Hindawi
Mathematical Problems in Engineering
Volume 2022, Article ID 8544122, 12 pages
https://doi.org/10.1155/2022/8544122

Research Article

Sparse-Coding-Based Autoencoder and Its Application for Cancer Survivability Prediction

Gang Huang, Hailun Wang, and Lu Zhang

College of Electrical and Information Engineering, Quzhou University, Quzhou 324000, China

Correspondence should be addressed to Gang Huang; huangg@qzc.edu.cn

Received 16 November 2021; Accepted 27 December 2021; Published 4 February 2022

Academic Editor: Muhammad Faisal Nadeem

Cancer-survivability prediction is one of the popular research topics, that attracted great attention from both the health service providers and academia. However, one remaining question comes from how to make full use of a large number of available factors (or features). This paper, accordingly, presents a novel autoencoder algorithm based on the concept of sparse coding to address this problem. The main contribution is twofold: the utilization of sparsity coding for input feature selection and a subsequent classification using latent information. Precisely, a typical autoencoder architecture is employed for reconstructing the original input. Then the sparse coding technique is applied to optimize the network structure, with the aim of selecting optimal features and enhancing the generalization capability. In addition, the refined latent information is further cast as alternative features for training a sparse classifier. To evaluate the performance of the proposed autoencoder architecture, we present a comprehensive analysis using a publicly available data repository (*i.e.*, Surveillance, Epidemiology, and End Results, SEER). Experimental study shows that the proposed approach has the ability of extracting important features from high-dimensional inputs and achieves competitive performance than other state-of-the-art classification techniques.

1. Introduction

Cancer is the second major cause of death globally, according to World Cancer Report (http://publications.iarc.fr/Non-Series-Publications/World-Cancer-Reports/World-CancerReport-2014). For example, more than 9.6 million deaths were reported, associated with the cancer disease in 2018. Due to its huge economical and health influence, cancer survivability prediction has received a lot of attention in the last decades. In general, the task of survivability prediction refers to estimate the life span and/or duration of patients according to their historical information. This research question has been of great interest to either healthcare providers and individual patient as the prediction results provide an effective suggestion and/or measurement to evaluate the prognosis and reduce the significant suffer. Not surprisingly, a great number of research effort has been dedicated to develop numerous prediction models. Section 2 provides more details in this regard.

The majority prediction models follow into the pipeline of "data pre-processing" and then "classification." The former aims to manipulate the original data to form a more manageable input, by the operation of missing-value imputation, important feature extraction/selection, outlier removal, *etc.* The latter process refers to specially design classification algorithms, such as Support Vector Machine (SVM), Neural Network (NN), and Random Forest (RF), to establish a suitable mapping relation between input samples and output labels. Among them, we argue that the most-desired aspect is "data pre-processing" as noisy inputs can have significantly negative impact on the subsequent "classification" process. In this context, the major motivation of this study is to propose an innovative and practical method for preprocessing input samples. That is, we aim to represent original inputs with a cleaner and more suitable data alternative, which in return could improve the following "classification" performance.

The autoencoder algorithm has received a great deal of attention over the last decades, as a group member of Deep

Learning family. Similar to a standard Multiple-Layer Perceptron (MLP), the architecture of a basic autoencoder is a three-layer network, including the layer of input, hidden, and output, respectively. Compared to MLPs, the major difference is that the output from standard autoencoders is equivalent to the input. As such, the autoencoder algorithm is characterized by its learning (representation) capability of input data, while preserving the most important information. Due to this advanced learning capability, relevant applications of autoencoder have spanned several disciplines, such as pattern recognition, decision-making, and statistical modeling. A brief review of autoencoder techniques will be given in the next section.

In spite of the general research interest in developing autoencoder-based applications, there are still research questions remaining to apply autoencoders for establishing prediction models. Firstly, conventional autoencoders are designed for data clustering or reconstruction purposes, not explicitly for classification. Therefore, we cannot directly apply autoencoders to build a classification model for predicting cancer survivability. Secondly, in the context of the medical data, there are usually a large number of available features. One critical problem then is to determine optimal features, using autoencoders, to achieve satisfactory classification performance. Finally, there is not a reliable strategy to determine the autoencoder structure; consequently, the cross-validation or trial-and-error method is usually performed which is very time-consuming.

To solve the aforementioned issues, this paper explores the applicability of autoencoders to establish a two-stage prediction model, via data reconstruction and classification using the latent information, respectively. At the first stage of data reconstruction, we consider to apply the concept of sparse coding to extract informative features while optimizing the network structure. The concept of sparse coding is to represent a target signal using a linear combination of a few elementary signals. As such, only few nonzero elements are capable of capturing or reconstructing the target signal. Sparse coding has therefore become a very active research area and has found its wide applications in many areas, such as machine-learning [1–3], and IoT applications [4], *etc*. At the second classification stage, we tempt to build a prediction model using extracted information from the first stage. In particular, the latent information (*i.e*., the output from the hidden layer) is cast as the training input. At last, this two-stage prediction model is formulated and represented using one unique objective function. We further employ an iterative computational strategy for optimizing this proposed function.

With this end in view, a novel autoencoder algorithm is introduced to facilitate the modeling of cancer survivability prediction. To evaluate its performance, a real-world data repository is introduced. Specifically, we apply the proposed algorithm to form a prediction model in the cancer survivability context. In comparison to other prediction algorithms, experimental results show that the proposed method achieves better classification results. In summary, this paper presents the following contributions:

(i) The sparse coding technique is introduced for autoencoders, with the aim of performing feature selection and data reconstruction simultaneously; in particular, the feature selection is done by determining features that contribute most to the subsequent classification

(ii) Instead of using the raw inputs, the latent information (*i.e*., the output from the hidden layer of autoencoders) is manipulated as the training inputs for classification

(iii) The training process of the proposed autoencoder algorithm is formulated and represented using one unique objective function, which is later solved using an iterative computational strategy

The remainder of this paper is organized as follows. Firstly, we briefly review some background concepts and related studies in Section 2. Next, the detail implementation of the proposed algorithm is discussed and evaluated in Section 3 and Section 4, respectively. Finally, we provide some remarks and future works in Section 5.

2. Background Concepts and Related Works

In this section, two main aspects studied in this paper are discussed, including existing applications for predicting cancer survivability and the conventional autoencoder algorithm.

2.1. Existing Prediction Models. Cancer is reported as the second major cause of human death globally. Apart from its fatality, this disease also leads to a huge economic impact. According to Ref. [5], healthcare expenditure is expected to reach $9.1 trillion by 2023. Due to its large economic and social impact, healthcare providers and authorities have been spending a lot of effort on cancer-related research, including effective treatment, accurate diagnose with early time and less cost, medical imaging, *etc*.

Among them, patient survivability prediction has been of great interest and significance to the medical industry. By integrating data from the patient's medical history and external risk factors, survivability prediction aims to estimate the living-span probability for one particular patient. Not surprisingly, machine-learning algorithms, such as Decision Tree (DT), Random Forest (RF), Support Vector Machine (SVM), and Artificial Neural Network (ANN), have been employed to address this survivability-prediction problem.

The DT and RF algorithms are two of the mainstream methods used in cancer survival-rate prediction field, due to their high interpretability and accuracy. On the one hand, DT is the fundamental technique of RF, which identifies and chooses significant features (using the criteria of information gain and Entropy) that are helpful in the classification. On the other hand, RF is a bootstrapping algorithm with the DT model, in which RF tries to build several different decision trees by choosing a random subset of training samples and features. At last, as an ensemble learning strategy, RF

combines all individual subtree results and makes the final prediction. The work from Ref. [6] establishes a DT-based prediction model by separating cancer patients into different groups based on their age and gender. Additionally, their results also identify two high-risk groups, that is, the female group with the age between 42 and 52 and the male group with their ages less than 42. Those findings can be further used to assist clinicians with useful guidance to provide better individual treatment plan. In Ref. [7], four different DT variants have been employed, including classification and regression trees (CART), the quick, unbiased, efficient statistical tree (QUEST), chi-square automatic interaction detector (CHAID) algorithm, and the C5.0 algorithm. Experiments have been performed using a dataset of 500 patient records and 13 features, and the results indicate that the C5.0 algorithm achieves the highest classification accuracy (86.4%). Additionally, they also conclude that factors of diagnosis age, histologic grade, axillary lymph node status, and type of surgery statistically impact on patient survivability. In the study of Ref. [8], another dataset with 6000 patient records and 23 features is employed. Several algorithms of DT, RF, ANN, and SVM are introduced, while experimental results show that the RF-based method generates the best result (approximately 82.7% classification accuracy). The work further identified several important features, such as cancer stages, Tumor Size, number of axillary lymph node, and types of primary treatment. Similarly, three datasets from UCI ML repositories are employed in the study of Ref. [9] to investigate the survivability-prediction problem. Again, their research results also reveal that RF leads to the best performance with a classification accuracy of 98%.

SVM is another popular machine-learning algorithm and has received a great deal of research attention in the medical domain. By projecting raw input data into a high-dimension feature space, SVM is able to identify complex nonlinear relationship to classify training samples. Both of the work from Refs. [10, 11] explore the SVM-based prediction model. On the one hand, the sequential minimal optimization technique is applied with SVM in Ref. [11]. Through a sensitivity analysis, they are able to gauge the prioritized importance of predictive factors in their prediction model. On the other hand, the work from Ref. [10] compares SVM and other two techniques, that is, logistic regression and decision tree via modeling a 1340-record dataset. Their results indicate that the SVM provides the best outcome for constructing the prediction model.

Another widely used technique is the ANN, which has a multilayer structure with many perceptrons. ANNs follow the principle of supervised learning, that is, the network is trained to explore the correlation between input and output variables, and to minimize the error between the actual and desired output. To achieve a stable prediction accuracy, Wang et al. apply ANNs in a cross-validation way by trailing different sizes of neural networks [12]. More precisely, they employ a typical three-layer architecture, while the number of hidden neurons is changed within the range of 5 to 15. The best outcome is approximately 85% classification accuracy and 0.79 of Area under Receiver Operating Characteristic curve (AUC). In addition, the work from Ref. [13] explores the prognosis of cancer recurrence using the ANN. They manually select a patient dataset with twenty attributes and apply the conventional ANN for modeling. The result shows that the ANN has achieved the best performance, also provides new insights of prognostic factors which need to be observed by medical experts.

The aforementioned methods are based on the concept of supervised learning, under the assumption that the label information is known. However, in reality, the number of labeled data can be very limited and labor expensive to acquire. In this context, some research work has been performed to investigate the possibility of dealing with the unlabeled data, also known as the process of semisupervised learning. The main idea is to exploit the knowledge from unlabeled data and integrate with available label information, which has shown some promising outcomes. For example, the work from Ref. [14] proposes a prediction model by cotraining with semi- and supervised methods, including ANN and SVM. Their result shows that the semisupervised method enhances the prediction stability via the noise reduction and results in the best performance of 0.81 AUC. More recently, a low-rank and sparse representation-based algorithm has been proposed [3] to address the noisy data. More precisely, the original input has been manipulated and its low-rank and sparse representation has been estimated. A sparse classifier was also considered afterwards which achieves the promising prediction results.

Despite some accurate prediction models, the raw health data are usually subject to different types of noise, including outlier, missing values, etc. Not surprisingly, the corrupted data render many machine-learning algorithm failing to estimate the accurate patients' survivability.

2.2. Basic Autoencoder Method. Deep Learning techniques have increasingly found their wide applications in many areas, due to their accurate and robust performance. Relevant work has spanned several disciplines, such as object detection, statistical modeling, and nature language processing. The autoencoder algorithm is a particular case of Deep Learning techniques, which has a similar network architecture like the Multiple-Layer Perceptrons (MLPs). More precisely, a basic autoencoder is a three-layer fully connected network with one hidden layer, one input layer, and one output layer. The major difference, compared to MLP, is that autoencoder aims to reconstruct the raw inputs; in other words, the output from autoencoders is optimized in such a way that it should be as close/similar as the original inputs. As such, the basic autoencoders also become one popular alternative in unsupervised fashion. The general architecture of an autoencoder is then illustrated in Figure 1. As observed, the input and hidden layers play a role of encoding and representing input data into low-dimensional representation. As such, the combination of input and hidden layers is also called as "encoder." By contrast, the hidden and output layers are designed to reconstruct the input data, which play a role of the "decoder."

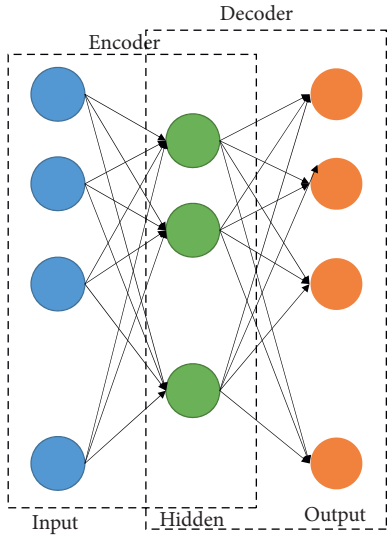

Figure 1: The network structure of a basic autoencoder.

Assume that the given L training samples are arranged as a matrix, where the i-th row represents the i-th input, that is, $X = [\mathbf{x}_1; \mathbf{x}_2; \ldots; \mathbf{x}_L]$, and ; represents the concatenation operation. In addition, let $Z \in \mathbb{R}^{L \times N}$ denote the output matrix of the hidden layer. Consequently, the calculation of Z (*i.e.*, output from the hidden layer) is as follows:

$$Z = f(XW), \tag{1}$$

where $f(\cdot)$ is the activation function and W is the weight matrix between the input and hidden layer. Furthermore, given the activation function $g(\cdot)$ from the output layer, the final output of this network can be expressed as follows:

$$\widehat{X} = g(ZV), \tag{2}$$

where V represents the weights between the hidden and output layer.

Note that the aim of autoencoders is to reconstruct the input data. As such, its training process is to minimize the error between the actual output (from the autoencoder) and the input matrix X, which can simply be expressed as follows:

$$\min Loss(X, \widehat{X}) = \sum_{l}^{L} \left\| \mathbf{x}_l - \widehat{\mathbf{x}}_l \right\|_2^2. \tag{3}$$

For training purposes, the algorithm of Back Propagation (BP) can be applied [15], in which the output error is utilized for updating weights. Let $E(t)$ represent the network, that is, $E(t) = Loss(X, \widehat{X})$ at the t-th iteration. Then, the gradient $\nabla E(t)$ with respect to network weights can be computed as $\nabla \mathbb{E}_W(t) = \partial \mathbb{E}(t)/\partial W(t)$ (for encoder) and $\nabla \mathbb{E}_V(t) = \partial \mathbb{E}(t)/\partial V(t)$ (for decoder), respectively. Accordingly, using the BP training algorithm, weights of the autoencoder can be simply updated as follows: $W(t+1) = W(t) - \alpha_1 \nabla \mathbb{E}_W(t)$ and $V(t+1) = V(t) - \alpha_2 \nabla \mathbb{E}_V(t)$, where α_1 and α_2 is the learning rate.

Recent years have witnessed the rapid development of the autoencoder-based methods and their wide applications in several domains, including cancer prediction [16], control engineering [17, 18], fault diagnosis [19], building energy management [20], and many more. In the work of Ref. [16], a stacked sparse autoencoder-based (SSAE) algorithm is proposed for cancer prediction. This method consists of the unsupervised feature extraction and the supervised classification stage, which is similar to our proposed method in terms of the overall idea. However, the main difference is that SSAE applies the sparsity constraint on each hidden neuron, rather than weights, while this constraint is further determined by a user-defined sparsity parameter ρ. As such, the performance of SSAE is impacted by ρ, which could be subjected to biased results. Loy Benitez *et al.* propose a memory-gated recurrent neural networks-based autoencoder (MG-RNN-AE), that is applied for indoor air quality (IAQ) control [18]. The main contribution is to replace the dense layers (from the traditional autoencoder structure) with the recurrent neural network, which benefits from identifying the correlation between each feature. The application of MG-RNN-AE is evaluated using an IAQ data from a D-subway station, showing its effectiveness of monitoring and controlling. A semi-supervised autoencoder (termed as discriminant autoencoder, DAE) is present in Ref. [19] for fault diagnosis. This method introduces a modified loss function based on the theory of mutual information (MI) to find a more appropriate representation. More precisely, apart from the data reconstruction error, two more constraints are employed in the modified loss function, including the l_2-norm regularization on weights and a MI-based distance measurement. The former is used to avoid over-fitting, while the latter one aims to increase the interclass separability and retains discriminant features. Compared to the proposed method, both the MG-RNN-AE and DAE apply different strategies to reconstruct the original data, while they still employ the fully connected manner without optimizing the network structure. At last, the work from Ref. [20] investigates the autoencoder method for anomaly detection in building energy management. An autoencoder-based ensemble method is proposed, by considering different architectures and training schemes. By examining the performance of different types and training methods of autoencoders, a comprehensive experiment is conducted, to provide insights into the applicability of autoencoders in detecting anomalies. Results indicate that autoencoder-based method can greatly alleviate the data preprocessing workload and maintain the data quality for further analysis. However, this paper applies the cross-validation and human expertise to manually determine the network architectures. By contrast, the proposed method introduces the sparse coding technique to automatically optimize the architecture, which is our main contribution.

3. Proposed Autoencoder Algorithm

This section details the proposed algorithm by integrating the concept of sparse coding with the autoencoder (termed as SRA), and the main pipeline is also demonstrated in Figure 2. Formally, the proposed algorithm is designed to (i)

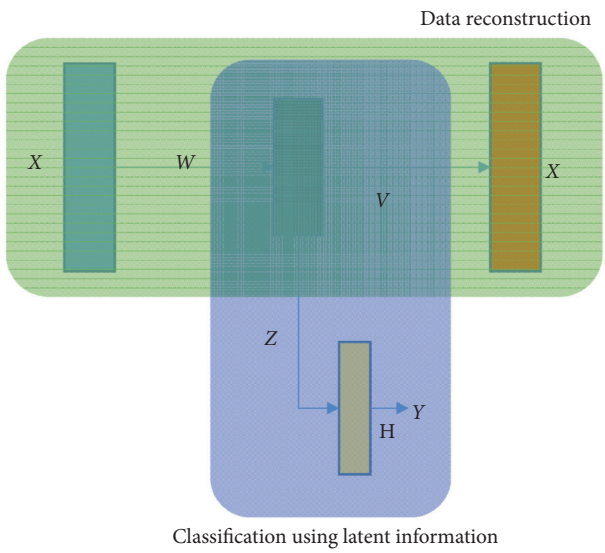

FIGURE 2: The overall pipeline of the proposed sparse autoencoder (SRA) algorithm.

identify the informative features; (ii) optimize the network structure; and (iii) to train the classifier simultaneously.

3.1. Problem Formulation. Consider a typical architecture of the autoencoder with a three-layer fully connected network. Let X be the input data matrix (where the i-th row is the i-th sample), and $Y = [\mathbf{y}_1, \mathbf{y}_2, \ldots, \mathbf{y}_L]$ be the desired output matrix of the training samples, and $\mathbf{y}_i$ is the corresponding label of $\mathbf{x}_i$. To train a classifier that produce the correct mapping relationship of (X, Y), the proposed sparse-coding-based algorithm (*i.e.*, SRA) consists of two stages: data reconstruction and classification using latent information.

3.1.1. Stage One: Data Reconstruction. Following the basic concept of autoencoders, the proposed method reconstructs the input data at its output. Furthermore, SRA utilizes the sparse coding technique to minimize the number of connections (*i.e.*, internal weights) for both of the encoder and decoder. As such, SRA not only minimizes the data reconstruction error but also rewards the sparse structure to identify informative features.

3.1.2. Stage Two: Classification Using Latent Information. We propose to use the output from the encoder (*i.e.*, latent information) to establish the classifier. Additionally, the dictionary-learning method is also leveraged to train and update this latent information according to the classification and reconstruction error.

3.2. Data Reconstruction. In the data reconstruction stage, the proposed SRA method aims to generate an output as close as the input (which is similar to the traditional autoencoders). Furthermore, to identify the informative features from the original inputs, we propose to minimize the number of weight connections within the input-hidden layer (or encoder). That is based on the fact that one single input neuron (in autoencoders) is associated with one particular feature.

Recall that each column from the input matrix X is associated with one feature; accordingly, rows from W are the relevant feature weights. Identifying an important feature from X is equivalent to setting all its associated weights to nonzero values; by contrast, skipping one feature requires setting all relevant weights to zeros. In other words, such a process of identifying important features is equivalent to minimizing the matrix sparsity for W (with only few important rows with nonzero values). In general, the sparsity of the matrix can be expressed using a mix of l_2 and l_1 norms. That is, let W_q denote the q-th row of W, then the matrix sparsity can be defined as $\|W\|_{2,1} = \sum_q \|W_q\|_2$.

Furthermore, without loss of generality, we assume that $f(\cdot)$ is an one-to-one activation function, which indicates it is invertible (*i.e.*, $f^{-1}(\cdot)$). Consequently, the feature identification process within the encoder can be cast as solving the following optimization problem:

$$\min \|W\|_{2,1} \text{ subject to } f^{-1}(Z) = XW, \tag{4}$$

For the decoder part, we apply the similar strategy to reduce the weight connection between the hidden-and-output layer, as well as minimize the error between the actual output and the input data. Meanwhile, we also assume that the active function $g(\cdot)$ is invertible (*i.e.*, $g^{-1}(\cdot)$ exists as $f^{-1}(\cdot)$). As such, we have

$$\min \|V\|_{2,1} \text{ subject to } g^{-1}(X) = ZV, \tag{5}$$

where V is the weight matrix between the hidden-and-output layer (or the decoder). By combining the loss function from both the encoder and decoder, we can formulate the following objection function for the proposed data-reconstruction process:

$$\min \|W\|_{2,1} + \|V\|_{2,1} \text{subject to} \left\| f^{-1}(Z) - XW \right\|_2^2 + \left\| g^{-1}(X) - ZV \right\|_2^2 \le \mathbf{e}, \tag{6}$$

where **e** represents the error boundary. Again, the proposed data-reconstruction process is similar to that of conventional autoencoder. The major difference, nevertheless, is that we further minimize the number of internal weights within the encoder and decoder, that is, apply the sparse coding to the network structure.

3.3. Classification Using Latent Information. During the previous stage, the proposed algorithm generates an output similar to the input as well as optimizing the network structure by minimizing the number of weights. At this stage, the latent information (*i.e.*, the output matrix Z from the encoder) is regarded as the training data for the classification purposes. We argue that the original data consider noise or outlines, which might not be suitable for training. After the process of data reconstruction, the encoder is trained to capture the characteristic of the original input. Consequently, we can utilize this latent information for training a classifier. Towards this end, assume that the desired output is Y, we consider a sparse linear classifier as follows:

$$\min \|H\|_{2,1} \text{subject to} Y = ZH, \tag{7}$$

where H is the weight matrix. The reason of applying the sparse coding for H, again, is to enhance the generalization capability of the classifier, which has demonstrated the promising performance in many applications.

3.4. Optimization Process. Overall, by combining the stage of data reconstruction (from equation (6)) and classification (from equation (7)), the final loss function for the proposed SRA algorithm is formulated as follows:

$$\mathscr{L} = \min \|W\|_{2,1} + \|V\|_{2,1} + \|H\|_{2,1} \text{ subject to} \left\| f^{-1}(Z) - XW \right\|_2^2 + \left\| g^{-1}(X) - ZV \right\|_2^2 + \|Y - ZH\|_2^2 \le \mathbf{e}. \tag{8}$$

As such, the training process of the proposed SRA algorithm is formulated and represented using one unique objective function as equation (8). Among them, the first term of $\|f^{-1}(Z) - XW\|_2^2$ is used to identify the informative features, and the second term (*i.e.*, $\|g^{-1}(X) - ZV\|_2^2$) is for minimizing the data reconstruction error for autoencoders. The last term is to establish the classification process by the minimization of $\|Y - ZH\|_2^2$.

Note that there are in total four free variable matrices in the proposed objective function, including Z (the output for the encoder), W (the weight matrix within encoder), V (the weight matrix for the decoder), and H (the weight matrix for training the sparse classifier). However, the problem from equation (8) is nonconvex with respect to all variable matrices, which means we cannot solve them at the same time. A typical strategy is to adopt the iterative computation to address (or update) only one variable (*i.e.*, matrix) at one time, via fixing some others. As such, the optimization process of equation (8) is split into the following computational steps.

3.4.1. Update V and H with Fixed Z and W. By fixing Z and W, the proposed objective function can be expressed as follows:

$$\widehat{V}, \widehat{H} = \operatorname{argmin}\left(\|V\|_{2,1} + \|H\|_{2,1} \right) + \lambda_{VH} \left(\|H\|_{2,1} \left\| g^{-1}(X) - ZV \right\|_2^2 + \|Y - ZH\|_2^2 \right), \tag{9}$$

where λ_{VH} is the penalty term for balancing the solution sparsity (*i.e.*, $\|V\|_{2,1} + \|H\|_{2,1}$) and the training error. Alternatively, if we introduce two auxiliary matrices, that is, $R = \begin{pmatrix} g^{-1}(X) \\ Y \end{pmatrix}$ and $A = \begin{pmatrix} V \\ H \end{pmatrix}$, then equation (9) can be rewritten as follows:

$$\widehat{A} = \operatorname{argmin} \|A\|_{2,1} + \lambda_{VH} \|R - ZA\|_2^2. \tag{10}$$

The optimization problem from equation (10) then can be solved using the linearized alternating-direction method (LADM) [21] as follows:

$$A_{k+1} = \arg\min_A \frac{1}{2} \left\| \mathscr{A}_k - A \right\|_2^2 + \alpha \lambda_{VH} \|A\|_{2,1}, \tag{11}$$

where $\mathscr{A}_k = A_k - \alpha Z^T (ZA_k - R)$ and the parameter α satisfies $\alpha \in (0, (1/\|Z\|_2^2))$ for convergence. In addition, we also introduce the soft-thresholding-based (STB) operator STB_τ. That is, for a given matrix X, we have $STB_\tau(X^{(q)}) = X^{(q)}/\|X^{(q)}\|_2 \max(\|X^{(q)}\|_2 - \tau, 0)$, where τ is a given constant, and $X^{(q)}$ represents the q-th row of the matrix X. Accordingly, the estimation for A_{k+1} (at the $(k+1)$-th iteration) is given by the following:

$$A_{k+1}^{(q)} = STB_{(\tau = \alpha\lambda_{VH})} \left(\mathscr{A}_k^{(q)} \right), \forall q, \tag{12}$$

where, again, α and λ_{VH} is the controlling and penalty parameter and $\mathscr{A}_k^{(q)}$ represents the q-th row from $\mathscr{A}_k$.

3.4.2. Update Z with Fixed V and H. The original objective function is converted in the following:

$$\widehat{Z} = \operatorname{argmin}\left\|g^{-1}(X) - ZV\right\|_2^2 + \|Y - ZH\|_2^2. \tag{13}$$

Similarly, we can reuse equation (10) by introducing (or substituting) R and A. Now, with fixed A, the problem becomes to solve Z under the sparsity constraint of A (*i.e.*, $\|A\|_{2,1}$). This can be cast as a typical *dictionary-learning* process in sparse representation. A great number of research work have been done on this dictionary-learning topic, which aims to update the dictionary (Z in our context) via maintaining the sparsity constraint. In particular, the Online Dictionary-Learning (ODL) method [22] is adopted in our study. ODL considers an l_2-norm constraint for all columns from the dictionary, that is, $\mathbf{z}_j^T\mathbf{z} \neq 1$, where $\mathbf{z}_j$ is the j-th column of Z.

3.4.3. Update W with Fixed V, H, and Z. At last, the proposed objective function is expressed as follows when V, H, and Z are fixed:

$$\min \|W\|_{2,1} \ \textit{subject to} \ \left\|f^{-1}(Z) - XW\right\|_2^2 \leq \mathbf{e}. \tag{14}$$

The aforementioned function has a similar format of equation (10). As such, we apply the same STB operator to solve W. More precisely, at the $(k+1)$-th iteration, the estimation of W_{k+1} is given by the following:

$$W_{k+1}^{(q)} = STB_{(\tau=\beta\lambda_W)}\left(\mathscr{W}_k^{(q)}\right), \ \forall q, \tag{15}$$

where β satisfies $\beta \in (0, 1/\|X\|_2^2)$, λ_W is a user-defined penalty parameter, and $\mathscr{W}_k^{(q)}$ represents the q-th row of the matrix $\mathscr{W}_k$, which can be calculated by $\mathscr{W}_k = W_k - \alpha X^T(XW_k - f^{-1}(Z))$.

Notably, the above optimization method follows a typical alternating-direction strategy, from which its convergence analysis can be guaranteed and found in Ref. [23]. That is, the convergence of the proposed algorithm can be stated as follows.

Theorem 1. *Assume at the $(k-1)$-th iteration, V_{k-1}, H_{k-1}, and W_{k-1} are the indeterminate solution. Using the proposed algorithm, we have the loss value $\mathscr{L}(V, H, W)$ from equation (8) which decreases and a local minimal can be obtained.*

3.5. Summary. From the aforementioned process, a novel sparse-coding-based autoencoder (termed as SRA) is proposed. This main contribution of our work can be summarized as follows:

(i) We introduce the sparse coding technique to optimize the encoder and decoder structure, with the aim of improving the generalization capability and identifying informative features.

(ii) The latent information is extracted and cast as an alternative of the original input, which is further leveraged for the subsequent classification.

(iii) The proposed algorithm is capable of reconstructing input data, selecting important features, and training the classifier simultaneously.

Finally, the proposed algorithm is summarized in Algorithm 1.

To halt the proposed algorithm, the termination criterion is set either the maximal iteration (K) is reached or the value of $\|\mathscr{L}_k - \mathscr{L}_{k-1}\|_2^2/\|\mathscr{L}_k\|_2^2$ is less than a threshold ϵ (from equation (8)), where ϵ is a user-defined value. As such, the proposed algorithm iteratively updates W, V, H, and D until the convergence condition is met.

4. Experimental Results

In this section, we present experimental results following the application of the proposed autoencoder algorithm for cancer patients' survivability prediction. As part of this process, we first discuss hyper-parameter sensitivity (such as the scale of latent information) and its impact on the performance of the proposed algorithm. Additionally, identified key features from the proposed autoencoder are also presented and investigated. We also compare the proposed algorithm with other existing methods.

4.1. Experimental Setup. To investigate the problem of survivability prediction, we employed an open-sourced public dataset from Surveillance, Epidemiology, and End Results (*i.e.*, SEER) website (https://seer.cancer.gov/). This initiative is to ensure high-quality medical data and comprehensive information display on cancer, in order to facilitate various institutions and laboratories to perform their own research. The cancer-related data have been accumulated since 1973. On the one hand, there are more than 6 types of cancer, including breast and lung, *etc.* On the other hand, this dataset covers a wide range of patient profile, such as demographics, primary tumor site, tumor morphology, stage at diagnosis, and first course of treatment, to name a few. Given different versions of SEER (due to the data updating), in our study, we employ the 2017 version to consider the relevant data collected between 1973 and 2015. We further select the breast cancer as our main focus, which is with 828,457 records and 147 variables.

A sophisticate preprocess is firstly applied to the raw data, with the aim of removing unnecessary data. For instance, to ensure the reliability of the outcome, the following exclusion criteria are applied: (i) records with the unknown year of birth, (ii) records with death due to other than cancer, and (iii) records with missing survival time. We further remove meaningless features, such as patient ID and features with only one value. For the corrupted SEER data (with missing values), we then apply the most-frequent values for the imputation. At last, we select 130 features for analytical purposes; among them, the feature of "srv_time_mon" is taken as the output label, which indicates the survivable time span for a patient. More precisely, we

categorize the problem of survivability prediction as a binary classification. That is, the cutting-off value for "srv_time_mon" is set as 60. As such, patients survived less than 60 months after diagnosis will be assigned into one class and with another class for more than 60 months. After the aforementioned data preprocessing, there are in total 540,138 records remained in our study (*i.e.*, 318,710 records with "srv_time_mon" $\geqslant 60$ and 221,428 records with < 60).

To evaluate the performance, we firstly introduce four classification measurements: TP (true positive), TN (true negative), FP (false positive), and FN (false negative). Table 1 further explains their detailed calculation. Based on this confusion matrix, we eventually form sensitivity, precision, accuracy, and $F1$ scores to measure the performance of our proposed model, as follows:

$$\begin{aligned} \text{sensitivity} &= \frac{\text{TP}}{(\text{TP} + \text{FN})}, \\ \text{precision} &= \frac{\text{TP}}{(\text{TP} + \text{FP})}, \\ \text{accuracy} &= \frac{(\text{TP} + \text{TN})}{(\text{TP} + \text{FN} + \text{TN} + \text{FP})}, \\ F1 &= 2 \times \frac{(\text{precision} \times \text{sensitivity})}{(\text{Precision} + \text{Sensitivity})}. \end{aligned} \tag{16}$$

At last, the entire dataset is randomly partitioned into two independent sets: a training and testing set. The size of the training and testing set is set as 80%, and 20%, respectively. Other training parameters are shown in Table 2.

4.2. Performance Validation. In this section, we analyze the robustness of the proposed algorithm in terms of the scale of latent information. Again, the raw input data are reconstructed by the encoder and the latent information is extract for the final classification. The latent information is represented by the output from the encoder (or the hidden layer in the autoencoder), or $Z \in \mathbb{R}^{Q \times N}$, where Q is the input dimension and N is the number of hidden nodes. As the key parameter, a bigger value of N might lead to an intensive computation, while smaller N might be insufficient to capture enough information for the subsequent training. Towards this end, we conduct experiments in this section to evaluate the impact of the scale of latent information (*i.e.*, N) on the proposed SRA algorithm. In particular, we consider the value of N based on the proportion of input dimension (Q), in which the range setting of N is set as $N \in [20\%, 40\%, 60\%, 80\%]$ (against of Q). On the other hand, for the subsequent classifier, we implement one hidden layer with the same number of hidden neurons as N. The experiments are repeated 30 times, and the comparison result is shown in Table 3.

From the average results, we notice that with a higher dimension (larger N), a better training performance is expected. The reason could be a smaller number of hidden nodes (*i.e.*, N) results in the information loss, thereby making it difficult for the classifier. By contrast, the accuracy on the testing sets seems to remain relatively stable regardless of the change of N. However, more computational time is required with the increase of N, due to the larger size of the autoencoder. As such, in the following experiments, we adopt $N = 40\%$ due to its accurate performance on the testing set and affordable computational time.

4.3. Feature Justification. In the following section, identified features from SRA are taken into account by comparing with manually selected ones. Again, the proposed SRA method applies the sparse coding technique to optimize the network structure (assigning the zero value to some weights). By doing so, features (with nonzero weights) are considered as most-informative ones and selected. Therefore, we will analyze those selected features from SRA and compare with existing work. To make a fair comparison, we take the number of selected features as 15 in our study.

Firstly, we manually identify 15 common features used in existing studies for breast cancer survival prediction and show them in Table 4. Then, we employed a standard Multiple-Layer Perceptron (MLP) to train on those features, for which we label as the baseline method. Secondly, we also run SRA to automatically select the Top-15 features. More precisely, we select features associated with largest magnitudes from W and compare their classification performance with the baseline method.

Table 5 shows the feature list identified by the proposed SRA method. Compared with those manually selected features in Table 4, there are six common features that have been identified by both of them, namely, Registry ID, Age at Diagnosis, Primary site, Tumor Size, Surgery Type, and Histology. Among them, the Registry ID represents the geography information for one particular patient. While patients' geography detail could carry significant information, such as external environment and individual background, this risk factor contributes to the breast cancer survival rate. As a result, both methods indicate that the feature of Registry ID plays a key role in predicting the survival rate, which is consistent with many existing research that shows a strong correlation between the disease and patients' profile. Similarly, other features, such as Age at Diagnosis, Primary site, and Histology, also indicate the identification capability of the proposed SRA method to select informative features. Nevertheless, our method also explore more disease-related features, including SITERWHO, ICDOTO9V, and CSSCHEMA, etc., which provide more details about the development of the cancer. For instance, the AYASITERWHO feature is used to record the Primary site and Histology on adolescent. Therefore, compared to existing research, the proposed algorithm is able to identify more disease-related features, instead of selecting general ones. To verify those features, we further compare the baseline and our proposed method by evaluating their classification performance.

Figure 3 presents the average test accuracy from these two methods. Compared to the baseline method, the proposed algorithm achieves a significant improvement in terms of the higher classification accuracy. For instance, the

Input: raw training examples X and Y, the number of maximal iterations K, the stopping threshold ϵ, and regularization parameters of $(\lambda_{VH}, \lambda_W)$;
Initialization:
randomly assign values to W, V and H;
calculate the output from the encoder $Z = f(XW)$, where $f(\cdot)$ is the activate function;
for $k = 1$ ***to*** K **do**
(1) **Estimate** V_k **and** H_k **with fixed** Z_{k-1} **and** W_{k-1} **(using equation** (12)**);**
(2) **Estimate** Z_k **with fixed** V_k **and** H_k **using the ODL method;**
(3) **Estimate** W_k **with fixed** V_k**,** H_k **and** Z_k **(using equation** (15)**);**
if the predefined termination condition (the threshold ϵ) **then**
Stop training;
end
end
Output: Return the optimal solution W_k, V_k, H_k, and D_k.

Algorithm 1: Proposed sparse representation-based Autoencoder algorithm.

Table 1: Employed confusion matrix in our study.

Confusion matrix		Actual	
		True	False
Prediction	True	TP	FP
	False	FN	TN

Table 2: Parameters for the proposed SRA algorithm.

(i) Activation function is Relu;
(ii) Maximum number of training iterations is 200;
(iii) Stopping threshold ϵ is 0.001;
(iv) Regularization parameters λ_{VH} and λ_W are 0.3.

Table 3: Average $F1$ scores as a function of the changing number of hidden nodes (*i.e.*, N).

N	20%	40%	60%	80%
Train	78.1%	81.2%	82.5%	83.1%
Test	75.8%	80.0%	79.9%	79.7%
Time (seconds)	85.3	89.2	93.5	100.1

Table 4: Manually identified features from existing research.

Feature list	[13]	[8]	[7]	[6]	[14]	[12]
Age at diagnosis		√	√	√	√	√
Race	√	√			√	
Marital status			√	√	√	√
Histology			√	√	√	√
Primary site	√		√			√
Laterality	√	√		√		
Surgery type	√	√	√	√		
Lymph node	√	√	√	√	√	√
Tumor size	√	√	√	√	√	√
Grade	√		√	√	√	
Radiation	√	√			√	√
Registry ID		√			√	

proposed method achieves 80% $F1$ score, compared to 72.3% from the baseline approach. The experimental results clearly show that the proposed method has identified more informative features, than those of manually selected ones. Again, from the feature list in Table 5, our method seems to be able to pick up disease-related features, instead of selecting general ones, thereby resulting in an improvement.

4.4. Comparison with State-of-the-Art Methods. In this section, the performance of the proposed algorithm is compared with other state-of-art methods. Four autoencoder-based algorithms are introduced here, including standard autoencoder (SAE) [20], SSAE [16], MG-RNN-AE [18], and DAE [19], respectively (we have introduced them in Section 2.2). Additionally, we also employ other approaches, such as Random Forest (RF [8]), Support Vector Machine (SVM [11]), and Artificial Neural Network (ANN [13]).

Note that except SSAE, algorithms of SAE, MG-RNN-AE, and DAE are designed for data reconstruction, not for the classification purpose. As such, to make a fair comparison with the proposed algorithm, we further manipulate them as the input for training another ANN. For all

Table 5: Feature list identified by the proposed SRA method.

Feature list	Manual	Feature list	Manual	Feature list	Manual
Registry ID	√	Age at diagnosis	√	YR_BRTH	
Primary site	√	Histologic type		Tumor size	√
Site-specific for tumor		Surgery type	√	Histology	√
SITERWHO		ICDOTO9V		ICDOT10V	
AYASITERWHO		LYMSUBRWHO		CSSCHEMA	

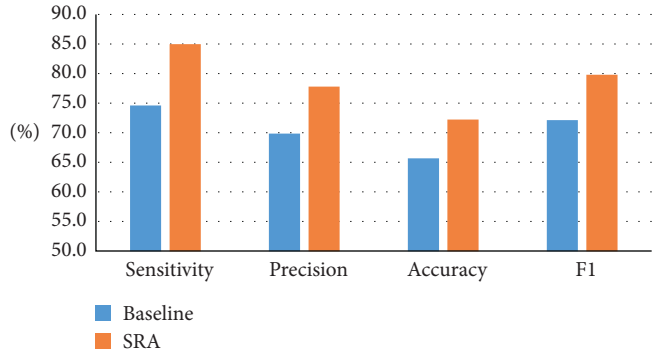

Figure 3: Comparison of classification performance (on the testing set) based on selected and SRA-identified features.

Table 6: Parameters for various training algorithms.

Algorithms	Training parameters
SAE	(i) A fully connected three-layer structure with 64 hidden neurons
SSAE	(i) A fully connected three-layer structure with 300 hidden neurons (ii) Aparsity parameter: 0.001
MG-RNN-AE	(i) A fully- connected three-layer structure with 10 hidden neurons (ii) Regularization parameters: 0.9 and 0.1
DAE	(i) A fully connected three-layer structure with 300 hidden neurons (ii) Regularization parameters: 0.6
RF	(i) The maximum depth per tree: 5 (ii) The number of trees: 7 (iii) The percentage of features used per tree: 15%
SVM	(i) Regularization parameter: 1 (ii) Kernel function: Radial Basis Function (RBF) (iii) Kernel coefficient: 0.01 (iv) Tolerance for stopping criterion: $1e-3$ (v) Maximum number of iterations: 500
ANN	(i) A fully connected three-layer structure with 64 hidden neurons (ii) The RPROP training algorithm with the maximum number of iterations 500 (iii) The activation function is Sigmoid

autoencoder-based methods, we set the training optimizer as Adam, learning rate as 0.001, and maximum number of iterations as 200. Other training parameters used are given accordingly in Table 6.

Figure 4 presents the average training and test accuracy obtained from different methods, respectively, while the accuracy standard deviation is also provided. As observed, the proposed SRA algorithm achieves a significant better result in terms of classification accuracy, in comparison to conventional training algorithms. For instance, for those autoencoder-based methods, on average, they achieve the result of 72.1% and 68.0% on the training and test sets, respectively, which is worse than that of classification accuracy from SRA (81.2% and 80.0% on training and testing). Although the MG-RNN-AE method has performed similar training results like ours, again the proposed algorithm is superior than that of MG-RNN-AE on the testing set. The similar observation is made from the nonautoencoder-based methods too. More precisely, our proposed method scores the best training and generalization outcome for prediction, compared to RF, SVM, and ANN. In addition, we also notice that the RF method has a notable variance in the testing set,

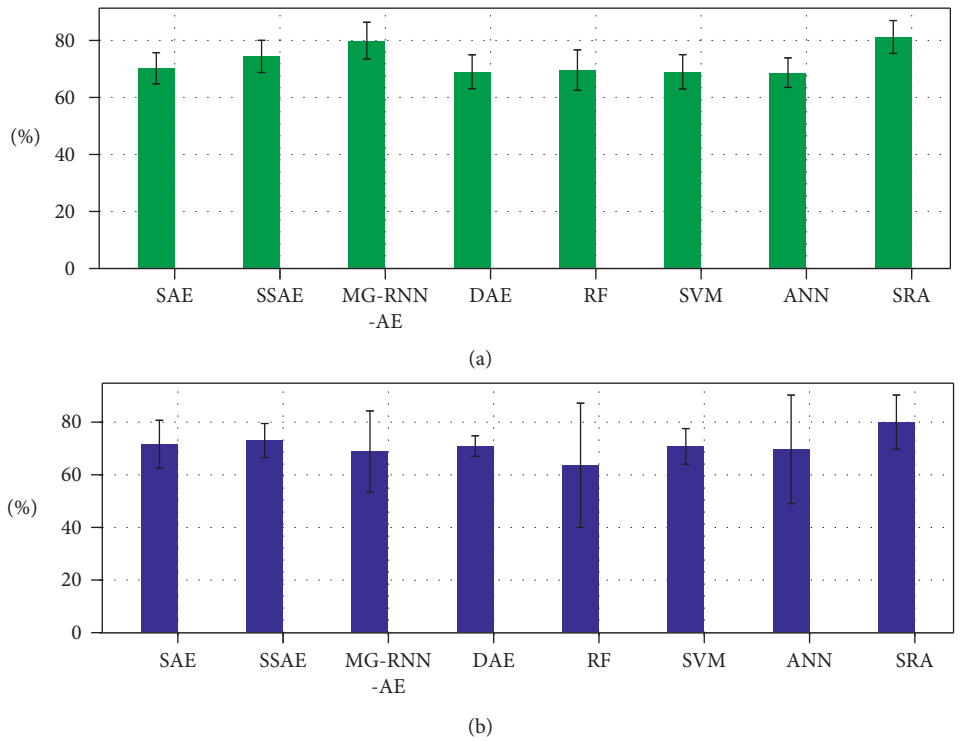

Figure 4: Average training and test accuracy obtained from different algorithms for the classification tasks.

which might be caused by the way of combining individual trees to form the final outcome. By contrast, the proposed method has a relatively stable performance from either the training and testing sets. Overall, it is empirically confirmed that the proposed SRA method leads to a significant improvement in comparison to other existing methods, in terms of the prediction accuracy.

In conclusion, it can be empirically confirmed that the proposed algorithm outperforms existing state-of-the-art approaches. The main reason is twofold: (i) SRA introduces the sparse coding technique to optimize the network structure. By contrast, other autoencoders (such as SAE, MG-RNN-AE, and DAE) rely on the cross-validation or trail-and-error to decide their structure, which results in a poor performance; (ii) SRA utilizes the latent information to train the final classifier, while traditional methods (including RF, SVM, and ANN) are directly employed to classify on the raw inputs. However, we argue that the original data may consist of noise and outliers that have a negative impact on training the classifier. As a result, the proposed SRA algorithm achieves an satisfactory classification accuracy, by integrating the sparse coding for the structure optimization and training the classifier using the latent information simultaneously.

5. Conclusion

In this study, we have proposed a novel sparse-coding based autoencoder (termed as SRA) algorithm for addressing the problem of cancer survivability prediction. In the real-world applications, the medical data are subject to some noise (such as missing values and outliers). As such, traditional methods find it difficult to predict the survivability span. By contrast, the proposed SRA method has contributed to the following improvement:

(i) To apply the sparse coding technique to optimize the network structure and more importantly, informative features are identified accordingly by minimizing the number of network weights

(ii) To employ the latent information (i.e., the output from the encoder) is manipulated as the training inputs, rather than the original data

(iii) To formulate the training process using one unique objective function, which is further solved by an iterative computational strategy

Experiments are conducted and the performance is evaluated using one of the popular health datasets from Surveillance, Epidemiology, and End Results (*i.e.*, SEER). The prediction results clearly indicate a more accurate outcome from the proposed method, compared to existing methods. In the future, we plan to apply the proposed method on other datasets from different domains. We also consider to improve the proposed method using other optimization methods or data-driven network structures.

Data Availability

Data are available from https://seer.cancer.gov/.

Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this study.

Acknowledgments

This work was supported by the Natural Science Foundation of Zhejiang Province (LGN20C050002, LGN21F010001, LZY22E0500003, and LZY22E0500005).

References

[1] J. Yang and J. Ma, "A structure optimization framework for feed-forward neural networks using sparse representation," *Knowledge-Based Systems*, vol. 109, pp. 61–70, 2016.

[2] J. Yang and J. Ma, "Feed-forward neural network training using sparse representation," *Expert Systems with Applications*, vol. 116, pp. 255–264, 2019.

[3] J. Yang, J. Ma, K. T. Win, J. Gao, and Z. Yang, "Low-rank and sparse representation based learning for cancer survivability prediction," *Information Sciences*, vol. 582, pp. 573–592, 2022.

[4] M. Amarlingam, P. K. Mishra, P. Rajalakshmi, S. S. Channappayya, and C. S. Sastry, "Novel light weight compressed data aggregation using sparse measurements for iot networks," *Journal of Network and Computer Applications*, vol. 121, pp. 119–134, 2018.

[5] N. Shahid, T. Rappon, and W. Berta, "Applications of artificial neural networks in health care organizational decision-making: a scoping review," *Plos One*, vol. 14, no. 2, pp. e0212356–22, 2019.

[6] C. Ponnuraja, B. C. Lakshmanan, V. Srinivasan, and K. Prasanth, "Decision tree classification and model evaluation for breast cancer survivability: a data mining approach," *Biomed Pharmacol Journal*, vol. 10, no. 1, pp. 281–289, 2017.

[7] S. Momenyan, A. R. Baghestani, N. Momenyan, P. Naseri, and M. E. Akbari, "Survival prediction of patients with breast cancer: comparisons of decision tree and logistic regression analysis," *Internationl Journal of cancer management*, vol. 11, no. 7, Article ID e9176, 2018.

[8] M. D. Ganggayah, N. A. Taib, Y. C. Har, P. Lio, and S. K. Dhillon, "Predicting factors for survival of breast cancer patients using machine learning techniques," *BMC Medical Informatics and Decision Making*, vol. 19, no. 1, p. 48, 2019.

[9] S. M. Basha, D. S. Rajput, N. C. S. N. Iyengar, and R. D. Caytiles, "A novel approach to perform analysis and prediction on breast cancer dataset using R," *International Journal of Grid and Distributed Computing*, vol. 11, no. 2, pp. 41–54, 2018.

[10] C.-M. Chao, Y.-W. Yu, B.-W. Cheng, and Y.-L. Kuo, "Construction the model on the breast cancer survival analysis use support vector machine, logistic regression and decision tree," *Journal of Medical Systems*, vol. 38, no. 10, p. 106, 2014.

[11] S. Jhajharia, S. Verma, and R. Kumar, "Predictive analytics for breast cancer survivability: a comparison of five predictive models," in *Proceedings of the Second International Conference on Information and Communication Technology for Competitive Strategies*, pp. 1–5, Association for Computing Machinery, Udaipur India, 2016.

[12] T. Wang, C. Cheng, and H. Chiu, "Predicting post-treatment survivability of patients with breast cancer using artificial neural network methods," in *Proceedings of the Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*, pp. 1290–1293, Osaka, July 2013.

[13] B. R. A. Cirkovic, A. M. Cvetkovic, S. M. Ninkovic, and N. D. Filipovic, "Prediction models for estimation of survival rate and relapse for breast cancer patients," in *Proceedings of the 2015 IEEE 15th International Conference on Bioinformatics and Bioengineering (BIBE)*, pp. 1–6, Belgrade, Serbia, November 2015.

[14] K. Park, A. Ali, D. Kim, Y. An, M. Kim, and H. Shin, "Robust predictive model for evaluating breast cancer survivability," *Engineering Applications of Artificial Intelligence*, vol. 26, no. 9, pp. 2194–2205, 2013.

[15] C. M. Bishop, *Neural Networks for Pattern Recognition*, Oxford University Press, Inc., New York, NY, USA, 1995.

[16] Y. Xiao, J. Wu, Z. Lin, and X. Zhao, "A semi-supervised deep learning method based on stacked sparse auto-encoder for cancer prediction using rna-seq data," *Computer Methods and Programs in Biomedicine*, vol. 166, pp. 99–105, 2018.

[17] Y. Liu, M. Zhai, J. Jin et al., "Intelligent online catastrophe assessment and preventive control via a stacked denoising autoencoder," *Neurocomputing*, vol. 380, pp. 306–320, 2020.

[18] J. Loy-Benitez, S. Heo, and C. Yoo, "Soft sensor validation for monitoring and resilient control of sequential subway indoor air quality through memory-gated recurrent neural networks-based autoencoders," *Control Engineering Practice*, vol. 97, Article ID 104330, 2020.

[19] X. Luo, X. Li, Z. Wang, and J. Liang, "Discriminant autoencoder for feature extraction in fault diagnosis," *Chemometrics and Intelligent Laboratory Systems*, vol. 192, Article ID 103814, 2019.

[20] C. Fan, F. Xiao, Y. Zhao, and J. Wang, "Analytical investigation of autoencoder-based methods for unsupervised anomaly detection in building energy data," *Applied Energy*, vol. 211, pp. 1123–1135, 2018.

[21] Z. Lin, R. Liu, and Z. Su, "Linearized alternating direction method with adaptive penalty for low-rank representation," in *Advances in Neural Information Processing Systems*, J. Shawe-Taylor, R. S. Zemel, P. L. Bartlett, F. Pereira, and K. Q. Weinberger, Eds., vol. 24, , pp. 612–620, Curran Associates, Inc, 2011.

[22] J. Mairal, F. Bach, J. Ponce, and G. Sapiro, "Online dictionary learning for sparse coding," in *Proceedings of the 26th Annual International Conference on Machine Learning, ICML '09*, pp. 689–696, Association for Computing Machinery, Montreal, Canada, June 2009.

[23] Z. Lin, R. Liu, and H. Li, "Linearized alternating direction method with parallel splitting and adaptive penalty for separable convex programs in Machine Learning," *Machine Learning*, vol. 99, no. 2, pp. 287–325, 2015.