IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received April 20, 2020, accepted May 11, 2020, date of publication May 25, 2020, date of current version June 16, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.2997255

Multi-Modal Stacked Denoising Autoencoder for Handling Missing Data in Healthcare Big Data

JOO-CHANG KIM[1] AND KYUNGYONG CHUNG[2]
[1]Department of Computer Science, Kyonggi University, Suwon-si 16227, South Korea
[2]Division of Computer Science and Engineering, Kyonggi University, Suwon-si 16227, South Korea

Corresponding author: Kyungyong Chung (dragonhci@gmail.com)

This work is supported by the Korea Agency for Infrastructure Technology Advancement(KAIA) grant funded by the Ministry of Land, Infrastructure and Transport (Grant 20CTAP-C157011-01).

ABSTRACT Supply and demand increase in response to healthcare trends. Moreover, personal health records (PHRs) are being managed by individuals. Such records are collected using different avenues and vary considerably in terms of their type and scope depending on the particular circumstances. As a result, some data may be missing, which has a negative effect on the data analysis, and such data should, therefore, be replaced with appropriate values. In this study, a method for estimating missing data using a multi-modal autoencoder applied to the field of healthcare big data is proposed. The proposed method uses a stacked denoising autoencoder to estimate the missing data that occur during the data collection and processing stages. Autoencoders are neural networks that output value of x^ similar to an input value of x. In the present study, data from the Korean National Health Nutrition Examination Survey (KNHNES), conducted by the Korea Centers for Disease Control and Prevention (KCDC), are used. As representative healthcare data from South Korea, they contain a large number of parameters identical to those used in the PHRs. Based on this, models can be generated to estimate missing data occurring in PHRs. Furthermore, PHRs involve a multi-modality that allows the data to be collected from multiple sources for a single object. Therefore, the stacked denoising autoencoder applied is configured under a multi-modal setting. Through pre-processing, a set of data without missing value in KNHNES is designed. In the data set based learning, a label is set as original data, and an autoencoder input is set as noised input that additionally has as many random zero numbers as noise factor. In this way, the autoencoder learns in the way of making the zero-based noise value similar to the original label value. When the amount of missing data in a dataset reaches approximately 25%, the accuracy of the proposed method using a multi-modal stacked denoising autoencoder is 0.9217, which is higher than that achieved by other ordinary methods. For a single-modal denoising autoencoder, the accuracy is 0.932, with a slight difference of approximately 0.01, which falls within the allowable limits in data analysis. In terms of computational performance, a single-modal autoencoder has 10,384 parameters, which is 5,594 more than those used in a multi-modal stacked autoencoder. These parameters affect the speed of the model. Both models exhibit a significant difference in the number of parameters but demonstrate a relatively small difference in accuracy, suggesting that the proposed multi-modal stacked denoising autoencoder is advantageous over a single-modal model when used on a personal device. Moreover, a multi-modal model can save additional time when processing large amounts of data in locations such as hospitals and institutions.

INDEX TERMS Autoencoder, data pre-processing, data estimation, data imputation, health big data, multi-modal, missing data, machine learning.

I. INTRODUCTION

Healthcare big data involve complex relationships among the different parameters and are adaptable to changes in the surroundings. As a result, soft computing technologies that make predictions and deductions regarding the parameters or other particular circumstances have been highlighted. Soft computing is a technique designed to handle imprecise and uncertain data in which mathematical modeling is difficult or impossible to apply. Many real-world problems cannot be

The associate editor coordinating the review of this manuscript and approving it for publication was Shuihua Wang.

clearly defined, and soft computing is used to computerize such ill-defined problems [1]. For example, the technique has been applied to find optimal answers to fuzzy propositions in the real world, such as ''big'', ''small'', ''cold'', ''hot'', ''light'', and ''heavy'', by converting them into a representation that can be understood by a computer. Moreover, soft computing is a machine learning technique designed to explore models with highest goodness-of-fit by repeating the encoding and evaluation for a given problem.

With advances in soft computing, health platforms integrating different sectors such as society, science, and industry are currently under development. These platforms utilize a variety of data, including electronic medical records (EMRs), personal health records (PHRs), and lifelogs [2], [3]. Furthermore, the computerization of existing accumulated medical records available in physical form allows for vast amounts of data to be continually collected. Existing medical records are difficult to integrate because they differ from the current system depending on the particular circumstances or institutions [4]. Handwritten diagnostic documents are collected in various forms depending on the format and are difficult to computerize. Despite the continued advancements in optical character recognition (OCR) technology, handwritten documents are often misunderstood or unrecognizable [5]. Meanwhile, the recognition rate of the cursive script has increased, reaching nearly 100%, with the development of both machine and deep learning. Nevertheless, research projects for utilizing recognized scripts have been undertaken in various fields, such as natural language processing, table processing, and language extension. As such, healthcare based on a soft computing approach fully utilizes all types of collectible data [1]–[5].

With the spread of smartphones and the compactness of personal health devices, new parameters are being generated, and the range of health data continues to expand. This represents a multi-modality that utilizes different data ranging from personal data, such as a patient's lifestyle, family history, or a pre-existing condition, to other areas, including the weather, GPS, and distance traveled [6]. A multi-modality is an environment in which different types of sensors or data sources are collected and utilized for a single object [7].

In a healthcare platform, users, patients, and regions act as an object. The multi-modal data collected through a multi-modal approach are used to make a prediction, deduction, and classification of the health conditions of the subject, thereby supporting the decision-making process. Furthermore, the provisioning of flexible and continuous services when considering the situation of the user is one of the main functions of a healthcare platform [8]. This demands various computing technologies such as data mining, context awareness, artificial intelligence, recommendation systems, and cognitive science. Creating a prediction, deduction, and classification model based on computing technologies requires vast amounts of data. Consequently, the multi-modal data collected from different objects need to be integrated. During the data integration process, the range of data collection varies depending on the particular circumstances, such as interest in an object, the device status, and the surrounding environment [9].

Missing data occur based on the specific circumstances of an object. Moreover, data duplication or omission may also occur during integration. To solve this problem, soft computing provides optimal estimates of the missing data [2]. Missing data affect the data analysis or learning, and a model generated from imperfect data tends to be less accurate during actual use. Duplicated or missing data can be estimated using values such as the mean, median, and mode, or using methods such as regression, neural network, singular value decomposition (SVD), or K-nearest neighbor (K-NN) [10]. Although an estimation using the mean value, median, and mode is simple to achieve, it is less accurate, which makes its application less viable. In addition, an estimation using a regression, SVD, or K-NN may achieve relatively high accuracy, but it requires user intervention, and extensive preprocessing is needed for algorithmic applications. Moreover, estimation using a neural network allows a model to learn the features from the data on its own, minimizing the need for user intervention. Therefore, in this study, a technique for estimating missing data, specifically the missing data in PHRs, using a multi-modal stacked denoising autoencoder in the area of healthcare big data is proposed.

The remainder of this paper is organized as follows. Chapter 2 discusses recent trends, including healthcare big data and the handling of missing data in machine learning. Chapter 3 describes the handling of missing data using a multi-modal stacked denoising autoencoder in the area of healthcare big data. Chapter 4 provides an evaluation of the performance of the proposed approach, and Chapter 5 provides some concluding remarks.

II. RELATED STUDIES

A. HEALTHCARE BIG DATA

Healthcare big data refers to any data related to human health. Because advancements in information and communication technology have facilitated data collection in the field of healthcare, healthcare data are currently being collected by individuals, government agencies, and hospitals [11]. Healthcare data can be classified into several categories, including personal genetic information, PHRs, and EMRs, depending on the target of the data collection. PHRs, EMRs, and lifelogs share common parameters, such as personal information and health screening items, and the composition of such parameters varies depending on the user. Parameters come in different forms, ranging from data obtained from surveys, such as the age, height, weight, and/or pre-existing condition of the individual, to contextual data, such as the weather, environment, and natural disasters [12]. As data utilization becomes more diverse, trends in the healthcare industry are shifting from treatment-oriented to prevention-oriented healthcare. This, in particular, has motivated the emergence of precision medicine focusing on individual patients.

Personal genetic information is unique information inherited from one's parents, and the human genome contains sequences of approximately 3 billion base pairs. As a key element of precision medicine tailored to provide optimal and patient-centered healthcare services, such information is playing a leading role in increasing the efficiency of treatment while reducing costs [13]. With recent developments in genetic engineering, increasing numbers of private companies are currently offering genetic testing services, consequently reducing costs. This also allows individuals to directly request for genetic testing, if desired, without having to visit a hospital. Similar to precision medicine, personal genetic information is used in human-centered healthcare services.

PHRs consist of data collected through sensors, smartphones, and personal healthcare and wearable devices, as well as data containing any medical practices recorded by an individual. PHRs include data collected, viewed, and managed by the subject of the health data collection. The composition of a PHR depends on the personal interests or devices of the individual, and many studies dealing with the integration and utilization of PHRs are currently being conducted [14]. PHRs are mainly used for daily health management, such as exercise, sleep, and weight management. For users requiring a follow-up, such as those with diabetes, PHRs can be used in conjunction with a personal health device for blood glucose monitoring. EMRs refers to any automated medical information in a hospital, such as diagnostic results, prescriptions, surgical records, and inpatient admission records. Computerization has enabled vast amounts of data to be processed, and EMRs incorporated into IT now serve as a basis for precision medicine [15]. For promoting better health, national healthcare information is collected in a database by the state (acting as the main agent). In South Korea, KNHNES [16] data are collected annually under the supervision of the KCDC. Such data contain information on the health and nutritional conditions to determine and evaluate the target indicators for the promotion of better public health. Such an examination is conducted to select vulnerable health groups through a survey regarding their tobacco and alcohol use, nutrient intake, physical activity, chronic conditions, disease recognition rate, and treatment rate, and to investigate trends in public health policy.

Personal information is provided in an unidentified form for the protection of personal health information. KNHNES data, including primary and statistical data, are available on the KNHNES website. Surveys are conducted on samples taken from census data from the most recent period and can be classified into health, health examination, and nutrition surveys. Health surveys are subdivided into household, health interview, and health behavior surveys, and consist of parameters such as the number of household members, household type, income level, education level, physical activity, tobacco and alcohol use, mental health, and safety awareness. An examination survey consists of the individual's weight, height, blood pressure, pulse rate, urine status, oral cavity condition, pulmonary function, and grip strength. A nutrition survey includes dietary lifestyle, dietary nutrition, supplements, and food knowledge.

As a continual area of focus, healthcare big data continue to evolve. By applying big data, image data such as those from an X-ray, CT, or MRI have recently been applied to machine learning. In particular, deep learning using a convolutional neural network (CNN) has achieved outstanding performance in the field of image recognition and classification, based upon which various healthcare models are being developed [17].

B. HANDLING MISSING DATA IN MACHINE LEARNING

Missing data are denoted using null, n/a, NaN, or blank spaces, indicating an empty space for non-zero data. This affects the machine learning outcomes to a significant extent, and if ignored frequently leads to errors in the training or analysis of the models [18]. Higher missing-data rates result in a poor-quality data analysis or modeling, which in turn causes significant losses to the industry. Therefore, proper processing of missing data is needed to conduct an accurate and meaningful data analysis or modeling. The simplest way to handle missing data is to eliminate all observations containing such data. An observation represents a sequence of transactions collected for an object. Most missing data occur in an irregular and different manner, and deleting an observation can cause loss of information [19]. Thus, it is necessary to estimate the most appropriate value by inferring any missing data. Missing data can be classified into three types: missing completely at random, missing at random, and not missing at random data. The methods for estimating missing data include applying the mean, mode, k-NN, support vector machine (SVM), or deep learning. Each method has its own strengths and weaknesses, and it is important to choose an appropriate estimation method based on the data characteristics [11], [20]. In addition to these universal methods, various techniques such as a randomized latent factor model [38], an effective scheme for QoS estimation [39], and an inherently nonnegative latent factor model [40] have been proposed depending on the field of application. Various types of data, such as human life, industry, and health data, are being collected, and the appropriate use and research into various techniques are required according to the data characteristics.

An estimation using the mean or median is a technique in which missing data are input through statistical calculations in a parameter column. Each column independently estimates missing data, and this type of estimation is available only for numerical data. Although an estimation method for calculating missing data estimates is simple and fast, the mean or mode does not consider the relationship between parameters and achieves a significantly lower accuracy for categorical data. Similarly, an estimation using the mode is also extremely simple to apply operationally. This type of estimation is primarily used for categorical data rather than numerical data. Similar to the mean or median, the model

does not consider the relationship between parameters. Furthermore, if the estimation involves using the most frequently observed values, bias may be introduced into the data. If bias is present in the data, there may be unintended outcomes of the data analysis or learning [19].

An estimation using k-NN [21] involves searching for k nearest neighbors to the observations in which missing data occur and imputing such data using a weighted mean of the neighbors. k-NN is an algorithm designed for a simple classification and uses the feature similarities to predict the values of the new data. The algorithm typically yields higher accuracy than the mean or mode. However, it requires many computations and only works when the entire training dataset is stored in memory. Furthermore, the appropriate value of k should be determined because the algorithm is sensitive to outliers, and the outcome will vary depending on k.

An estimation using an SVD [22] is a technique for predicting missing data using an output value x^ similar to input x by diagonalizing a matrix in linear algebra. This technique initializes a missing data value as 0 or as the mean of the column and iteratively applies an estimation through a linear combination of the k-most significant eigenvalue parameters until converging to the next threshold. An SVD is applicable to any m $\times$ n matrix. An orthogonal matrix is formed through an eigenvalue decomposition, and the created orthogonal matrix is used to generate the output value x^ similar to input x. The missing portion of x can be inferred from x^ for processing the missing data.

An estimation using deep learning [23] is a missing data prediction method that uses multiple weighted values generated through neural network learning. This method allows for a variety of representations based on neural network architectures. An autoencoder has a typical neural network architecture designed to estimate missing data. A stacked autoencoder is a learning method similar to a deep neural network (DNN) and is referred to as a deep network when the hidden layer comprises multiple layers. Autoencoders are neural networks that generate an output of value x^ similar to an input of value x and are mainly used for data compression and reducing the number of dimensions.

The encoder and decoder are symmetrically constructed: The network from the input layer up to the hidden layer in the middle is called the encoder, and the network from the output layer up to the hidden middle layer is called the decoder. The weights w and w^ of the symmetrical position is configured to be equal. If the hidden node is smaller than the input node in the neural network, input data compression and feature extraction can both be achieved. Autoencoders are generally used in pre-learning and to recover a source using the characteristics of the manifold and generative model learning.

In deep learning, the use of a generative adversarial network (GAN) is also an approach to replace missing values [45]. Using a GAN, we can construct a neural network that outputs x^ similar to the input x. Owing to its neural network structure, the use of a GAN has been attracting attention since it was first proposed. This approach produces a new output value for input x according to the learning direction. In a learned GAN, a generator and a discriminator compete with each other to learn. This produces virtual data with an output similar to the distribution of the actual data. A GAN is highly regarded owing to its outstanding performance in numerous domains and has demonstrated excellent performance in image reconstruction. In addition, a neural network can be applied to continuous data because the image is converted into a vector and then calculated.

However, a problem occurs in that learning becomes difficult, owing to the complicated structure. In addition, it is difficult to specify a specific time point for terminating the learning, resulting in a vanishing gradient from overfitting. Although a GAN is useful for generating new data, its learning requires much more training data than a normal neural network. Healthcare data require an output that is closest to the input. In addition, because the types of data that can be collected vary depending on the user's particular situation, generalization and scalability are required. The processing of healthcare data using a GAN faces a problem in that the data are difficult to apply to a real situation owing to high learning difficulty and low scalability. In addition, healthcare data contain many variables, and thus it is necessary to consider the relationship between them.

III. HANDLING OF MISSING DATA USING MULTI-MODAL STACKED DENOISING AUTOENCODER IN HEALTHCARE BIG DATA

KNHNES [16] data can be classified into health, health examination, and nutritional survey data. Health survey data consist of an individual's lifestyle, family history, and disease, and medical records. A health examination survey consists of the pulse rate, blood pressure, weight, height, and blood glucose level of the individual. A nutritional survey consists of an individual's meal frequency, meal size, water intake, and use of dietary supplements. KNHNES is not a typical PHR but has a high potential owing to its vast amount of data, which are also contained in the PHR. Approximately 600 parameters containing a number of health-related items are applied, most of which can be prepared and managed by individuals. Therefore, a model derived from KNHNES is highly applicable to a PHR. Moreover, KNHNES is a type of multi-modal data collected through a variety of modes. In fact, data from health surveys, health examinations, and dietary nutrition are collected through each mode according to the data collection path. During an integration process, missing data are introduced through diverse circumstances. If there are numerous transactions containing missing data, the outcomes of the data analysis will vary depending on the pre-processing techniques. This requires an appropriate processing technique that minimizes the effects of the missing data on the outcome of the data analysis. In this study, a method for estimating missing data using a multi-modal stacked denoising autoencoder in the field of healthcare big data is proposed. This technique

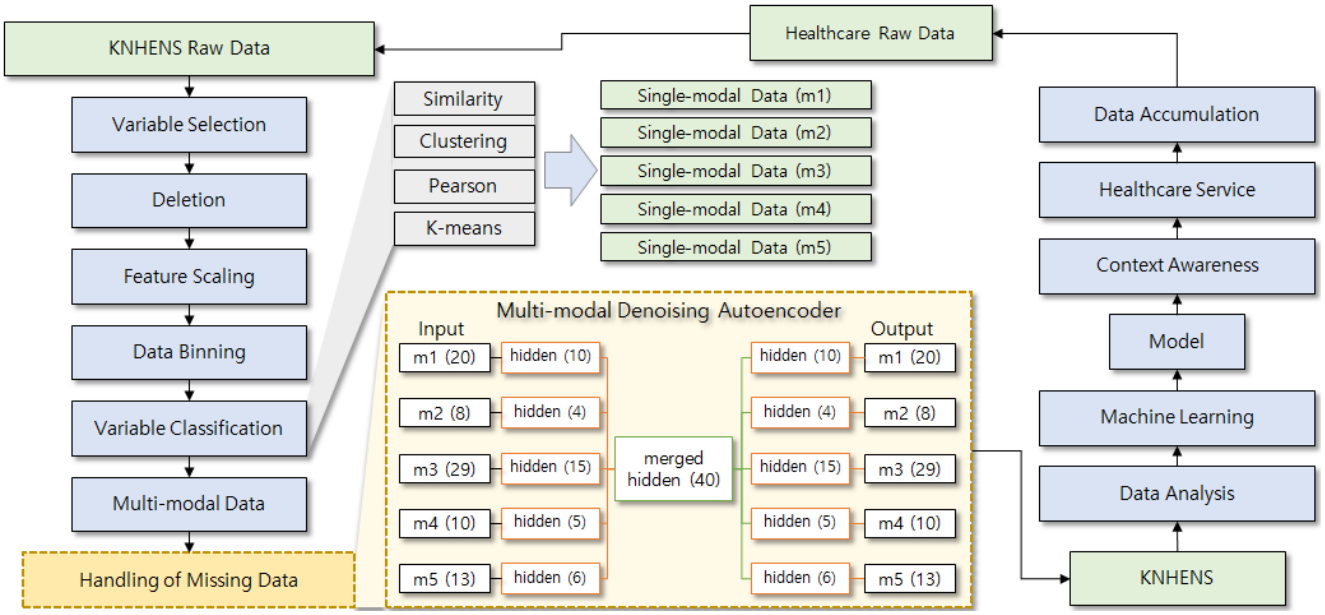

FIGURE 1. Configuration used for estimating missing data.

estimates missing data using a trained autoencoder for multi-modal data. KNHNES includes various variables, and various multi-modal configurations are possible according to the classification method used for the variables. By handling missing data, better results from a data analysis and machine learning can be expected. Fig. 1 shows the configuration used for estimating such missing data.

A. PRE-PROCESSING OF NATIONAL HEALTH AND NUTRITION SURVEY DATA

Among the primary data released by KNHNES [16] from 2013 to 2017, data from health, health examination, and nutritional surveys are used in the present study. In addition, only common parameters shared among the surveys are applied because the scope of each survey varies annually, and any parameters having 20% of data missing or greater in a column scan are excluded. Among the primary data, 198 parameters are selected, including pre-existing health conditions, physician diagnosis, tobacco use, alcohol use, stress, and depression. For the 189 parameters selected, 14,688 cases were applied for this study, excluding those cases with a missing data rate of 20% or higher determined through a column scan.

KNHNES primary data include "Not Applicable (NA)" (8, 88, and 888) and "No Answer" (9, 99, and 999) responses. Such responses are not considered to be missing data, although certain classes may affect the outcome of the data analysis. In particular, an NA response is included in numerous parameters, although the parameters containing a significant number of NA responses (3,000 or more) are excluded from all 14,688 cases. Accordingly, 80 parameters are selected and used as the experimental data. KNHNES contains data indicated by a 0 (class value), and proper processing is therefore required. For categorical parameters in the 14,668 test data, each class is assigned a value of +1 to change the values across all classes as follows: 0(no)→1(no), 1(yes)→ 2(yes). Moreover, No Answer (9, 99, and 999), NA, and NULL values are pre-processed into a value of 0 to indicate missing data. If continuous parameters have different scales, this may result in excessively large values or a convergence of the weights to 0 [18], [19]. A difference in scale between parameters may lead to unstable learning of the weights, thereby requiring a feature scaling [24].

Feature scaling is a technique used to normalize the ranges of the parameters to be scaled equally. This technique is employed to evaluate the influence of a particular parameter during data analysis, such as through a regression or clustering analysis, as well as in a neural network model. Therefore, it converts the parameters into the categorical type via data binning while applying feature scaling. Binning is a categorization technique for dividing continuous parameters into intervals (bins). KNHNES facilitating the application of data binning and feature scaling methods to categorize the parameters of different units, including the individual's age, height, and blood pressure. With a minimum parameter value of 0% and a maximum value of 100%, the parameters are categorized into nine intervals. Table 1 shows the binning and features scaling of continuous parameters from the KNHNES data. Because missing data are set to a value of 0 in the experimental data, the parameters are binned within the range of 1–9.

B. ESTIMATION OF MISSING VALUES USING STACKED DENOISING AUTOENCODER

The denoising autoencoder applied is a modification of the learning methods used in ordinary autoencoders [25]. This autoencoder adds random noise to noise-free input data and

TABLE 1. Binning and feature scaling of continuous variables.

Binning	0	1	2	3	4
Normalization	N/A	0–0.1	0.1–0.2	0.2–0.3	0.3–0.4
Binning	5	6	7	8	9
Normalization	0.4–0.5	0.6–0.7	0.7–0.8	0.8–0.9	0.9–1.0

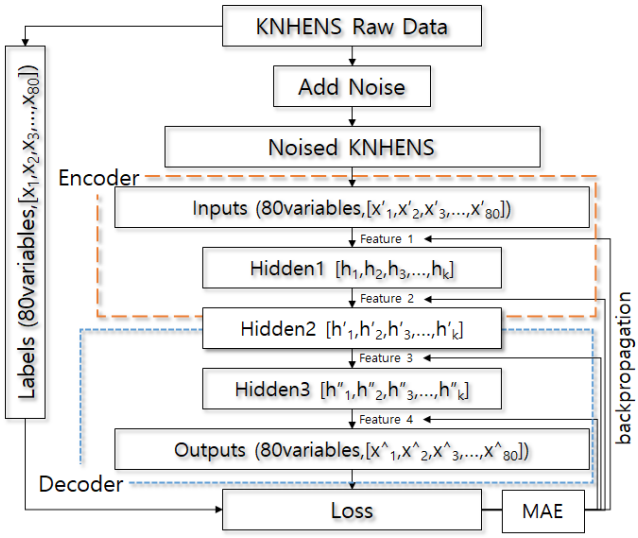

FIGURE 2. Estimation of missing values using stacked denoising autoencoder.

learns with the aim of restoring the original noise-free data. It repeats the process of randomly adding noise to the input value and then restoring it to the original data. The stacked denoising autoencoder randomly selects a value from the original data before data entry and converts it into a 0. In neural network learning, missing data are normally estimated as 0. Similarly, when applying denoising autoencoder learning, the noise is estimated as 0 and restored to the original data when missing data occur. Accordingly, a value of 0 is entered when missing data occur, which in turn will be input using a non-zero predicted value through a trained neural network. When expressing the modality according to the data characteristics as a neural network, several hidden layers are required. In addition, when stacking hidden layers into multiple layers, various forms can be configured using a stacked denoising autoencoder. This is divided into unsupervised learning and supervised learning according to the learning method. Early stacked denoising autoencoders were introduced by applying a restricted Boltzmann machine (RBM) in the form of a deep belief network (DBN) [41], [42]. This is used to overcome the huge amount of computation required, the local minima, and vanishing gradient problems. Currently, the use of various propagation functions and optimizers makes learning easier when using backpropagation [43], [44]. In this study, we experimented with a supervised stacked denoising autoencoder. Input x' is composed of data in which 25% of missing values (0) are randomly generated as noise. The label data consist of the original data x without missing values. The autoencoder is trained through a backpropagation with a loss of the MAE of output x^ and x generated as a result of one-time learning. Fig. 2 shows an estimation of missing data using the stacked denoising autoencoder. As shown in Fig. 2, the original data undergo a noise addition step and are converted into training data. Furthermore, with the original data being a label, weights are learned using the error in the output values of the neural network. Fig. 2 shows an illustration of a stacked denoising autoencoder consisting of one input layer, three hidden layers, and one output layer. For example, an autoencoder consisting of five hidden layers (54, 32, 16, 32, and 64) is as follows: 80(input) → 64(hidden) → 32(hidden) → 16(hidden) → 32(hidden) → 64(hidden) → 80(output), the first half of which is an encoder, ranging from 80 to 16, and the remaining half is a decoder, ranging from 16 to 80. The transactions in the source data have a missing value rate of approximately 25%. During stacked denoising autoencoder learning, the noise factor increases by 0.05, starting from 0.05 until it reaches 0.30. A rectifier linear unit [26] and Adam [27] were employed for the activation function and optimizer, respectively. The data selected in KNHNES for experiments are 14,688 data without missing value. Accordingly, in the step of Add Noise, as many input numbers as noise factor are randomly replaced by 0, and thereby a set of Noised KNHENS is created. At this time, the label is the original data without noise, and it is used to calculate an error of an output. Experimental data on 14,688 cases were randomly assigned as follows: 70% to the training data, 10% to the model validation data, and 20% to the test data. A total of 80 parameters were selected, with 80 input nodes and 80 output nodes. As the number of hidden layers increases, the repeated experiments show a lower accuracy and higher loss, which also occurs in general deep learning because healthcare data are not large in scale. Therefore, the KNHNES data show that the neural network structure is higher. The number of hidden nodes increases to half the number of input nodes, and the results of the repeated experiments show the smallest difference between the accuracy and loss of the training and verification data when 64 nodes are configured. Table 2 shows the learning results according to the autoencoder applied. In general, models containing more parameters have more feature information. The learning outcome shows that the highest accuracy and smallest loss can be achieved when a model is configured using a single hidden layer and 64 nodes. Despite the fact that the performance of the actual system decreases with an increase in the number of parameters, because the two models have only a slight difference in the number of parameters (480), the 80-64-80 model was selected for this study. The accuracy of the selected model is 0.9421, which indicates that most of the estimates for the missing data are meaningful.

C. MULTI-MODAL STACKED DENOISING AUTOENCODER FOR ESTIMATING MISSING VALUES

Various types of data are frequently utilized in the field of machine learning, and many different data types can be

TABLE 2. Learning results according to autoencoder applied.

Model	Denoising autoencoder		Stacked denoising autoencoder	
Input Layer	80		80	
Hidden	64		64-32-64	
Output Layer	80		80	
Parameter	10,384		9,904	
Noise Factor	0.25		0.25	
Data	Training	Validation	Training	Validation
Accuracy	0.9321	0.9232	0.9287	0.9111
loss	0.4152	0.4230	0.5767	0.5868

collected for a single object. Modeling can be conducted by integrating data concurrently or by applying only the necessary parameters individually. However, the structure of the data collected for an object varies depending on the observational tools. Configuring them as a single transaction may cause a data distortion or loss in terms of the relationship or features. In particular, the data available in the healthcare field may vary depending on particular circumstances in the application of the generated model. In this regard, a multi-modality has emerged as an approach in which a model generated for an object is divided into several parts based on the data characteristics.

By applying a stacked denoising autoencoder, missing data can be estimated, although learning using a single autoencoder will result in a high computational workload and loss in learning efficiency because there are numerous types of parameters used. An autoencoder is an ordinary type of neural network, and its performance varies greatly according to the learning method or configuration applied. For low-impact parameters, in particular, if the weights converge to 0, it may result in an increase in the computational workload because such convergence is of little significance, although valid values still remain. To achieve personalization and customization, healthcare models have been developed with a focus on small devices or smartphones, thus requiring an efficient model with a low computational workload [28]. In this regard, it is necessary to reconfigure the training data based on such workload by considering the classification of the parameters and to construct a multi-modal autoencoder. Therefore, in this study, a technique for estimating the missing data using a multi-modal stacked denoising autoencoder is proposed. The proposed method involves handling missing data using integrated KNHNES data in a single-modal approach for each parameter class. For this purpose, several hidden nodes are required. In this structure, each single-modal autoencoder is merged into a hidden layer and then output as each single-modal.

A total of 80 parameters consisting of existing chronic conditions, diagnosed chronic conditions, the time of the initial diagnosis showing chronic conditions, current treatments for chronic conditions, physical activities, exercise information, health examination information, dietary nutrition, and stress are selected. KNHNES data are arranged in

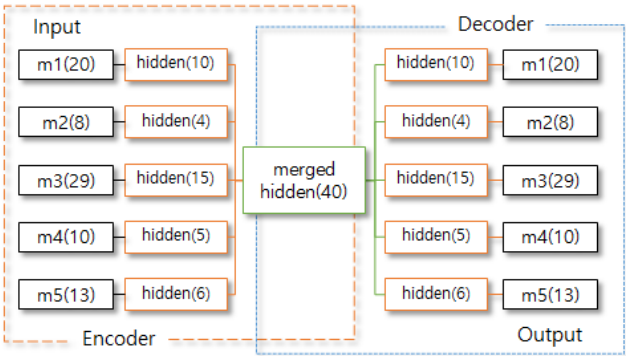

FIGURE 3. Multi-modal stacked denoising autoencoder.

a hierarchical form and can be classified into super- and sub-classes. For example, parameters sharing a common superclass can be categorized into a prevalence class, such as ''pre-existing hypertension'', ''pre-existing diabetes'', ''pre-existing hyperlipidemia'', ''diagnosed arthritis'', and ''time of the first diagnosis with hypertension.'' Single-modal data with a superclass, consisting of chronic conditions, physical activities, health examination information, dietary nutrition, and subjective health conditions, are employed in this study.

The parameters are classified into five modes: chronic conditions (m1), physical activities (m2), health examination (m3), dietary nutrition (m4), and subjective health conditions (m5), and 20, 8, 29, 10, and 13 parameters are assigned to each mode, respectively. When applying a multi-modal stacked denoising autoencoder, the structure of the hidden nodes is modified with the number of classified input nodes, whereas the parameters remain the same. All single-modal data have a different input value depending on the number of parameters classified into each class, the middle hidden value of which is smaller than the input value. The encoder side consists of 20×10 chronic conditions (m1), 8×4 physical activities (m2), 29×15 health examination data (m3), 10×4 dietary nutrition data (m4), and 13×6 subjective health conditions (m5), with an added hidden layer of 40 nodes that merge the parameters. The decoder is configured symmetrically with the encoder. Fig. 3 illustrates the multi-modal stacked denoising autoencoder. As shown in Fig. 3, layer m1(20) indicates that the single-modal chronic conditions (m1) consist of 20 hidden nodes. This configuration allows a feature extraction and learning to be conducted for each class. Moreover, all single-modal data can be separated and used as an initial value for a new neural network model.

IV. PERFORMANCE EVALUATION

A. EVALUATION RESULTS ACCORDING TO METHODS FOR ESTIMATING MISSING DATA

For validation, the proposed method is compared with existing missing data estimation methods. The experimental data are processed using a multi-modal stacked denoising autoencoder (MMSDAE), K-NN [21], SVD [22], and the column means (c.mean), and accordingly organized into the

TABLE 3. Evaluation results according to the methods used for estimating missing values.

Missing Values Estimation Method (Accuracy)					
Noise factor	SVD	K-NN	c.mean	SDAE	MMSDAE
0.05	0.9142	0.8901	0.7144	0.9675	0.9617
0.10	0.8941	0.8764	0.6751	0.9619	0.9571
0.15	0.8595	0.8490	0.6214	0.9556	0.9459
0.20	0.8012	0.7907	0.5451	0.9458	0.9345
0.25	0.7427	0.7242	0.5048	0.9321	0.9217
0.30	0.7019	0.6726	0.4191	0.8790	0.8610

training data. In the c.mean method, the mean of each column replaces a missing value. The training data are then applied to the same machine learning algorithm to generate a model, and the resulting model is evaluated. For the experimental data, 14,668 cases are prepared by pre-processing the KNHNES [16] data. Among the experimental data, 70% of the parameters are randomly assigned to the training data, 10% to the model validation data, and 20% to the test data. In the method, data do not include a missing value. For experiments, input data are randomly replaced by 0 according to the noise factor. Therefore, each model receives the input data with a virtual missing value and predicts a missing value. The training data are used to generate each model, which will be evaluated through the validation data. The test data generate a random missing data value by applying the generated model, which randomly generates missing data, and the noise factor increases by 0.05 within a range of 0.05 to 0.30. A noise factor of 0.05 indicates that the randomly generated missing data is 5%. The test data are applied to each model, and the output is compared with the original test data to identify any errors. The missing data are then compared with the original data to calculate the margin of error. In the error measurement, the true value is the original test data, and the predicted value is the restored test data. The test data input into each model is t', the output data are t^, and the correct answer data are t. The error is evaluated using the correct data t as the actual value and the output data t^ as the predicted value. For performance evaluation, each model is evaluated ten times repeatedly according to the noise factor, and the mean of accuracy values is calculated. Table 3 shows the evaluation results according to the methods used for estimating the missing data. The table also shows the accuracy of the missing data estimation for each model based on the noise factor. The missing data estimation method using an autoencoder exhibits the highest accuracy, regardless of the noise factor, and a higher noise factor results in overall lower accuracy.

B. EVALUATION RESULTS ACCORDING TO MULTI-MODAL CONFIGURATION

In machine learning, parameters such as the learning efficiency and learning time vary depending on the parameter

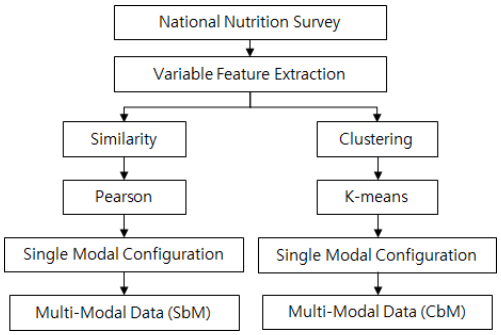

FIGURE 4. Multi-modal data configuration based on a feature analysis.

construction or training data. A parameter construction that handles all parameters using a single algorithm is simple and easy to implement [29], [30]. Unfortunately, it increases the computational workload of the algorithm and may decrease the efficiency in the actual application. Furthermore, the accuracy and computational speed may decrease as the relationships among the ill-defined input parameters are learned. Thus, for a dataset having many input parameters, a systemic strategy for parameter construction is needed to cluster parameters with a high mutual impact based on a feature analysis. An additional experiment was conducted to evaluate the performance of the models for different multi-modal configurations [31]–[33].

The characteristics of the parameters are examined through a similarity and cluster analysis. Using each estimation method, a combination of parameters is applied to construct the multi-modal data. Based on this, the following three datasets can be generated: similarity-based multi-modal data, cluster-based multi-modal data, and category-based multi-modal data. Fig. 4 shows the multi-modal data configuration based on a feature analysis. Based on the similarity [34], a cluster analysis [35], and the category construction, various multi-modal autoencoders are configured and evaluated. Therefore, in this study, we developed appropriate models for a similarity-based multi-modal stacked denoising autoencoder (SbM_SDAE), a clustering-based multi-modal stacked denoising autoencoder (CbM_SDAE), and a hierarchical structure-based multi-modal stacked denoising autoencoder (HbM_DAE), and we evaluated their performance.

HbM_DAE is a multi-modal data model based on the hierarchical architecture described in section 3.C. HbM_DAE utilizes the categories classified by KNHNES so as to design variables in multi-modal. For the fair evaluation of models, the number of hidden nodes is set to 1/2 of input nodes. Each model has a different Epoch value. The maximum Epoch is 50, and Epoch occurs at the convergent point of accuracy. SbM_SDAE ends learning at 31 Epoch, CbM_SDAE at 30 Epoch, and HbM_DAE at 27 Epoch.

Similarity analysis is a method for determining whether parameters are positively or negatively correlated with each other. These correlations are represented within a range of

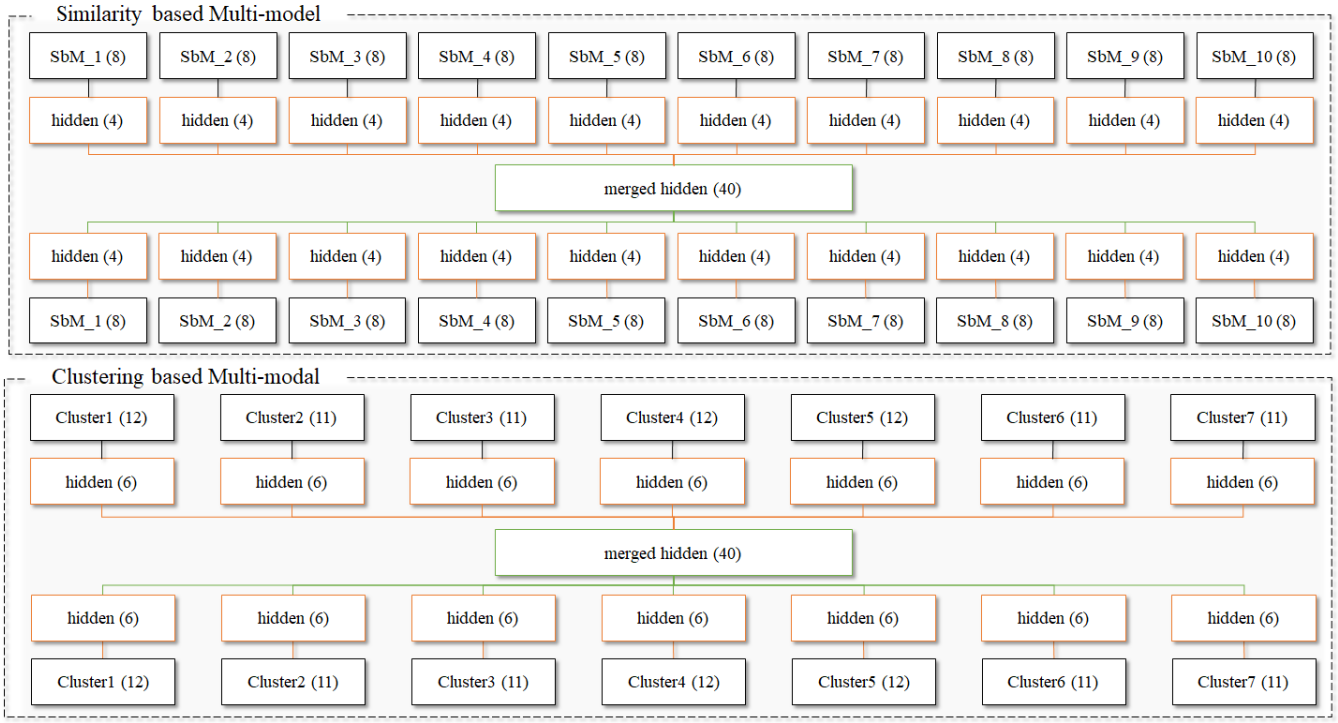

FIGURE 5. Multi-modal stacked denoising autoencoders according to data construction.

−1 to +1, in which a value closer to −1 indicates that the relationship is moving more in the opposite direction, whereas a value closer to +1 indicates that the relationship is moving more in the same direction [36], [37]. Values closer to 0 indicate a lower mutual impact. A Pearson's coefficient [34], a widely known and simple-to-implement correlation coefficient, is used in this analysis. In addition, many other correlation coefficients are available for user convenience and different types of data. Cluster analysis involves grouping a set of parameters in proximity to each other in terms of the vector space.

In this analysis, the k-means algorithm is applied [35], and the k value at which the entire cluster achieves the minimum variance is selected. A categorical analysis is a method for categorizing the parameters that have been classified as being of the same class in the KNHNES data. This is the most common and basic clustering algorithm available and exhibits excellent performance, particularly considering the difficulties in its implementation. In addition, many other methods are available for constructing multi-modal data. Performance evaluation of the proposed method applied to a multi-modal data construction is conducted in an attempt to determine its effects on the accuracy of the model and to identify the need for diversity when selecting an algorithm according to the different datasets used.

Fig. 5 shows the multi-modal stacked autoencoders according to the data construction. A similarity-based data configuration using the Pearson's correlation coefficient is used to randomly select a parameter, and eight parameters having a high absolute value of similarity are clustered. This process is then repeated using the parameters other than those already selected.

TABLE 4. Learning results of multi-modal stacked denoising autoencoder.

SbM		CbM		HbM	
Parameter	5,032	Parameter	6,488	Parameter	4,790
Noise Factor	Accuracy	Noise Factor	Accuracy	Noise Factor	Accuracy
0.05	0.9601	0.05	0.9577	0.05	0.9617
0.10	0.9559	0.10	0.9510	0.10	0.9571
0.15	0.9412	0.15	0.9426	0.15	0.9459
0.20	0.9313	0.20	0.9312	0.20	0.9345
0.25	0.9209	0.25	0.9241	0.25	0.9217
0.30	0.8497	0.30	0.8550	0.30	0.8610

As a result, a total of ten groups consisting of eight parameters are generated. Using k-means clustering [35], a value of 7 is used as the k value. With the Bouldin index, clustering results are evaluated, and thus the optimal k value is found. A category-based data configuration consists of five groups according to the data label, such as a physician diagnosis (dg) and health examination (HE). As shown in Fig. 5, the SbM_1(8) layer indicates that group 1 of the similarity-based multi-modal data has eight input nodes.

During multi-modal autoencoder learning, all conditions remain the same, except for the number of hidden nodes, depending on the parameter configuration. The multi-modal stacked denoising autoencoder consists of three hidden layers in a single-modal. Each single-modal needs to be

integrated, and to this end, the total number of weights is adjusted. Table 4 shows the learning results of the multi-modal stacked denoising autoencoder. As indicated in Table 4, the parameters are determined based on the configuration of the autoencoder's input, hidden, and output nodes and the HbM appears to be the smallest, indicating that this model requires the least number of resources for actual use.

Moreover, HbM also shows the highest accuracy according to the noise factor. This can be explained based on the fact that the classes categorized in the KNHNES [16] data consider real-world situations. If a dataset with ill-defined classes is used to construct multi-modal data, it may yield different outcomes.

V. CONCLUSIONS

The proposed method involves inputting the missing data into the PHRs by using a multi-modal stacked denoising autoencoder trained using KNHNES data. A total of 80 parameters are selected from the preprocessed KNHNES data. A single-modal denoising autoencoder is trained using the selected parameters to estimate the missing data. The results show that the accuracy of the proposed method is 0.9321 when a noise factor of 0.25 is applied. Given that KNHNES or PHR data have a multi-modality feature, in this study, a multi-modal stacked denoising autoencoder is constructed. For this purpose, 80 selected parameters are classified into 5 single-modals. The multi-modal stacked denoising autoencoder is trained using the classified parameters, and the resulting accuracy is 0.9217 when the noise factor is 0.25; this accuracy is higher than that of other ordinary missing data estimation methods such as SVD, K-NN, and the column mean. For a single-modal feature, 10,384 parameters are generated, whereas, for a multi-modal feature, 4,790 parameters are generated. This indicates that the multi-modal stacked denoising autoencoder is more suitable for a personal device than a single-modal feature because the number of parameters for a multi-modal feature is nearly half, with only a slight difference in accuracy. Healthcare data may be composed of all missing data according to the particular situation of the user. In this case, it may be possible to predict the missing data using a pre-learned autoencoder. In addition, a difference in the learning efficiency occurs depending on the configuration of the variables. A modality configuration that clearly shows the characteristics of the variables improves the learning efficiency and allows more accurate neural network models to be generated. Further studies are required to examine and evaluate the effects of missing data estimated using a multi-modal stacked denoising autoencoder on healthcare-related machine learning models.

REFERENCES

[1] J. S. R. Jang, C. T. Sun, and E. Mizutani, "Neuro-fuzzy and soft computing—A computational approach to learning and machine intelligence [Book Review]," *IEEE Trans. Autom. Control*, vol. 42, no. 10, pp. 1482–1484, Oct. 1997.
[2] S. Mitra, S. K. Pal, and P. Mitra, "Data mining in soft computing framework: A survey," *IEEE Trans. Neural Netw.*, vol. 13, no. 1, pp. 3–14, Jan. 2002.
[3] F. S. Collins and H. Varmus, "A new initiative on precision medicine," *New England J. Med.*, vol. 372, no. 9, pp. 793–795, Feb. 2015.
[4] J. Kim, J. Kim, D. Lee, and K.-Y. Chung, "Ontology driven interactive healthcare with wearable sensors," *Multimedia Tools Appl.*, vol. 71, no. 2, pp. 827–841, Jul. 2014.
[5] L. C. Huang, H. C. Chu, C. Y. Lien, C. H. Hsiao, and T. Kao, "Privacy preservation and information security protection for patients' portable electronic health records," *Comput. Biol. Med.*, vol. 39, no. 9, pp. 743–750, 2009.
[6] A. Nasrollahi, W. Deng, Z. Ma, and P. Rizzo, "Multimodal structural health monitoring based on active and passive sensing," *Struct. Health Monitor.*, vol. 17, no. 2, pp. 395–409, Mar. 2018.
[7] W. Wang, B. C. Ooi, X. Yang, D. Zhang, and Y. Zhuang, "Effective multi-modal retrieval based on stacked auto-encoders," *Proc. VLDB Endowment*, vol. 7, no. 8, pp. 649–660, Apr. 2014.
[8] H. Jung and K. Chung, "Knowledge-based dietary nutrition recommendation for obese management," *Inf. Technol. Manage.*, vol. 17, no. 1, pp. 29–42, Mar. 2016.
[9] J.-C. Kim and K. Chung, "Mining health-risk factors using PHR similarity in a hybrid P2P network," *Peer-to-Peer Netw. Appl.*, vol. 11, no. 6, pp. 1278–1287, Nov. 2018.
[10] R. Agrawal and R. Srikant, "Mining sequential patterns," in *Proc. 11th Int. Conf. Data Eng.*, Mar. 1995, pp. 3–14.
[11] K. Chung, J.-C. Kim, and R. C. Park, "Knowledge-based health service considering user convenience using hybrid Wi-Fi P2P," *Inf. Technol. Manage.*, vol. 17, no. 1, pp. 67–80, Mar. 2016.
[12] H. Yoo and K. Chung, "Mining-based lifecare recommendation using peer-to-peer dataset and adaptive decision feedback," *Peer-to-Peer Netw. Appl.*, vol. 11, no. 6, pp. 1309–1320, Nov. 2018.
[13] S.-Y. Oh, K. Chung, and J.-S. Han, "Towards ubiquitous health with convergence," *Technol. Health Care*, vol. 24, no. 3, pp. 411–413, May 2016.
[14] H. Yoo and K. Chung, "PHR based diabetes index service model using life behavior analysis," *Wireless Pers. Commun.*, vol. 93, no. 1, pp. 161–174, Mar. 2017.
[15] E.-Y. Jung, J. Kim, K.-Y. Chung, and D. K. Park, "Mobile healthcare application with EMR interoperability for diabetes patients," *Cluster Comput.*, vol. 17, no. 3, pp. 871–880, Sep. 2014.
[16] *Korean National Health and Nutrition Examination Survey, Korea Centers for Disease Control and Prevention*. Accessed: Mar. 2020. [Online]. Available: https://knhanes.cdc.go.kr/
[17] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Proc. Adv. Neural Inf. Process. Syst.*, 2012, pp. 1097–1105.
[18] M. E. Duffy, "Handling missing data: A commonly encountered problem in quantitative research," *Clin. Nurse Spec.*, vol. 20, no. 6, pp. 273–276, Nov. 2006.
[19] D. B. Rubin and N. Schenker, "Multiple imputation in health-are databases: An overview and some applications," *Statist. Med.*, vol. 10, no. 4, pp. 585–598, Apr. 1991.
[20] Z. Che, S. Purushotham, K. Cho, D. Sontag, and Y. Liu, "Recurrent neural networks for multivariate time series with missing values," *Sci. Rep.*, vol. 8, no. 1, p. 6085, Dec. 2018.
[21] K. Hron, M. Templ, and P. Filzmoser, "Imputation of missing values for compositional data using classical and robust methods," *Comput. Statist. Data Anal.*, vol. 54, no. 12, pp. 3095–3107, Dec. 2010.
[22] M. Kurucz, A. A. Benczúr, and K. Csalogány, "Methods for large scale SVD with missing values," in *Proc. KDD Cup Workshop*, vol. 12, 2007, pp. 31–38.
[23] J.-C. Kim and K. Chung, "Neural-network based adaptive context prediction model for ambient intelligence," *J. Ambient Intell. Humanized Comput.*, vol. 11, no. 4, pp. 1451–1458, Apr. 2020.
[24] S. García, J. Luengo, and F. Herrera, *Data Preprocessing in Data Mining* New York, NY, USA: Springer, 2015, pp. 59–139.
[25] P. Vincent, H. Larochelle, Y. Bengio, and P.-A. Manzagol, "Extracting and composing robust features with denoising autoencoders," in *Proc. 25th Int. Conf. Mach. Learn. (ICML)*, 2008, pp. 1096–1103.
[26] X. Glorot, A. Bordes, and Y. Bengio, "Deep sparse rectifier neural networks," in *Proc. 14th Int. Conf. Artif. Intell. Statist.*, 2011, pp. 315–323.
[27] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," 2014, *arXiv:1412.6980*. [Online]. Available: http://arxiv.org/abs/1412.6980

[28] K. Chung and R. C. Park, "PHR open platform based smart health service using distributed object group framework," *Cluster Comput.*, vol. 19, no. 1, pp. 505–517, Mar. 2016.
[29] I. H. Witten, E. Frank, M. A. Hall, and C. J. Pal, *Data Mining: Practical Machine Learning Tools and Techniques*. San Mateo, CA, USA: Morgan Kaufmann, 2016.
[30] C. Cadena, A. Dick, and I. D. Reid, "Multi-modal auto-encoders as joint estimators for robotics scene understanding," in *Proc. 12th Robot., Sci. Syst.*, 2016, pp. 1–9.
[31] J.-W. Baek, J.-C. Kim, J. Chun, and K. Chung, "Hybrid clustering based health decision-making for improving dietary habits," *Technol. Health Care*, vol. 27, no. 5, pp. 459–472, Sep. 2019.
[32] J.-C. Kim and K. Chung, "Mining based time-series sleeping pattern analysis for life big-data," *Wireless Pers. Commun.*, vol. 105, no. 2, pp. 475–489, Mar. 2019.
[33] J. C. Kim and K. Chung, "Prediction model of user physical activity using data characteristics-based long short-term memory recurrent neural networks," *KSII Trans. Internet Inf. Syst.*, vol. 13, no. 4, pp. 2060–2077, Apr. 2019.
[34] J. Benesty, J. Chen, Y. Huang, and I. Cohen, "Pearson correlation coefficient," in *Noise Reduction in Speech Processing*. Berlin, Germany: Springer, 2009, pp. 1–4.
[35] J. A. Hartigan and M. A. Wong, "Algorithm AS 136: A k-means clustering algorithm," *J. Roy. Stat. Soc. C, Appl. Statist.*, vol. 28, no. 1, pp. 100–108, 1979.
[36] K. Chung and J. Kim, "Activity-based nutrition management model for healthcare using similar group analysis," *Technol. Health Care*, vol. 27, no. 5, pp. 473–485, Sep. 2019.
[37] H. Jung, J. Yang, J.-I. Woo, B.-M. Lee, J. Ouyang, K. Chung, and Y. Lee, "Evolutionary rule decision using similarity based associative chronic disease patients," *Cluster Comput.*, vol. 18, no. 1, pp. 279–291, Mar. 2015.
[38] M. Shang, X. Luo, Z. Liu, J. Chen, Y. Yuan, and M. Zhou, "Randomized latent factor model for high-dimensional and sparse matrices from industrial applications," *IEEE/CAA J. Automatica Sinica*, vol. 6, no. 1, pp. 131–141, Jan. 2019.
[39] X. Luo, M. Zhou, Z. Wang, Y. Xia, and Q. Zhu, "An effective scheme for QoS estimation via alternating direction method-based matrix factorization," *IEEE Trans. Services Comput.*, vol. 12, no. 4, pp. 503–518, Jul. 2019.
[40] X. Luo, M. Zhou, S. Li, and M. Shang, "An inherently nonnegative latent factor model for high-dimensional and sparse matrices from industrial applications," *IEEE Trans. Ind. Informat.*, vol. 14, no. 5, pp. 2011–2022, May 2018.
[41] J. Qiao, G. Wang, W. Li, and M. Chen, "An adaptive deep Q-learning strategy for handwritten digit recognition," *Neural Netw.*, vol. 107, pp. 61–71, Nov. 2018.
[42] G. Jiang, H. He, P. Xie, and Y. Tang, "Stacked multilevel-denoising autoencoders: A new representation learning approach for wind turbine gearbox fault diagnosis," *IEEE Trans. Instrum. Meas.*, vol. 66, no. 9, pp. 2391–2402, Sep. 2017.
[43] F. Gu, K. Khoshelham, S. Valaee, J. Shang, and R. Zhang, "Locomotion activity recognition using stacked denoising autoencoders," *IEEE Internet Things J.*, vol. 5, no. 3, pp. 2085–2093, Jun. 2018.
[44] Y.-J. Zheng, S.-Y. Chen, Y. Xue, and J.-Y. Xue, "A pythagorean-type fuzzy deep denoising autoencoder for industrial accident early warning," *IEEE Trans. Fuzzy Syst.*, vol. 25, no. 6, pp. 1561–1575, Dec. 2017.
[45] K. Wang, C. Gou, Y. Duan, Y. Lin, X. Zheng, and F.-Y. Wang, "Generative adversarial networks: Introduction and outlook," *IEEE/CAA J. Automatica Sinica*, vol. 4, no. 4, pp. 588–598, Nov. 2017.

JOO-CHANG KIM received the B.S. and M.S. degrees from the School of Computer Information Engineering, Sangji University, South Korea, in 2014 and 2016, respectively. He is currently pursuing the Ph.D. degree with the Department of Computer Science, Kyonggi University, South Korea. He has been a Researcher with the Data Mining Laboratory, Kyonggi University. His research interests include data mining, data management, knowledge systems, machine learning, deep learning, big data, healthcare, and recommendation systems.

KYUNGYONG CHUNG received the B.S., M.S., and Ph.D. degrees from the Department of Computer Information Engineering, Inha University, South Korea, in 2000, 2002, and 2005, respectively. He has worked with the Software Technology Leading Department, Korea IT Industry Promotion Agency (KIPA). From 2006 to 2016, he was a Professor with the School of Computer Information Engineering, Sangji University, South Korea. Since 2017, he has been a Professor with the Division of Computer Science and Engineering, Kyonggi University, South Korea. His research interests include data mining, artificial intelligence, healthcare, knowledge systems, HCI, and recommendation systems.