
Generalization
Validation
Data models, generalized results, etc.
Complex Data Analysis
-Data Mining/Knowledge Discovery
-Prediction/Classification/Diagnosis
-Decision Support/Process Planning
Analysis of the results comparison, etc.

Fig. 1. The process of neural network data analysis.

capability. In this section, an entire process of neural network data analysis is presented, as shown in Fig. 1.

As can be seen from Fig. 1, neural network modeling for complex data analysis has four main processes: problem identification, data preparation, neural network modeling, and data analysis. In the first process, we can identify a problem by analyzing its expected results and consulting the relevant domain experts. Problem definitions and expected results are formulated to guide the subsequent tasks. The aim of the second process, data preparation, which will be described later, is to prepare high-quality data for data analysis so as to obtain satisfactory results. In the third process, after initialization, neural network models are trained iteratively. If the results of the data validation are rational, the generalized results obtained from the trained networks can be used for data analysis. Finally, depending on the generalized results, the goal of the complex data analysis, such as data mining and decision support, can be realized.

3 The Proposed Integrated Data Preparation Scheme

In this section, we first propose an integrated data preparation scheme based on the entire process of neural network data analysis framework. We then present details of the scheme and overview some related literature. Subsequently, some intelligent solutions to some important issues and dilemmas in the integrated scheme are presented.

3.1 The Integrated Data Preparation Scheme for Neural Network Data Analysis

As noted earlier, neural network data preparation for complex data analysis is very important. However, no standard data preparation framework for neural network modeling has so far been suggested (Problem I). In view of the importance of data preparation, we propose an integrated data preparation scheme for neural network modeling, as illustrated in Fig. 2.

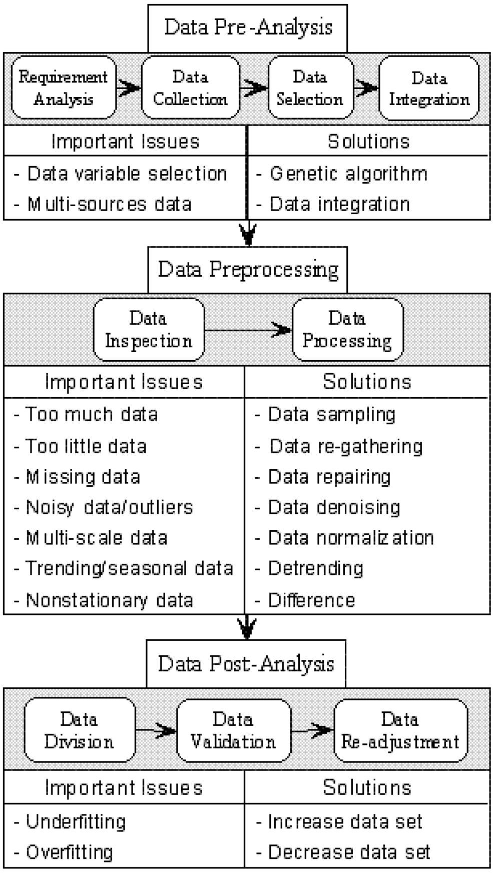

Fig. 2. The integrated data preparation scheme for neural network data analysis.

As is shown in Fig. 2, the integrated data preparation scheme consists of three phases: data preanalysis, in which data of interest are identified and collected; data preprocessing, in which data are examined and analyzed and in which some data may be restructured or transformed to make them more useful; and data postanalysis, in which some data are validated and readjusted. In the integrated data preparation scheme, every phase is comprised of different processing steps. For example, the data preanalysis phase includes data requirement analysis, data collection, data selection, and data integration. Data preprocessing is comprised of data inspection and data processing. Data postanalysis contains data division and redivision, data validation, and data readjustment. This phase is called "postanalysis" because the data preparation tasks may be adjusted in terms of feedback information in the process of neural network training, learning, and validation (i.e., modeling process). Because the postanalysis adjusts the data for modeling purpose, the data postanalysis is still seen as the range of data preparation. In almost all existing studies, data preparation includes only the second phase, data preprocessing. Therefore, our proposed data preparation scheme is broader than others, which distinguishes our study from others. This is an important contribution that can fill up the gap in the literature.

In the following sections, the three phases of the integrated data preparation scheme are described in detail. First, the detailed steps of every phase and some data problems that are normally encountered are discussed and some existing methods and techniques to overcome these problems are overviewed step-by-step. For some important issues and dilemmas (Problem II), they are described and some rational intelligent solutions presented.

3.2 Data PreAnalysis of the Integrated Data Preparation Scheme

As seen in Fig. 2, this phase consists of four steps: data requirement analysis, data selection, data collection, and data integration.

3.2.1 Data Requirement Analysis

For a specific data analysis project, the first step is to understand the data requirements of the project in conjunction with the problem definitions and expected objectives. If the problem is outside one's field of expertise, interviewing specialists or domain experts may provide insight into the underlying process so that some potential problems may be avoided [23]. Questions may include the following:

1. What information would we like to have?
2. What data are required for a specific task?
3. Where can the data be found?
4. What format are the data in?
5. What external sources of data are available?

When understanding the data requirements, data will be collected from various sources.

3.2.2 Data Collection

This is an important step because the outcome of the step will restrict subsequent phases or steps. Based on the data requirements, all kinds of approaches, such as information retrieval [33] and text mining [34], will be used to collect various data from various sources. In some situations, some important data may be hard to collect. Thus, surrogate data is useful and necessary.

3.2.3 Data Variable Selection

Once data are collected, determining variables for modeling becomes possible. The goal of any model should be parsimony, i.e., to find the simplest explanation of the facts using the fewest variables. Therefore, it is best to identify the variables that will save modeling time and reduces the problem space [23]. There are many means of variable selection [35], [36], [37], [38], [39]. For example, Lemke and Muller [35] used a modular approach and self-organizing variable selection to realize variable reduction, while Tuv and Runger [36] presented a nonhierarchical metric clustering method to deal with high-dimensional classification. Some other methods, such as correlation analysis with the Granger causality method [37], principal component analysis (PCA) [38], and stepwise multiple regression [39] are also mentioned in the literature.

3.2.4 Data Integration

If data are collected from many different sources by several different groups, the data are still in disorder and scattered, and data integration treatment becomes vital [22]. This is especially true when data contain text and symbolic attributes and have to be combined for further analysis.

In general, data sources can be divided into internal and external sources [40]. Similarly, data representations can be divided roughly into structural representation and non structural representation. Therefore, there are different data integration methods for different data representations from multisources. Regarding structural data from different sources, we can utilize mature database techniques, such as virtual views and data warehouse techniques [41], to integrate data relations from different sources via join and/or union operations. Semantic and descriptive conflicts can be solved by renaming operations and conversions. Some metalevel conflicts and instance-level conflicts can be solved by advanced schema transformation (e.g., transposition of relations), reconciliation function, and user-defined aggregates. More information can be found in [41], [42]. With regard to the integration of nonstructural data from different sources, [43] and [44] present some solutions.

3.2.5 Important Issues and Dilemmas of this Phase

In this phase, two important issues are described and discussed in detail.

1. **Important Issue I: Data Variable selection with genetic algorithms.** From the previous analysis, we find that data variable selection is an extremely important issue and many related studies [35], [36], [37], [38], [39] are presented. Here, we present an intelligent solution—genetic algorithm (GA)—to this important issue for neural network data analysis. To date, genetic algorithms (GAs) have become a popular optimization method as they often succeed in finding the best optimum in contrast to most common optimization algorithms. Genetic algorithms imitate the natural selection process in biological evolution with selection, mating reproduction and mutation, and the sequence of the different operations of a genetic algorithm is shown in the left part of Fig. 3. The parameters to be optimized are represented by a chromosome whereby each parameter is encoded in a binary string called gene. Thus, a chromosome consists of as many genes as parameters to be optimized. Interested readers can be referred to [45], [46] for more details. In the following, GA for data variable selection is discussed.

   First of all, a population, which consists of a given number of chromosomes, is initially created by randomly assigning "1" and "0" to all genes. In the case of variable selection, a gene contains only a single bit string for the presence and absence of a variable. The top right part of Fig. 3 shows a population of four chromosomes for a three-variable selection problem. In this study, the initial population of the GA is randomly generated, except for one chromosome, which was set to use all variables. The binary string of the chromosomes has the same size as variables to select from whereby the presence of a variable is coded as "1" and the absence of a variable as "0." Consequently, the binary string of a gene

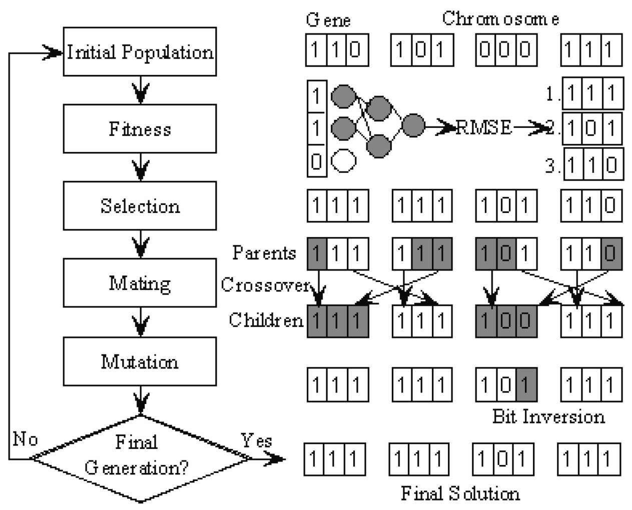

Fig. 3. The data varaible selection with the gentic algorithm.

consists of only one single bit. The subsequent work is to evaluate the chromosomes generated by previous operation by a so-called fitness function, while the design of the fitness function is a crucial point in using GA, which determines what a GA should optimize. In the case of a variable selection for neural network data analysis, the goal is to find a small subset of variables, which are most significant for complex data analysis. In this study, the complex data analysis is based on neural networks for modeling the relationship between the input variables and the responses. Thus, the evaluation of the fitness starts with the encoding of the chromosomes into neural networks whereby "1" indicates that a specific variable is used and "0" that a variable is not used by the network. Then, the networks are trained with a training data set and, after that, a testing data set is predicted. Finally, the fitness is calculated by a so-called fitness function f. For a prediction/classification problem, for example, our fitness function for the GA variable selections can use the following form:

$$\begin{aligned} f = 0.3RMSE_{training} + 0.7RMSE_{testing} \\ - \alpha(1 - n_v/n_{tot}), \end{aligned} \tag{1}$$

where n_v is the number of variables used by the neural networks, n_{tot} is the total number of variables, and *RMSE* is the root mean square error, which is defined in (2) with N as total number of samples predicted, y_t as the actual value and $\hat{y}_t$ as the predicted value:

$$RMSE = \sqrt{\frac{1}{N}\sum\nolimits_{t=1}^{N} (\hat{y}_t - y_t)^2}. \tag{2}$$

From (1), we find that the fitness function can be broken up into three parts. The first two parts correspond to the accuracy of the neural networks. Thereby, $RMSE_{training}$ is based on the prediction of the training data used to build the neural nets, whereas $RMSE_{testing}$ is based on the prediction of separate test data not used for training the neural networks. It was demonstrated in [47] that using the same data for the variable selection and for the model calibration introduces a bias. Thus, variables are selected based on data poorly representing the true relationship. On the other hand, it was also shown that a variable selection based on a small data set is unlikely to find an optimal subset of variables [47]. Therefore, a ratio of 3:7 between the influence of training and testing data was chosen. Although being partly arbitrary, this ratio should give as little influence to the training data as to bias the feature selection yet taking the samples of the larger training set partly into account. The third part of the fitness function rewards small networks using only few variables by an amount proportional to the parameter a. The choice of an influence is the number of variables used by the evolved neural nets. A high value of results is in only a few variables selected for each GA, whereas a small value results in more variables being selected. In sum, the advantage of this fitness function is that it takes into account not only the testing error of test data but also partially the training error and primarily the number of variables used to build the corresponding neural nets.

After evolving the fitness of the population, the best chromosomes with the highest fitness value are selected by means of the roulette wheel. The chromosomes are thereby allocated space on a roulette wheel proportional to their fitness and, thus, the fittest chromosomes are more likely selected. In the following mating step, offspring chromosomes are created by a crossover technique. A so-called one-point crossover technique is employed, which randomly selects a crossover point within the chromosome. Then, two parent chromosomes are interchanged at this point to produce two new offspring. After that, the chromosomes are mutated with a probability of 0.005 per gene by randomly changing genes from "0" to "1" and vice versa. The mutation prevents the GA from converging too quickly in a small area of the search space. Finally, the final generation will be judged. If yes, then the optimized subsets are selected. If no, then the evaluation and reproduction steps are repeated until a certain number of generations, until a defined fitness, or until a convergence criterion of the population are reached. In the ideal case, all chromosomes of the last generation have the same genes representing the optimal solution.

2. **Important Issue II: The integration of nonstructural data.** Another important issue is the integration of nonstructural data. Nonstructural data consists of unstructured data and semistructural data. As earlier noted, integrating nonstructural data from different sources is difficult. Here, a three-phase approach for this task is proposed.

The first phase is to extract related features from text data or documents by semantic analysis and formulate an event-specific summary. This extraction makes nonstructural data more readable and representative. Some string matching algorithms, such as [48], [49], can be used. The second phase is to transform the summary into corresponding nominal variables or numerical variables by classification algorithms, such as [49]. This transformation makes modeling easier because the transformed variables can be treated as dummy variables in models. The last phase is to tabulate the transformed variables, making them more easily used by models. Non structural data can thereby be formulated into a single data set using previous data integration techniques.

3.3 Data Preprocessing Phase of the Integrated Data Preparation Scheme

After the data are identified and collected, they must be examined to identify any characteristics that may be unusual or indicative of more complex relationship. This is because the data from the previous phase may be impure, divergent, untrustworthy, or even fraudulent. Therefore, data preprocessing is required. In this study, data preprocessing is a transformation, or conditioning, of data designed to make modeling more robust, which include data inspection and processing.

3.3.1 Data Inspection

The first step of data preprocessing is data inspection. The goal of data inspection is to find problems with data. Data inspection includes data quantity and quality inspection. The former is to check the size of data sets and the latter any unusual data set patterns. The data quantity inspection can be performed by observation. Generally, there are two main problems with this: a too-large data size or a too-small data size. The data quality inspection can also be performed by statistical methods including the check of data noise, data missing, data scale, and data trending and data nonstationarity. There are four approaches to data quality inspection: line graph for checking missing data and data trending, control plot [50] for data noise detection, unit root test [51] for data nonstationarity check, and SVM-OD [52] for outlier detection.

3.3.2 Data Processing

The above step (i.e., data inspection) can identify seven main problems: too many data, too few data, noisy data including outliers and errors, missing data, multiscale data, trending (or seasonal) data, and nonstationary data. Accordingly, several processing techniques—data sampling [53], [54], data regathering, data denoising [52], [55], [56], [57], data repairing [4], [58], [59], [60], [61], [62], [63], data normalization [3], [64], and data difference [64], [65], [66]—are used to deal with them.

3.3.3 Important Issues and Dilemmas of This Phase

In this section, six important issues and two dilemmas about data preprocessing are presented.

1. **Important Issue I: Too many data and data sampling.** In many domains, such as space (e.g., image data) and finance (e.g., stock price data every five minutes), the volume of data and the rate at which data are produced may be a limiting factor in performing on-time data analysis. Furthermore, the amount of data is sometimes beyond the capability of the hardware and software available for data analysis. Therefore, sample space reduction is important. Here, clustering and data discretization are used to treat the problem.

   If there are a large number of observations, i.e., a large sample size, a useful approach is to obtain a representative subset of data or a data sampling. An effective way of doing this is to divide the sample by forming clusters of sample observations. Every cluster can then be represented by one observation. This can be 1) one specific observation, 2) the mean value of all observations in the cluster, and 3) observation which has the lowest distance from all the others, and so on. In addition, data sampling techniques [53], [54], which selects a representative subset from a large population of data, can also be used.

   If data clustering is difficult, discretization can be used. Discretization is aimed at reducing the number of distinct values for a given attribute, particularly for analysis methods requiring discrete attribute values. Possible methods of discretization are 1) histogram-based discretization, 2) discretization based on concept-hierarchies [53], and 3) entropy-based discretization [54]. Here, we focus on histogram-based discretization due to its simplicity.
2. **Important Issue II: Too few data and data regathering.** Conversely, if too few data are collected, data regathering will be necessary for complex data analysis. Nguyen and Chan [31] found that neural networks perform worse when few data are available or data are insufficient. As data regathering may be difficult, all kinds of information channels and gathering tools should be used for this task.
3. **Important Issue III: Noisy data and data denoising.** As has been stated [22], noise in the data weakens the predictive capability of the features. Therefore, noise reduction or data denoising is very important for neural network data analysis. In the existing literature, noise elimination has been extensively studied [52], [55], [56], [57]. Here, we propose a regression-based data denoising approach to eliminate the effect of noise.

   In our approach, the first step in noise reduction is noise detection. We use a control plot to detect the noise, as previously mentioned. The second step in noise reduction is noise filtering to eliminate outliers. Here, we use regression technique. In linear regression, also known as least square method, the goal is to find a straight line modeling a two-dimensional data set. This line $y = \alpha x + \beta$ is specified by the parameters α and β, which are calculated from the known values of the attribute x and y. Let

   $$\bar{x} = \sum x_i/n \text{ and } \bar{y} = \sum y_i/n, \tag{3}$$

   then

$$\alpha = \sum (x_i - \bar{x})(y_i - \bar{y}) \Big/ \sum (x_i - \bar{x})^2 \text{ and } \\ \beta = \bar{y} - \alpha\bar{x}. \tag{4}$$

The parameters α and β can now be used to remove data items well away from the regression line. For example, this can be decided simply based on the absolute distance or by removing n percent of items with the largest distance as noise or outliers.

4. **Important Issue IV: Missing data and data repairing.** Roughly, missing data can be divided into two types: missing attributes and missing attribute values. Missing or insufficient attributes are examples of data problems that may complicate data analysis tasks, such as learning, and hinder accurate performance of most data analysis systems [22]. For example, in the case of learning, these data insufficiencies limit the performance of a learning algorithm or statistical tool applied to the collected data, no matter how complex the algorithm is or how many data are used. Furthermore, missing attributes are a source of too few data, as was previously revealed. Therefore, related attribute data should be further regathered.

   However, in most practical applications, an important problem is the handling of missing attribute values in a data set. Several studies have been done on dealing with missing values with numerous methods (see [4], [58], [59], [60], [61], [62], [63]). The aim of these methods is to recover the missing values that are as close as possible to the original values. The methods of doing this can be categorized into two types: imputation-based and data mining-based methods. The former is primarily for handling missing values of numerical data, while the latter is for category data. The principle of imputation methods is to estimate the missing values by using the existing values as an auxiliary base. The underlying assumption is that there are certain correlations between different data tuples over all attributes. The existing methods include mean imputation [58], hot-deck or cold-deck imputation [59], regression, and composite imputation [60]. For the data mining-based methods, techniques such as associations [61], clustering [62], and regression [63] are used to discover similar patterns between data tuples so as to predict the missing values.

5. **Important Issue V: Multiscale data and data normalization.** In neural network learning, data with different scales often lead to the instability of neural networks [64]. At the very least, data must be scaled into the range used by the input neurons in the neural network. This is typically -1 to 1 or zero to 1 [3]. Many commercially available generic neural network development programs, such as *BrainMaker*, automatically scale each input. Moreover, neural networks always require that the range of the data is neither too small nor too large, so that the precision limits of the computer are not exceeded. Otherwise, the data should be scaled. Furthermore, data normalization helps to improve the performance of neural networks [3]. Therefore, data normalization is necessary for treating multiscale data. The main reason is that the neural network models often rely on Euclidean measures, and unscaled data could bias or interfere with the training process. Line scaling and sigmoidal function normalization are the commonly used methods.

   The line scaling method is a simple and effective approach. Let the maximal and minimal value of input range be I_{max} and I_{min}. Then, the formula for transforming each data D to an input value I is:

$$I = I_{\min} + [(I_{\max} - I_{\min}) \times (D - D_{\min})]/(D_{\max} - D_{\min}), \tag{5}$$

   where D_{max} and D_{min} are the maximal and minimal value of a given input range. This method of normalization will scale input data into the appropriate range.

   In addition, a logistic function can be used as a data normalization method, depending on the characteristics of the data. Here, a sigmoidal function is utilized in the following:

$$I(x) = \frac{r_i}{1 + \exp[-p_i(x_i - q_i)]}, i = 1, 2, \cdots, n, \tag{6}$$

   where r_i is used to constrain the range for the *i*th transformed element of the *n*-element data set, q_i can be selected as the smallest in the *i*th element of the data set, and p_i decides the sharpness of the transferred function. Also, (6) can compress the abnormal data to a specific range. Note that different continuous and differentiable transformation function can also be selected.

6. **Important Issue VI: Trending data, seasonal data, and nonstationary data.** For a neural predictor, the presence of a trend may have undesired effects on the prediction performance [64]. Similarly, researchers [65], [66] have demonstrated that seasonal data have a significant impact on neural network prediction. As to univariate time series analysis with neural networks, nonstationarity is a problem for time-series analysis [66]. Therefore, data detrending and deseasonalization and data stationarity are also important issues in complex data analysis. For trending and seasonal data and nonstationary data, difference or log-difference [64], [65], [66] is a simple and effective treatment method that is widely used.

7. **Dilemma I: Data sampling and sample representative trade-off.** In this phase, the first dilemma is a data sampling size and sample representative trade-off problem. Generally, as a larger data sample is taken, the variability of the sample tends to fluctuate even less between the smaller and larger samples. To resolve the trade-off problem, this study presents a novel convergence approach.

   The convergence approach has two types: incremental and decremental. In the incremental type, a random sample is first selected and the distribution properties (such as mean, standard deviation, skewness, and kurtosis) are calculated. Then, the sample distribution is tested repeatedly by adding additional instances. If the sample distribution is recalculated as each additional instance is added, a low number of instances will appear in the sample; that is, each addition will make a large impact on the shape of the curve. However, when the number of instances in the sample is modest, the overall shape of the curve will settle down and will change little as new instance values are added. This settling down of the overall curve is the key to deciding the "convergence" between two different data sets. The decremental method is the opposite of the incremental method.

8. **Dilemma II: Noise and nonstationarity trade-off.** The second dilemma of this phase is the so-called "noise-nonstationarity trade-off [66]" for neural network univariate time-series models. That is, when there is noise and nonstationarity in the time series at the same time, neural network training on older data sets (longer training window) can induce biases in predictions because of nonstationarity, whereas using a shorter training window can increase estimation error (too much model variance) because of the noise in the limited data set. Moody [66] suggested using the testing error against the training window length in the choice of the optimal training window. We followed this suggestion.

3.4 Data Postanalysis Phase of the Integrated Data Preparation Scheme

3.4.1 Data Division and Redivision

Following data preprocessing, data obtained from the previous phase is used for network training and generalization. The first main data preparation task in this phase is to split data into subsets for neural network learning. Usually, a data set is divided into training data and testing data (sometimes, there is a third data set—a validation set). So far, there is no universal rule to determine the size of either a training data set or a testing data set. Brainmaker software randomly selects 10 percent of the facts from the data set and uses them for testing. Yao and Tan [67] suggested that historical data be divided into three sets: training, validation, and testing. The training set contains 70 percent of the collected data, while the validation and the testing sets contain 20 percent and 10 percent, respectively. Sometimes, through feedback of modeling results, data redivision is required.

3.4.2 Data Validation

Generally, the training error always decreases with an increase in the number of cycles or epochs. In contrast, the testing error does not have a continuously decreasing trend where a minimum value is found on the curve. Thus, there are two classes of problems (overfitting and underfitting) to disturb the network. In overfitting, a network usually performs worse instead of better after a certain point during training. This is because such long training may make the network memorize the training patterns, including all of their peculiarities. Underfitting results from insufficient training. This makes the network's generalization very poor. The solution to two problems is to validate the data with a section of extra data by cross-validation.

3.4.3 Data Readjustment

Depending on the feedback of data validation and model training results, data readjustment is often necessary. Different feedbacks mean different adjustments. For example, if the training result of neural network is not satisfactory, data redivision may be necessary. If there is overfitting or underfitting in the data validation, data redivision is also required. After data adjustment is completed, new learning begins once more.

3.4.4 Important Issues and Dilemmas of this Phase

In this phase, an important issue and two dilemmas are described in the following.

1. **Important Issue I: Overfitting, underfitting, and model complexity.** Neural networks are often referred to as universal function approximators since, theoretically, any continuous function can be approximated to a prescribed degree of accuracy by increasing the number of neurons in the hidden layer of a feedforward network [68]. Yet, in reality, the objective of a data analysis (e.g., prediction) is not to approximate a data set with an ultimate accuracy, but to find a suitable model with the best possible generalizing ability [69]. The gap between the approximation of a data set and the model generalization ability becomes the more problematic the higher the number of variables and the smaller the data set, which will be explained below.

   For a prediction problem, the best measure for the generalizing ability is the prediction error of as many independent separate validation data as possible. According to the left side of Fig. 4, the prediction error is composed of two main contributions, the remaining interference error and the estimation error [70]. The interference error is the systematic error (bias) due to unmodeled interference in the data, as the data analysis model (e.g., a prediction model) is not complex enough to capture all the interferences of the relationship. The estimation error is caused by modeling measured random noise of various kinds. The optimal prediction is obtained when the remaining interference error and the estimation error balance each other, as shown in Fig. 4. The effect of the prediction error increasing due to a too simple model is called underfitting, whereas the effect of the increased prediction error due to a too complex model is called overfitting.

   In the right side of Fig. 4, it is shown that the optimal complexity of the model highly depends on the size and quality of the data set. For data sets which are noisy and limited in size, a simple model is needed to prevent the overfitting. Neural networks which are too complex (too big), are in danger of learning these data by heart and, consequently, model noise of the data. For big data sets, which contain only a little noise, the best model is more

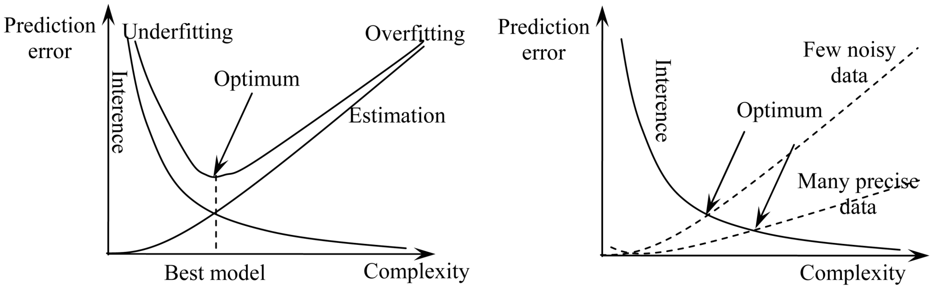

Fig. 4. Overfitting, underfitting, and model complexity.

complex, resulting in an overall smaller prediction error for the same functional relationship. Consequently, for each data set, an optimal model complexity has to be found whereby the complexity of the models is directly related with the number of data variables utilized by the model.

2. **Dilemma I: Training set size and model fitting.** In this phase, the first dilemma is the training set size and model fitting dilemma, that is, how to determine the rational training data size. Generally, training data size is not too large or too small. A too large training set may lead to a long training time and slower training speed, particularly when the entire training set is presented to a network between weights updates and even lead to overfitting. When a training set size is too small, the network cannot learn effectively; this leads to underfitting and weak generalization. To solve this problem, cross-validation, such as *k*-fold and leave-one-out, can be used.
3. **Dilemma II: Training epochs and network generalization.** Another dilemma is the number of training epochs or cycles and network generalization. Usually, if the number of training cycles increases, the training error rate should decrease. The error rate on test cases should begin to decrease and then it will eventually turn upward. This corresponds to the dynamic of underfitting and then overfitting. So far, there is no universal rule to determine training epochs, except trial and error with incremental algorithms.

A key issue of our study is, however, whether the integrated data preparation scheme is of value in neural network data analysis given that data preparation is time-consuming. In the next section, we will discuss this issue from a general viewpoint.

4 Costs-Benefits Analysis of the Integrated Data Preparation Scheme

Data preparation requires extra time and effort, hence, the question of costs versus benefits arises (Problem III). Decision makers may not be willing to invest in data preparation if it has little impact on the final data analysis results of neural network models. This problem is analyzed from three aspects:

1. **Total time saving for neural network modeling.** Although additional time is needed to prepare data, the learning time of neural network data analysis may decrease sharply. As has been stated [71], every hour invested in data preparation may save many days in training a network. Therefore, our scheme will result in an overall time saving.
2. **Model complexity reduction for neural network modeling.** Usually, the model complexity of neural networks can be referred to as the number of parameters, namely, the number of weights and the number of biases. The complexity of neural network models can be mainly reduced to the number of adjustable parameters. Generally, the complexity of neural network models can be calculated as:

$$C(n) = n_h(n_i + 1) + n_o(n_h + 1), \tag{7}$$

where $C(n)$ is the model complexity, n_i is the number of input nodes, n_h is the number of hidden neurons, and n_o is the number of output nodes. From (7), we can see that the model complexity can be reduced by appropriate data preparation work, e.g., variable selection.

3. **Performance improvement for complex data analysis.** In practice, many applications [12], [13], [14], [15], [16], [17], [26], [27], [28], [29], [30], [31], also revealed that data preparation minimizes error. To explain this further, a bias-variance-noise decomposition achieved by extending the bias-variance decomposition originally proposed by [72] is used to analyze performance improvement.

Considering a classification or prediction problem, mean squared error is defined as a loss function of a neural network model. Assume that there is a true function, $y = f(x) + \varepsilon$, where ε is normally distributed with zero mean and standard deviation σ. Given a set of training sets $D : \{(x_i, y_i)\}$, we fit the unknown function $h(x) = w \cdot x + \varepsilon$ to the data by minimizing the squared error $\sum_i [y_i - h(x_i)]^2$. Given a new data point x^* with the observed value $y^* = f(x^*) + \varepsilon$, the expected error $E[(y^* - h(x^*))^2]$ can be decomposed into bias, variance, and noise below:

$$\begin{aligned}
E[h(x^*) - y^*)^2] &= E[(h(x^*))^2 - 2h(x^*)y^* + (y^*)^2] \\
&= E[h(x^*))^2] - 2E[h(x^*)]E(y^*)] + E(y^*)^2] \\
&\quad (\because E(Z - \bar{Z})^2 = E(Z^2) - \bar{Z}^2) \\
&= E[h(x^*) - \bar{h}(x^*))^2] + (\bar{h}(x^*))^2 \\
&\quad - 2h(x^*)f(x^*) + E(y^* - f(x^*))^2] + (f(x^*))^2 \\
&= E[h(x^*) - \bar{h}(x^*))^2] + E(y^* - f(x^*))^2] \\
&\quad + (f(x^*))^2 \\
&= Var(h(x^*)) + E(\varepsilon^2) + Bias^2(h(x^*)) \\
&= Var(h(x^*)) + \sigma^2 + Bias^2(h(x^*)).
\end{aligned} \tag{8}$$

As revealed in (8), we can improve the data analysis performance by three-fold data preparation work. First, noise reduction and filtering can alleviate the effects of noise because the noise does not always follow the normal distribution. Next, data division, data validation, and data regrouping can effectively eliminate the effects of bias. Finally, every data preprocessing technique can reduce the effects of variance. However, this is only a theoretical discussion. The impact of the data preparation on neural network data analysis will be verified in the following.

5 Empirical Study

In this section, we provide a typical example, "business financial risk classification," to show the application of the data preparation technique and to explore the impact of data preparation on neural network data analysis. In this study, a back propagation neural network (BPNN), a widely used network type, is selected as an agent to test the impact of data preparation.

5.1 Experimental Design—Basics of Data Preparation and Experiment Settings

This application is related to the problems of evaluating corporate financial risk. Generally, corporate financial risk is roughly divided into four types: security, light-warning, heavy-warning, and crisis. The objective of financial risk classification is to evaluate a certain corporate financial condition. The data to be analyzed are a large amount of financial data from financial statements. Through objective analysis, we can use financial ratios to realize financial risk evaluation. A large number of firms from the Shanghai and Shenzhen Stock Exchange in China from 1991 to 2003 were collected using open financial statements. From this large set, 100 firms meeting the criteria of 1) having been in business for more than 10 years and 2) having data available were selected. If the data collected in this phase are used in neural network learning and data analysis, we call the phase "***neural network data analysis model with simple data preparation***." In this phase, we have 11 variables (total assets, net income, gross profit, net worth, long-term debt, current liabilities, inventories, current assets, net fixed assets, quick assets, working capital) for analysis. Through many experiments, a BPNN with an 11-23-4 architecture is chosen to classify the corporate financial risk. Because there is no systematic and normalized data sets (i.e., database) for this specified classification task, almost all data are scattered. Furthermore, many financial statements collected are paper manuscripts. Therefore, these data are required to be integrated. As we collect 100 firms with 10-year data, such a simple data preparation is a time-consuming task.

At this stage in our proposed integrated data preparation scheme, the data collected are raw and cannot be used for direct learning. Therefore, data combination and integration are first used. Using the financial statements of the firms (i.e., balance sheets and income statements), 27 financial ratios (variables) were calculated: net income/gross profit, gross profit/total assets, net income/total assets, net income/net worth, net income/(long-term debt + current liabilities), inventories/total assets, inventories/current assets, current liabilities/(long-term debt + current liabilities), net fixed assets/total assets, current assets/current liabilities, quick assets/current liabilities, working capital/total assets, working capital/current assets, (long-term debt + current liabilities)/net worth, (long-term debt + current liabilities)/net fixed assets, net worth/(long-term debt +net worth), net income/working capital, current liabilities/inventories, current liabilities/net worth, net worth/net fixed assets, inventories/working capital, and (long-term debt + current liabilities)/working capital, net worth/total assets, current liabilities/total assets, quick assets/total assets, working capital/net worth, current assets/total assets. In this phase, these data can be modeled by neural networks. We call this "***neural network data analysis model with ordinary data preparation***." By trial and error, a BPNN with a 27-51-4 structure is used for classification.

In terms of our proposed data preparation scheme, some other data preparation tasks are required at this stage. The first is variable selection. Here, GA is used, resulting in the retention of 12 financial ratios from 27 available ratios. The 12 are: net income/gross profit, gross profit/total assets, net income/total assets, net income/net worth, current assets/current liabilities, quick assets/current liabilities, (long-term debt + current liabilities)/total assets, net worth/(net worth + long-term debt), net worth/net fixed assets, inventories/working capital, current liabilities/total assets, and working capital/net worth. Similarly, with regard to missing data and the nonavailability of sales volumes, corresponding processing techniques are used to eliminate the effects of data anomalies. In addition, all samples are divided into three parts: training sets (60 firms) for learning, validation sets (25 firms) for reducing the fitting problem of network learning, and testing sets (15 firms) for testing the generalization of the network. According to the feedback of the neural network, we can adjust data division so as to make learning effective. Here, we use the term "***neural network data analysis model with integrated data preparation scheme***." In this phase, we use a BPNN with a 12-25-4 architecture to classify the corporate financial risk based upon the results of many experiments.

Specifically, the experimental platform is windows and the BPNN model is based on the *Matlab neural network toolbox*. Accordingly, a learning rate of 0.50, a momentum rate of 0.15, and random initial weights are chosen. The maximum number of learning epochs (cycles) is set at 10,000. A learning epoch means that the network goes

TABLE 1
Comparison of Preparation Time and Learning Time

Time comparison	Preparation time	Learning time	Total time
(1) Simple data preparation	1.16 hours	13.55 hours	14.71 hours
(2) Ordinary data preparation	2.21 hours	21.18 hours	23.39 hours
(3) Integrated data preparation	3.57 hours	1.67 hours	5.24 hours
(4) Improvement I: (2)-(1)	1.05 hours	7.63 hours	8.68 hours
(5) Improvement II: (3)-(1)	2.41 hours	-11.88 hours	-9.47 hours
(6) Improvement III: (3)-(2)	1.36 hours	-19.51 hours	-18.15 hours

TABLE 2
Comparison of Model Complexity

Complexity comparison	Complexity (In terms of Equation (7))
(1) Simple data preparation	372
(2) Ordinary data preparation	1636
(3) Integrated data preparation	429
(4) Improvement I: (2)-(1)	1264
(5) Improvement II: (3)-(1)	57
(6) Improvement III: (3)-(2)	-1207

through all the years of training data once. An activation function of the logistic function for the hidden layer and a linear function for the output layer are selected. The stop rule is that MSE is less than 0.0002. In addition, the output (1, 0, 0, 0), (0, 1, 0, 0), (0, 0, 1, 0), and (0, 0, 0, 1) represent the four financial conditions, i.e., security, light warning, heavy warning, and crisis, as earlier mentioned.

5.2 Experimental Results—The Impact of Data Preparation on Neural Data Analysis

In this section, we focus on analyzing the impact of data preparation on neural network data analysis from three perspectives: total time saving, model complexity reduction, and performance improvement. Based on the previous descriptions, neural network data analysis can be performed with three different data preparations. A comprehensive analysis is undertaken to evaluate the effects of data preparation on neural network data analysis. First, we compare data preparation time and network learning time with different degrees of data preparation (see Table 1). It is worth noting that the calculation of data preparation time starts data requirement analysis and ends data readjustment.

From Table 1, we can see 1) that, although integrated data preparation time takes a little longer than simple and ordinary data preparation (cost invested), the overall learning time with our integrated data preparation scheme is much shorter (benefit obtained). 2) Usually, prepared data can speed up network learning time, such as improvements II and III. If data are processed incompletely, the network learning time may be longer due to increased complexity of the network, as improvement I indicates. 3) In general, total time saved is about 10 hours for this classification task. We can save more time in extremely complex data analysis, such as complex satellite image data. This implies that the integrated data preparation scheme has a significant impact on neural network learning time.

Subsequently, we compare the model complexity among three different data preparation schemes, as shown in Table 2.

From Table 2, we can find that the integrated data preparation scheme can effectively reduce model complexity from improvement II, relative to the ordinary data preparation. The main reason leading to this situation is that ordinary data preparation cannot process data completely and resulting in the increase of model complexity.

Table 3 shows performance improvement via classification accuracy, with some meaningful results:

1. The classification accuracy of a neural network model with an integrated data preparation scheme is much greater than those of neural network models with simple and ordinary data preparation, in terms of training set, validation set, and testing set. The main reason is because redundant information is reduced or eliminated under the proposed scheme.
2. The classification accuracy of the testing set is not as high as in the training and validation sets, as is shown in lines 1-3 in Table 2. A possible reason is that classifying the unknown mode is difficult because of the uncertainty of future events.
3. Unlike in Table 1, a neural network with ordinary data preparation can obtain better classification results than a neural network with simple data

TABLE 3
Comparison of Classification Accuracy

Accuracy comparison	Training set (%)	Validation set (%)	Testing set (%)
(1) Simple data preparation	83.33	56.00	40.00
(2) Ordinary data preparation	91.67	84.00	73.33
(3) Integrated data preparation	96.67	96.00	86.67
(4) Improvement I: (2)-(1)	8.34	28.00	33.33
(5) Improvement II: (3)-(1)	13.34	40.00	46.67
(6) Improvement III: (3)-(2)	5.00	12.00	13.34

preparation, although the former increases learning time.

4. As expected, the performance improvement with the integrated data preparation scheme is great relative to simple data preparation.

For example, the improvement of the testing set is over 40 percent (86.67 percent - 40.00 percent = 46.67 percent). That is, although it takes additional effort to process the data, the benefit obtained is very large. This implies that additional efforts to perform data preparation in practical data analysis applications are worthwhile.

To summarize, the benefits of data preparation are threefold: 1) decreased running time of neural networks modeling, 2) reduced the model complexity of neural network modeling, and 3) improved performance in neural network data analysis. These findings imply that data preparation has a significant effect on neural network data analysis. In addition, empirical results demonstrated the effectiveness and efficacy of the proposed scheme. Therefore, the proposed data preparation scheme is worth being generalized.

6 Conclusions and Future Directions

In this study, a comprehensive data preparation scheme with three-phase data processing is proposed to obtain better performance for specific neural network data analysis task. The novel integrated data preparation scheme proposed in this paper will enhance the neural network learning process and reduce the complexity of neural network models and will be of immense help for complex data analysis. Through empirical investigations, several important conclusions were obtained: 1) The integrated data preparation scheme can significantly speed up data analysis, reduce model complexity, and improve the performance of data analysis tasks. 2) The scheme is necessary and beneficial in data preparation for neural network data analysis. As the proposed integrated data preparation scheme has been proven to be very effective in the performance improvement of neural network data analysis, this leads to the final conclusion. 3) The proposed integrated data preparation scheme can be used as a promising solution to improve the performance of neural network data analysis, which is worth being generalized in the future.

To summarize, the main contributions of this study are the following four. One contribution is to propose an integrated data preparation scheme with three-phase data processing for neural network data analysis. Although some related studies about data preparation are presented in the literature, a systematic study of data preparation has not been formulated so far. Another contribution is to present an overview of data preparation. In neural network data analysis, we have discussed a number of data preparation techniques in three phases and, accordingly, some practical solutions to some dilemmas are provided. Although many techniques are shown in the previous literature, such as the work of Fayyad et al. [73], but there are some distinct differences between our work and the previous work. First of all, the previous work, such as [21], [73], only focused on the data preprocessing, while our work is broader and mainly focuses on all data processing including data preanalysis and data postanalysis in addition to data preprocessing. Second, their work does not present a systematic study for data preparation. Their works about data preparation are scattered and nonsystematic. Comparatively speaking, our proposed integrated scheme presents a more comprehensive study for data preparation. The third contribution of our integrated data preparation scheme is to present a comprehensive survey about neural network data preparation, which is different from others. In addition, some new data preparation techniques, such as GA for variable selection, are suggested for neural network data preparation. The final contribution is to provide a full cost-benefit analysis framework for integrated data preparation scheme. These contributions fill up the gap in previous studies. However, there are still some important issues to be considered for future research on the data preparation for complex data analysis:

1. The scope of this study is limited to neural network data analysis. Future research should extend more data analysis models such as data mining and knowledge discovery models.
2. The study only presents some limited data preparation techniques in the integrated data preparation scheme. More new data preparation techniques in all steps of the integrated data preparation scheme are worth exploring.

3. To perform meaningful data preparation, either the domain expert should be a member of the data analysis team or the domain should be extensively studied before the data are preprocessed. The involvement of the domain expert would lead to useful feedback for verifying and validating the use of particular data preparation techniques. Thus, the integration of expert opinions into a neural network data preparation framework is an important issue.
4. A module for analyzing the effects of data preparation should be added to neural network software packages so that users working in other domains can more easily understand the impact of data preparation techniques on their work.

ACKNOWLEDGMENTS

The authors would like to thank the guest editors and three anonymous reviewers for their valuable comments and suggestions. Their comments helped to improve the quality of the paper immensely. This work is partially supported by NSFC, CAS, SRG of City University of Hong Kong.

REFERENCES

[1] X. Hu, "DB-H Reduction: A Data Preprocessing Algorithm for Data Mining Applications," *Applied Math. Letters,* vol. 16, pp. 889-895, 2003.

[2] K.U. Sattler and E. Schallehn, "A Data Preparation Framework Based on a Multidatabase Language," *Proc. Int'l Symp. Database Eng. & Applications,* pp. 219-228, 2001.

[3] M. Lou, "Preprocessing Data for Neural Networks," *Technical Analysis of Stocks & Commodities Magazine,* Oct. 1993.

[4] D. Pyle, *Data Preparation for Data Mining.* Morgan Kaufmann, 1999.

[5] M.W. Gardner and S.R. Dorling, "Artificial Neural Networks (the Multilayer Perceptron)—A Review of Applications in the Atmospheric Sciences," *Atmospheric Environment,* vol. 32, pp. 2627-2636, 1998.

[6] M.Y. Rafiq, G. Bugmann, and D.J. Easterbrook, "Neural Network Design for Engineering Applications," *Computers & Structures,* vol. 79, pp. 1541-1552, 2001.

[7] K.A. Krycha and U. Wagner, "Applications of Artificial Neural Networks in Management Science: A Survey," *J. Retailing and Consumer Services,* vol. 6, pp. 185-203, 1999.

[8] K.J. Hunt, D. Sbarbaro, R. Bikowski, and P.J. Gawthrop, "Neural Networks for Control Systems—A Survey," *Automatica,* vol. 28, pp. 1083-1112, 1992.

[9] D.E. Rumelhart, "The Basic Ideas in Neural Networks," *Comm. ACM,* vol. 37, pp. 87-92, 1994.

[10] K.S. Narendra and K. Parthasarathy, "Identification and Control of Dynamic Systems Using Neural Networks," *IEEE Trans. Neural Networks,* vol. 1, pp. 4-27, 1990.

[11] M.R. Azimi-Sadjadi and S.A. Stricker, "Detection and Classification of Buried Dielectric Anomalies Using Neural Networks—Further Results," *IEEE Trans. Instrumentations and Measurement,* vol. 43, pp. 34-39, 1994.

[12] A. Beltratli, S. Margarita, and P. Terna, *Neural Networks for Fconomic and Financial Modeling.* London: Int'l Thomson Publishing Inc., 1996.

[13] Y. Senol and M.P. Gouch, "The Application of Transputers to a Sounding Rocket Instrumentation: On-Board Autocorrelators with Neural Network Data Analysis," *Parallel Computing and Transputer Applications,* pp. 798-806, 1992.

[14] E.J. Gately, *Neural Networks for Financial Forecasting.* New York: John Wiley & Sons, Inc., 1996.

[15] A.N. Refenes, Y. Abu-Mostafa, J. Moody, and A. Weigend, *Neural Networks in Financial Engineering.* World Scientific Publishing Company, 1996.

[16] K.A. Smith and J.N. D. Gupta, *Neural Networks in Business: Techniques and Applications.* Hershey, Pa.: Idea Group Publishing, 2002.

[17] G.P. Zhang, *Neural Networks in Business Forecasting.* IRM Press, 2004.

[18] B.D. Klein and D.F. Rossin, "Data Quality in Neural Network Models: Effect of Error Rate and Magnitude of Error on Predictive Accuracy," *OMEGA, The Int'l J. Management Science,* vol. 27, pp. 569-582, 1999.

[19] T.C. Redman, *Data Quality: Management and Technology.* New York: Bantam Books, 1992.

[20] T.C. Redman, *Data Quality for the Information Age.* Norwood, Mass.: Artech House, Inc., 1996.

[21] S. Zhang, C. Zhang, and Q Yang, "Data Preparation for Data Mining," *Applied Artificial Intelligence,* vol. 17, pp. 375-381, 2003.

[22] A. Famili, W. Shen, R. Weber, and E. Simoudis, "Data Preprocessing and Intelligent Data Analysis," *Intelligent Data Analysis,* vol. 1, pp. 3-23, 1997.

[23] R. Stein, "Selecting Data for Neural Networks," *AI Expert,* vol. 8, no. 2, pp. 42-47, 1993.

[24] R. Stein, "Preprocessing Data for Neural Networks," *AI Expert,* vol. 8, no. 3, pp. 32-37, 1993.

[25] A.D. McAulay and J. Li, "Wavelet Data Compression for Neural Network Preprocessing," *Signal Processing, Sensor Fusion, and Target Recognition SPIE,* vol. 1699, pp. 356-365, 1992.

[26] V. Nedeljkovic and M. Milosavljevic, "On the Influence of the Training Set Data Preprocessing on Neural Networks Training," *Proc. 11th IAPR Int'l Conf. Pattern Recognition*, pp. 1041-1045, 1992.

[27] J. Sjoberg, "Regularization as a Substitute for Preprocessing of Data in Neural Network Training," *Artificial Intelligence in Real-Time Control,* pp. 31-35, 1992.

[28] O.E. De Noord, "The Influence of Data Preprocessing on the Robustness and Parsimony of Multivariate Calibration Models," *Chemometrics and Intelligent Laboratory Systems,* vol. 23, pp. 65-70, 1994.

[29] J. DeWitt, "Adaptive Filtering Network for Associative Memory Data Preprocessing," *Proc. World Congress on Neural Networks,* vol. IV, pp. 34-38, 1994.

[30] D. Joo, D. Choi, and H. Park, "The Effects of Data Preprocessing in the Determination of Coagulant Dosing Rate," *Water Research,* vol. 34, pp. 3295-3302, 2000.

[31] H.H. Nguyen and C.W. Chan, "A Comparison of Data Preprocessing Strategies for Neural Network Modeling of Oil Production Prediction," *Proc. Third IEEE Int'l Conf. Cognitive Informatics,* 2004.

[32] J. Pickett, *The American Heritage Dictionary*, 4th ed. Boston: Houghton Mifflin, 2000.

[33] P. Ingwersen, *Information Retrieval Interaction.* London: Taylor Graham, 1992.

[34] U.Y. Nahm, "Text Mining with Information Extraction: Mining Prediction Rules from Unstructured Text," PhD thesis, 2001.

[35] F. Lemke and J.A. Muller, "Self-Organizing Data Mining," *Systems Analysis Modelling Simulation,* vol. 43, pp. 231-240, 2003.

[36] E. Tuv and G. Runger, "Preprocessing of High-Dimensional Categorical Predictors in Classification Setting," *Applied Artificial Intelligence,* vol. 17, pp. 419-429, 2003.

[37] C.W.J Granger, "Investigating Causal Relations by Econometric Models and Cross-Spectral Methods," *Econometrica,* vol. 37, pp. 424-438, 1969.

[38] K.I. Diamantaras and S.Y. Kung, *Principal Component Neural Networks: Theory and Applications.* John Wiley and Sons, Inc., 1996.

[39] D.W. Ashley and A. Allegrucci, "A Spreadsheet Method for Interactive Stepwise Multiple Regression," *Proceedings,* pp. 594-596, Western Decision Sciences Inst., 1999.

[40] X. Yan, C. Zhang, and S. Zhang, "Toward Databases Mining: Preprocessing Collected Data," *Applied Artificial Intelligence,* vol. 17, pp. 545-561, 2003.

[41] S. Chaudhuri and U. Dayal, "A Overview of Data Warehousing and OLAP Technology," *SIGMOD Record,* vol. 26, pp. 65-74, 1997.

[42] S. Abiteboul, S. Cluet, T. Milo, P. Mogilevsky, J. Simeon, and S. Zohar, "Tools for Translation and Integration," *IEEE Data Eng. Bull.,* vol. 22, pp. 3-8, 1999.

[43] A. Baumgarten, "Probabilistic Solution to the Selection and Fusion Problem in Distributed Information Retrieval," *Proc. SIGIR'99,* pp. 246-253, 1999.

[44] Y. Li, C. Zhang, and S. Zhang, "Cooperative Strategy for Web Data Mining and Cleaning," *Applied Artificial Intelligence,* vol. 17, pp. 443-460, 2003.

[45] J.H. Holland, "Genetic Algorithms," *Scientific Am.,* vol. 267, pp. 66-72, 1992.

[46] D.E. Goldberg, *Genetic Algorithm in Search, Optimization, and Machine Learning.* Reading, Mass.: Addison-Wesley, 1989.

[47] A.M. Kupinski and M.L. Giger, "Feature Selection with Limited Datasets," *Medical Physics,* vol. 26, pp. 2176-2182, 1999.

[48] Mani Bloedorn and E. Bloedorn, "Multidocument Summarization by Graph Search and Matching," *Proc. 15th Nat'l Conf. Artificial Intelligence,* pp. 622-628, 1997.

[49] M. Saravanan, P.C. Reghu Raj, and S. Raman, "Summarization and Categorization of Text Data in High-Level Data Cleaning for Information Retrieval," *Applied Artificial Intelligence,* vol. 17, pp. 461-474, 2003.

[50] W.A. Shewhart, *Economic Control of Quality of Manufactured Product.* New York: D. Van Nostrand, 1931.

[51] D.A. Dickey and W.A. Fuller, "Distribution of the Estimators for Autoregressive Time Series with a Unit Root," *J. Am. Statistical Assoc.,* vol. 74, pp. 427-431, 1979.

[52] J. Wang, C. Zhang, X. Wu, H. Qi, and J. Wang, "SVM-OD: A New SVM Algorithm for Outlier Detection," *Proc. ICDM'03 Workshop Foundations and New Directions of Data Mining,* pp. 203-209, 2003.

[53] J. Han and Y. Fu, "Dynamic Generation and Refinement of Concept Hierarchies for Knowledge Discovery in Database," *Proc. AAAI '94 Workshop Knowledge Discovery in Database,* pp. 157-168, 1994.

[54] U. Fayyad and K. Irani, "Multiinterval Discretization of Continuous-Valued Attributes for Classification Learning," *Proc. 13th Int'l Joint Conf. Artificial Intelligence,* pp. 1022-1027, 1993.

[55] A. Srinivasan, S. Muggleton, and M. Bain, "Distinguishing Exceptions from Noise in Nonmonotonic Learning," *Proc. Second Int'l Workshop Inductive Logic Programming,* 1992.

[56] G.H. John, "Robust Decision Trees: Removing Outliers from Data," *Proc. First Int'l Conf. Knowledge Discovery and Data Mining,* pp. 174-179, 1995.

[57] D. Gamberger, N. Lavrac, and S. Dzeroski, "Noise Detection and Elimination in Data Preprocessing: Experiments in Medical Domains," *Applied Artificial Intelligence,* vol. 14, pp. 205-223, 2000.

[58] G.E. Batista and M.C. Monard, "Experimental Comparison of K-Nearest Neighbor and Mean or Mode Imputation Methods with the Internal Strategies Used by C4.5 and CN2 to Treat Missing Data," Technical Report 186, ICMC USP, 2003.

[59] G.E. Batista and M.C. Monard, "An Analysis of Four Missing Data Treatment Methods for Supervised Learning," *Applied Artificial Intelligence,* vol. 17, pp. 519-533, 2003.

[60] R.J. Little and P.M. Murphy, *Statistical Analysis with Missing Data.* New York: John Wiley and Sons, 1987.

[61] A. Ragel and B. Cremilleux, "Treatment of Missing Values for Association Rules," *Proc. Second Pacific-Asia Conf. Knowledge Discovery and Data Mining,* pp. 258-270, 1998.

[62] R.C.T. Lee, J.R. Slagle, and C.T. Mong, "Application of Clustering to Estimate Missing Data and Improve Data Integrity," *Proc. Int'l Conf. Software Eng.,* pp. 539-544, 1976.

[63] S.M. Tseng, K.H. Wang, and C.I. Lee, "A Preprocessing Method to Deal with Missing Values by Integrating Clustering and Regression Techniques," *Applied Artificial Intelligence,* vol. 17, pp. 535-544, 2003.

[64] A.S. Weigend and N.A. Gershenfeld, *Time Series Prediction: Forecasting the Future and Understanding the Past.* Addison-Wesley, 1994.

[65] F.M. Tseng, H.C. Yu, and G.H. Tzeng, "Combining Neural Network Model with Seasonal Time Series ARIMA Model," *Technological, Forecasting, and Social Change,* vol. 69, pp. 71-87, 2002.

[66] J. Moody, "Economic Forecasting: Challenges and Neural Network Solution," *Proc. Int'l Symp. Artificial Neural Networks,* 1995.

[67] J.T. Yao and C.L. Tan, "A Case Study on Using Neural Networks to Perform Technical Forecasting of Forex," *Neurocomputing,* vol. 34, pp. 79-98, 2000.

[68] K. Hornik, M. Stinchcombe, and H. White, "Multilayer Feedforward Networks Are Universal Approximators," *Neural Networks,* vol. 2, no. 5, pp. 359-366, 1989.

[69] A. Esposito, M. Marinaro, D. Oricchio, and S. Scarpetta, "Approximation of Continuous and Discontinuous Mappings by a Growing Neural RBF Based Algorithm," *Neural Networks,* vol. 13, pp. 651-665, 2000.

[70] H. Martens and T. Naes, *Multivariate Calibration.* New York: John Wiley & Sons Inc., 1989.

[71] R. Rojas, *Neural Networks: A Systematic Introduction.* Berlin: Springer-Verlag, 1996.

[72] S. Geman, E. Bienenstock, and R. Doursat, "Neural Networks and the Bias/Variance Dilemma," *Neural Computation,* vol. 4, pp. 1-58, 1992.

[73] U.M. Fayyad, G. Piatetsky-Shapiro, P. Smyth, and R. Uthurusamy, *Advances in Knowledge Discovery and Data Mining.* Menlo Park, Calif.: AAAI Press, 1996.

Lean Yu received the PhD degree in management sciences and engineering from the Institute of Systems Science, Academy of Mathematics and Systems Sciences, Chinese Academy of Sciences. He is currently a research fellow in the Department of Management Sciences at the City University of Hong Kong. His research interests include artificial neural networks, computer simulation, decision support systems, and financial forecasting.

Shouyang Wang received the PhD degree in operations research from the Institute of Systems Science, Chinese Academy of Sciences (CAS), Beijing, in 1986. He is currently a Bairen Distinguished Professor of Management Science in the Academy of Mathematics and Systems Sciences at CAS and a Lotus Chair Professor at Hunan University, Changsha. He is the editor-in-chief or a coeditor of 12 journals. He has published 18 books and more than 120 journal papers. His current research interests include financial engineering, e-auctions, and decision support systems.

K.K. Lai received the PhD degree from Michigan State University. He is the Chair Professor of management science at City University of Hong Kong and he is also the associate dean of the Faculty of Business. Currently, he is also acting as the dean of the College of Business Administration at Hunan University, China. Prior to his current post, he was a senior operational research analyst at Cathay Pacific Airways and the area manager on marketing information systems at Union Carbide Eastern. Professor Lai's main research interests include logistics and operations management, computer simulation, AI, and business decision modeling.