IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received April 7, 2020, accepted April 14, 2020, date of publication April 24, 2020, date of current version May 11, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.2990375

Feature Extraction Methods in Quantitative Structure–Activity Relationship Modeling: A Comparative Study

SHROOQ A. ALSENAN[1], (Member, IEEE), ISRA M. AL-TURAIKI[2], AND ALAAELDIN M. HAFEZ[3], (Member, IEEE)

[1]Research Center, College of Computer and Information Sciences, Princess Nourah Bint Abdulrahman University, Riyadh 11671, Saudi Arabia
[2]Information Technology Department, College of Computer and Information Sciences, King Saud University, Riyadh 11451, Saudi Arabia
[3]Information Systems Department, College of Computer and Information Sciences, King Saud University, Riyadh 11451, Saudi Arabia

Corresponding author: Shrooq A. Alsenan (saalsnan@pnu.edu.sa)

This work was supported by the Deanship of Scientific Research at Princess Nourah Bint Abdulrahman University through the Fast-Track Research Funding Program.

ABSTRACT Computational approaches for synthesizing new chemical compounds have resulted in a major explosion of chemical data in the field of drug discovery. The quantitative structure–activity relationship (QSAR) is a widely used classification and regression method used to represent the relationship between a chemical structure and its activities. This research focuses on the effect of dimensionality-reduction techniques on a high-dimensional QSAR dataset. Because of the multi-dimensional nature of QSAR, dimensionality-reduction techniques have become an integral part of its modeling process. Principal component analysis (PCA) is a feature extraction technique with several applications in exploratory data analysis, visualization and dimensionality reduction. However, linear PCA is inadequate to handle the complex structure of QSAR data. In light of the wide array of current feature-extraction techniques, we perform a comparative empirical study to investigate five feature-extraction techniques: PCA, kernel PCA, deep generalized autoencoder (dGAE), Gaussian random projection (GRP), and sparse random projection (SRP). The experiments are performed on a high-dimensional QSAR dataset, which comprises 6394 features. The transformed low-dimensional dataset is inputted into a deep learning classification model to predict a QSAR biological activity. Three approaches are adopted to validate and measure the proposed techniques: (i) comparing the performance of the classification models, (ii) visualizing the relationship (correlation) between features in the low-dimension Euclidean space, and (iii) validating the proposed techniques using an external dataset. To the best of our knowledge, this study is the first to investigate and compare the aforementioned feature-extraction techniques in QSAR modeling context. The results obtained provide invaluable insights regarding the behavior of different techniques with both negative and positive classes. With linear PCA as a baseline, we prove that the investigated techniques substantially outperform the baseline in multiple accuracy measures and demonstrate useful ways of extracting significant features.

INDEX TERMS Autoencoder, blood-brain barrier (BBB) permeability, deep generalized autoencoder (dGAE), dimensioanlity reduction, feature extraction, Gaussian random projection, principal component analysis, quantitative structure–activity relation (QSAR), sparse random projection.

I. INTRODUCTION

The rapid development of technology had led to explosive growth in data in many fields. Drug discovery has benefited from the computational approaches for synthesizing new compounds. Chemical synthesis and virtual screening enabled the fast-paced generation of biological and chemical data and automated modeling [1]. This resulted in a need for practical methods to model the relationship between molecular structures and properties [2]. *Quantitative structure–activity relation* (QSAR) modeling is a widely used classification and regression method that represents the

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

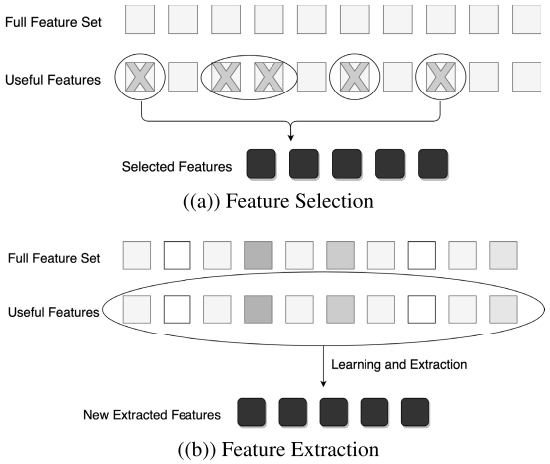


FIGURE 1. Feature extraction and selection.

relationship between a chemical structure and its activities. Features in QSAR dataset are called *molecular descriptors* or simply *descriptors*. QSAR models are characterized by being high-dimensional. However, high dimensionality of datasets is not desirable because it causes noise, redundancy, and increases computational complexity [3]. A large number of irrelevant features that do not contribute to the classification task may cause model overfitting [4]. With overfitting, the classifier learns exceptions specific to the training data and fails to generalize to newly encountered data [5]. According to Ladha [6], reducing the high-dimensional space of molecular descriptors provides many advantages, including limiting storage requirements, accelerating the algorithm's learning speed, improving the data quality, increasing the model accuracy and hence improving performance, saving resources for subsequent data collection, and providing further knowledge since the data are simple to understand and visualize.

Manual feature selection or selection based on prior knowledge has a clear potential for bias modelling, especially if certain features are known to be more effective than others in a classification problem [7], [8]. Many previous studies relied on the prior knowledge of experts for selecting descriptors [9]–[11] or applying dimensionality-reduction techniques to a formerly selected set of descriptors [12], [13].

There are two broad approaches for reducing the number of features in datasets: *feature extraction* and *feature selection* [3], [14]. Feature selection is a process in which a subset of the feature's space is chosen according to its relevance to the output of the classifier. The ultimate goal is to obtain the most effective subset of existing features without creating new features [3]. Feature-extraction techniques produce new features by combining existing ones and then discarding the original features. A simple overview of feature selection and extraction methods is shown in Figure 1. Feature-extraction techniques make it possible to handle the complexity of high-dimensional data, extract invaluable feature patterns, and improve classification accuracy. Feature-selection methods are effective in removing redundant features, while extraction methods are more capable of handling the complexity of high-dimensional data [15], [16]. The most widely used feature-extraction method in QSAR is *Principal Component Analysis* (PCA) [17], used due to its simplicity and low computational cost. PCA analyzes inter-correlated dependent variables to obtain a new set of variables called (principal components). However, studies have shown that other feature-extraction methods can handle the complexity of high-dimensional data more efficiently than linear PCA. According to Sharma [18], non-linear feature-extraction methods, such as *autoencoder* [8], [19] or *Kernel PCA* [20], [21], are found to be more effective in extracting complex structures and hidden patterns. Other studies argued that linear feature-extraction methods such as *random projection (RP)* have achieved satisfactory results with less computational overhead when compared to PCA [22], [23].

To address the formally presented perspectives, we addressed the following fundamental question: *How can the high dimensionality of QSAR datasets be reduced to a low-dimensional space with a minimum loss of valuable information?* To answer this question, we investigate a number of feature-extraction techniques that have proven to be successful in the context of dimensionality reduction. In particular, we experiment with KPCA [20], deep generalized autoencoder (dGAE) [24], Gaussian random projection (GRP) [22], and sparse random projection (SRP) [25]. Previous efforts to review feature-extraction methods by Storcheus *et al.* [14] and Idakwo *et al.* [26] indicated that there is an urgent need to demonstrate the performance capabilities of different feature-extraction techniques in handling a high-dimensional QSAR dataset. The choice of the aforementioned feature-extraction methods relied on three main reasons: (i) multiple studies reported successful application of each of these techniques in other domains, [16], [21], [22], [24], [25], (ii) the proposed techniques represent branches of both linear and non-linear feature-extraction techniques, and (iii) there is a lack of work comparing these five feature-extraction methods collectively in a QSAR problem.

To the best of our knowledge, this is the first empirical comparative analysis of feature-extraction methods on a high-dimensional QSAR dataset, as compared previous endeavors have centered around feature-selection techniques in QSAR [3], [17], [27]. One essential requirement in a binary QSAR classification problem is the capability of separating class labels efficiently. Although dimensionality-reduction techniques facilitate this task, linear PCA may fail to extract non-linear relationships or extract complex hidden structures when modeling QSAR problems. This study contributes to QSAR modeling by investigating other linear and non-linear feature-extraction methods that have proven to outperform both selection methods and linear PCA in other contexts. We want to investigate the potential of feature-extraction methods in separating class labels in QSAR datasets. Specifically, we seek to understand whether the investigated feature-extraction methods are capable of improving

specificity scores (predicting the negative class) and separating class labels in a binary classification problem.

The selected high-dimensional QSAR dataset is for blood–brain barrier (BBB) permeability [13]. The BBB permeability dataset was chosen for this study for two primary reasons: it is a benchmark dataset consisting of 2350 compounds, which were used to compare our results, and there are no research studies that have applied feature-extraction methods (besides linear PCA) on this dataset. With the availability of online tools to calculate descriptors, generating a high-dimensional dataset with over 6390 descriptors is feasible, which could aid in accomplishing the research objectives.

According to Idakwo *et al.* [26], one way to assess the performance of feature-extraction methods is based on the classification model's accuracy. This study follows this approach in comparing the five feature-extraction methods by inputting the dataset with the reduced dimensions into a deep learning (DL) classification model. The performance of the classification model is assessed using multiple accuracy measures to closely investigate the classifier's capability in predicting both class labels.

In addition to validating the feature-extraction methods using the classifier's accuracy, this paper provides another view of the proposed techniques through scatter-plot visualization of the distribution of data points. This step serves two purposes: (i) to detect if the transformed feature space demonstrates positive correlation between features, and (ii) to showcase the proposed the ability of the proposed technique to separate class labels prior to inputting the data into the classification model [28]. When data move in the same positive direction corresponding to two or more features, it indicates a statistical correlation between these features. The occurrence of correlated features in the low-transformed feature space is not a desirable quality, as it indicates the inability of the technique to retain invaluable information [29].

The accuracy measures used in this study consider the imbalanced nature of the dataset. We assess the classifier performance based on the area under the curve (AUC), Matthew Correlation Coefficient (MCC), accuracy, sensitivity, and specificity. The AUC showcases the ability of a feature-extraction technique to separate instances of different classes. Achieving high accuracy in both sensitivity and specificity has been a challenging task in previous QSAR studies [13], [30].

This paper is organized as follows. First, in Section (II) we review previous attempts using QSAR datasets for dimensionality reduction; then, we provide an overview of both feature-selection and extraction methods. Although a brief review of feature-selection methods is presented, the primary focus is on feature-extraction methods in QSAR modeling. Next, in Section IV we present our research methodology and three ways to assess the proposed techniques: a classification model, a visualization analysis of the relationships between features in a QSAR dataset, and finally a visualization analysis on an external dataset.

II. LITERATURE REVIEW

QSAR models encode features related to chemical compounds using molecular descriptors (MDs). QSAR datasets are characterized by their high dimensionality. Thousands of features are generated to model QSAR classification problems [26]. Early application of QSAR models were reliant on a small number of linearly correlated MDs. However, current QSAR models are non-linear and include thousands of chemical compounds and their respective (MDs) [31]. Feature selection and extraction are challenging tasks in recent non-linear QSAR models. High dimensionality affects the performance of the classifier because of data noise, redundant data, and high complexity [32], [33].

Studies on MD types, their usefulness, and their selection and extraction methods are plentiful [3], [15], [34], [35]. Danishuddin *et al.* presented a review of a number of feature-selection methods for QSAR modeling, including filters, wrappers, and embedded/hybrid approaches [3]. They concluded that feature-selection methods vary in their importance depending on the task under consideration. However, feature-selection methods are commonly preferred in QSAR for removing redundant data while still applying feature-extraction techniques to efficiently handle the complexity and high dimensionality of QSAR data [15]. Hechinger *et al.* [36] provided insights on the heterogeneity of which various descriptors are obtained. They concluded that the use of molecule conformations for 3D descriptors and the many computational programs for generating descriptors may lead to inaccurate information about the chemical structure. Feature-selection methods have been used extensively in BBB permeability problems. Li *et al.* [12] performed one of the early works on dimensionality reduction on a BBB dataset using a feature-selection method called recursive feature elimination (RFE) to extract features. They reported that features selected by RFE contributed to the best-performing classification model.

At the most basic level, dimensionality-reduction techniques can be linear or non-linear and supervised or unsupervised [14]. Widely used linear feature-extraction methods include PCA [37], Linear Discriminant Analysis (LDA) [38], and random projection (RP) [39].

PCA is a widely used dimensionality-reduction technique in QSAR. Yoo and Shahlaei [40] tested PCA on a dataset of chemokine receptor 5 inhibitors. Their results showed that three principle components were able to describe the variance with a minimum loss of original information. In addition, they showed that PCA can contain important information held in multiple descriptors in only a few principle components. LDA is a supervised machine learning technique that employs a linear transformation of the features to reach the optimal class discrimination [38] for the purposes of dimensionality-reduction or classification. As with linear PCA, LDA closely extracts linear combinations of features that best describe the data. LDA is used for classification and for feature-extraction purposes. LDA has had many recent

applications, such as extracting EEG and EMG signals [41], emotion recognition [42], and face recognition [43]. According to Idakwo *et al.* [26], it is common for LDA to be ineffective when dealing with complex data structures. Studies with binary classification problems have reported that PCA works better than LDA in separating two-class labels. Therefore, LDA was not considered for this study [44].

RP is a linear method based on metric learning that calculates the distance between data points [14]. It is a useful and computationally inexpensive linear dimensionality-reduction method based on the Johnson- Lindenstrauss lemma theorem [39]. Dimitris Achlioptas presented a less computationally complex version of RP [23]. RP is yet to be explored in QSAR studies, although it has been proven successful in classification problems when compared to PCA [25]. Li *et al.* [45] presented a comparative analysis of feature-extraction techniques in optical projection tomography. They reported that RP excelled in efficiency and simplicity.

Kernel PCA is a non-linear feature-extraction method. As with linear PCA, KPCA is an embedding method that maps input vectors into another space. The main difference between the two methods is that the former transforms data in Euclidean space, while KPCA projects data in kernel space [46]. Kernel PCA is a type of ''manifold leanring'' in which data points are projected to a lower-dimensional space. A number of algorithms exist, such as Locally Linear Embedding (LLE) [47], Isomap [48], and Hessian LLE (HLLE) [49]. According to Chen and Liu *et al.* [50], many manifold algorithms, such as LLE and HLLE, are sensitive to noise and do not perform well on unseen data. Other studies confirmed this view, such as [51], [52], where linear PCA was more robust than LLE and Isomap in a noisy dataset. In addition, these algorithms require a higher computational overhead and excessive parameter adjustment. In QSAR models, L'Heureux *et al.* [53] confirmed that LLE handles non-linearity in smaller datasets compared to autoencoder. In addition, LLE has a limitation of learning one-way mapping, as compared to autonecoder, which is capable of learning two-way mapping between the high- and low-dimension space [54]. LLE has a major limitation of not being able to gradually extract features; hence, features are extracted at once, and the relationship between samples is not properly preserved [16].

Other types of non-linear feature-extraction methods include graph-based methods [55] and autoencoder [56]. Graph-based methods have demonstrated great success in extracting feature in image datasets. Autoencoder is a feature-extraction method based on representation learning. Some of the early attempts to use Artificial Neural Networks (ANN) architecture to extract features were conducted by Dorronsoro *et al.* [57] and Guerra *et al.* [58]. They developed an unsupervised ANN model to extract descriptors for a QSAR classification problem. Since then, researchers have determined that the autoencoder architecture is useful for dimensionality-reduction purposes [16]. Unlike PCA, autonedoer can handle non-linear data efficiently. Hinton and Salakhutdinov [59], a leading group of scientists in neural network research, presented one of the early publications on the application of autoencoder for dimensionality-reduction purposes. Hu *et al.* [60] attempted to utilize autoencoder in QSAR studies. A fully connected autoencoder model was developed to predict drug likeness to identify compound candidates that could become marketed drugs. They reported that autoencoder provided promising insights for extracting meaningful features. Autoencoder is an appealing choice for developing models for molecule generation. Gómez-Bombarelli *et al.* [61] developed a model to generate molecules using the autoencoder architecture. Their model was also able to predict the properties of molecules using the latent space (the bottleneck hidden layer) of the autoencoder. Bjerrum and Sattarov [62] analyzed the effect of choosing different representations in the input and output during the training of autoencoder to produce vector representations of molecules that are considered descriptors of the model. They showed that properties represented by autoencoder are influenced significantly by the choice of the training data.

A variety of dimensionality-reduction techniques have been applied to QSAR, such as genetic algorithms (GA), [63] partial least squares (PLS) [64], a hybrid GA and PLS approach [65], K-means clustering algorithm [66], and ant colony optimization (ACO) [67]. However, PCA was delievered superior results in QSAR [40].

Idakwo *et al.* [26] presented one of the few reviews of feature-extraction methods in QSAR prediction. Various review studies focused on feature-selection methods in QSAR, such as the work of Danishuddin and Khan [3] and [15]. There is a major need for comparative studies focused on feature-extraction methods in QSAR, as the relevance and efficiency of feature-extraction techniques (besides linear PCA) in QSAR modeling have not yet been explored.

III. DIMENSIOANLITY REDUCTION

For a given dataset with n data points or records and p features, high-dimensional data is observed when the number of features p is higher than the number of records n, as $p>n$ [68]. Dimensionality-reduction has the following advantages [6], [17], [27], [34]:

1) Computationally less expensive: The lower number of features reduces computational overhead on the hardware resources. This ensures the preservation of time and storage.
2) Easier visualization: When data are visualized in multi-dimensional space, they are more difficult to analyze and understand.
3) Improved prediction performance: When data are higher dimensional, they can be noisy, sparse, or have missing values. Such sparsity affects the capability of machine learning models; hence, it affects the accuracy of the model.

There are two broad approaches to reduce the number of features in datasets: feature extraction and feature selection.

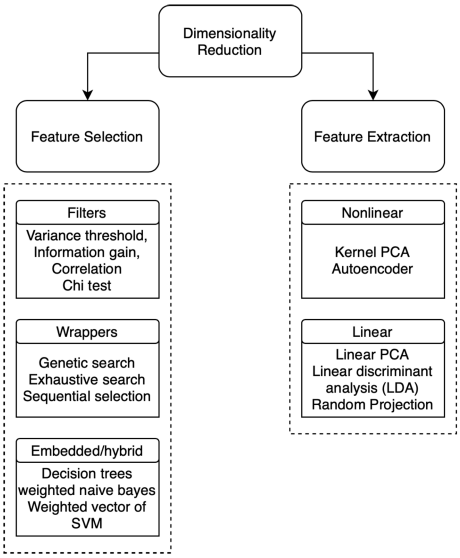


FIGURE 2. Dimensionality reduction techniques.

A. FEATURE SELECTION

Feature selection is a process in which a subset of the features' space is chosen according to its relevance to the output of the classifier. The ultimate goal is to extract the most effective subset of features and to remove redundant irrelevant information. The selection methods are categorized into filter methods, wrappers, and embedded/hybrid methods [3], [34].

1) **Filters** Filter methods simply select the feature subset independent of the classifier by calculating a feature relevance score or by applying a feature-searching algorithm [69]. For example, the variance of each feature is computed based on a certain threshold without considering the relationship between the features and the classifier. This method is simple and quick but yields lower accuracy compared to wrapper and embedded methods. The most popular filter methods are the correlation coefficient score, Chi squared test, and T-test.
2) **Wrappers** In contrast, wrapper methods work as a predictive model by selecting the feature subset and calculating the error in the classifier function. This method outperforms the filter method because it works on optimizing the classifier performance by calculating the error until the optimal feature subset is selected [70]. However, it is not as simple and quick as the filter method, as it requires more computational complexity and time. A third approach in feature selection is called the embedded or hybrid method.
3) **Embedded/Hybrid** Embedded methods work similarly to wrappers in terms of being dependent on the classifier; however, they requires less computational complexity. Embedded methods differ from wrappers, as the selection of features is dependent on the type of classifier used and might not work with other classifiers. An example of an embedded method is random forest, in which multiple random forests are created iteratiely until a forest of features with the lowest error rate is selected [5].

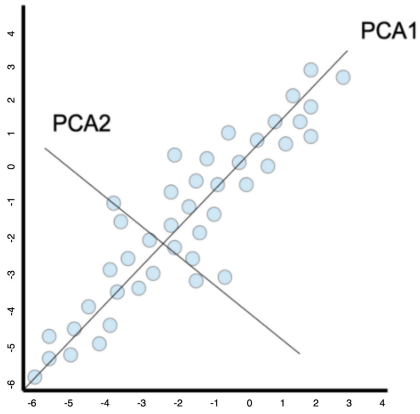


FIGURE 3. Principal component analysis.

Another feature-selection method classification is based on dataset characteristics: supervised (labeled dataset), semi-supervised (partially labeled dataset), or unsupervised (labeled dataset) [69], [71]. Recent developments in various fields require diverse types of feature-selection methods, such as online feature selection, ensemble feature selection, and methods for extreme datasets [69].

B. FEATURE EXTRACTION

Feature-extraction or feature-reduction techniques identify a new subset of features that are transformed or combined from the original feature space to obtain a more significant set of features. Feature-extraction methods can be linear or non-linear. Linear PCA is the dominant feature-extraction technique in QSAR [17]. However, there are many recent feature-extraction methods, such as graph-based methods [55], which have demonstrated success in dealing with image datasets. PCA is used as a baseline to compare the proposed dimensionality-reduction techniques in this study. Next, we provide a brief introduction of each feature-extraction technique used in this study.

1) PRINCIPAL COMPONENT ANALYSIS

PCA minimizes redundancy by measuring variance and eliminating redundant and noisy features. It is commonly used in QSAR studies for dimensionality-reduction purposes [40]. PCA uses a matrix to calculate the covariance by analyzing inter-correlated dependent variables to obtain a new set of variables called principal components or eigenvectors. The first principal component contains the most effective set of variables. The second component contains the second most important variables and so on, as shown in Figure 3.

Eventually, PCA attempts to identify the top K principal components, and the remaining less influential variables are dropped [17]. The principal component (eigenvector) has a value known as the eigenvalue that corresponds to the variance. When the eigenvalue is large, the principle component

represents a large variance in the data. In the QSAR context, this definition implies that PCA finds a set of descriptors that best characterizes and describes compounds pertaining to a specific QSAR activity or property with minimum redundancy. Eigenvectors and eigenvalues can be calculated as a square matrix. λ is an eigenvalue for a matrix A, i.e,:

$$\det(\lambda \mathrm{I} - \mathrm{A}) = 0 \tag{1}$$

where, I is the identity matrix of the same dimension as A, which is a required condition for the matrix subtraction as well as in this case, and "det" is the determinant of the matrix. For each eigenvalue λ, a corresponding eigenvector, v, can be determined by the following equation:

$$(\lambda \mathrm{I} - \mathrm{A})\mathrm{v} = 0 \tag{2}$$

Eigenvalues are sorted from the largest to the smallest to provide the principle components in the order of their significance. To reduce the dimensionality, we selected the first K components and dropped the rest.

2) KERNEL PRINCIPAL COMPONENT ANALYSIS

Linear PCA is a powerful technique for reducing dimensionality, but it lacks the ability to identify and perceive all structures in the feature space of a dataset. Because linear PCA assumes a linear relationship between variables and only functions with numeric values, an alternative non-linear PCA emerged that can handle nonlinear representations of data [20]. Given a dataset x_i, where $i = 1, 2, 3 \ldots, N$, and x_i is a D dimensional vector that needs to be projected to a new dimension space M. The datapoint x_i is transformed to a non-linear representation $\Phi(\mathrm{x})$. Similar to PCA, the covariance matrix $M \times M$ is calculated by:

$$\mathbf{C} = \frac{1}{N} \sum_{i=1}^{N} \phi(\mathbf{x}_i)\, \phi(\mathbf{x}_i)^{\mathrm{T}} \tag{3}$$

Eigenvalues and enginevectors are calculated by:

$$\mathbf{C}\mathbf{v}_k = \lambda_k \mathbf{v}_k \tag{4}$$

The kernel principle component is calculated as follows:

$$y_k(\mathbf{x}) = \phi(\mathbf{x})^{\mathrm{T}} \mathbf{v}_k = \sum_{i=1}^{N} a_{ki} \kappa(\mathbf{x}, \mathbf{x}_i) \tag{5}$$

The kernel matrix is calculated directly from the training data point x_i [72]. The most important aspect of KPCA is that it can identify the complex non-linear structures found in QSAR [21]. As with PCA, KPCA creates a covariance matrix to determine features with high variance. While identifying the eigenvector (principle component) and eigenvalue, KPCA differs in that it maps each data point to another vector space using the Φ (x) function. Subsequently, it applies linear PCA to each mapped data point in the new dimension.

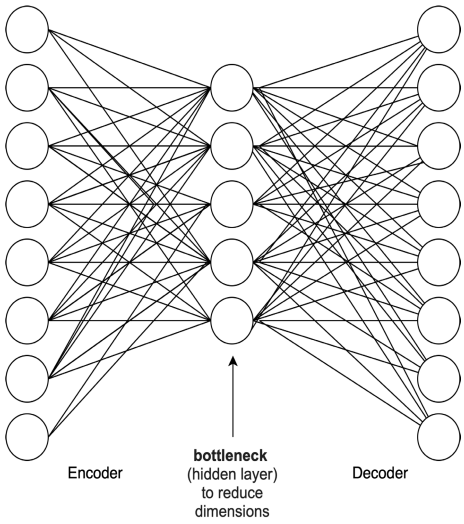


FIGURE 4. Autoencoder architecture.

IV. AUTOENCODER

Deep learning algorithms are widely used for classification purposes. Due to their ability to extract hidden patterns and important features, they are also employed for dimensionality-reduction purposes [18]. Autoencoder is a branch of ANN that is recognized for dimensionality reduction. However, it has not been fully implemented in QSAR studies. Autoencoder is a feed-forward neural network that is composed of an encoder, one or more hidden layers, and a decoder, as shown in Figure 4. The input to the hidden layer is called the encoder, and the hidden layer to the output layer is the decoder. Autoencoder maps the vectors in the input layer to the same number of vectors in the output layer [19]. An autoencoder model is expected to reconstruct the same inputs that originally passed through the input layer. Thus, the decoder works as a mirror image of the encoder with a matching number of neurons. Autoencoders are widely used for image compression, dimensionality reduction, and information retrieval [8]. While mapping the instances x_i, the encoder maps the input x_i to a reduced representation y_i using a function $g()$:

$$y_i = g(Wx_i) \tag{6}$$

where $g()$ is a sigmoid function, and W is a weight matrix $d_y \times d_x$. Using the same reduced representation y_i, the decoder reconstructs x_i' as:

$$x_i' = f\left(W' y_i\right) \tag{7}$$

Autoencoders are typically trained with both the encoder and the decoder. However, to utilize it in the context of dimensionality reduction, the smallest hidden layer in the architecture (also known as the bottleneck) is used to compress the input to the lowest level of space (called latent space) to achieve a dimensionality-reduction effect [16]. The decoder is used in the training process to measure the error rate of the model but not to restore the original input dimension.

To handle more complex data structures, the hidden layers of autoencoder can be expanded to comprise multiple hidden layers instead of a single one. This architecture is known as a dGAE [24], which we adopted in this research.

V. RANDOM PROJECTION (RP)

RP is a linear technique used for dimensionality-reduction purposes that works based on matrix multiplications. This technique is inspired by the Johnson-Lindenstrauss lemma theorem [39]. For a given dataset with d dimensions that must be reduced to k dimensions and n number of datapoints (instances), $Y = RX$ We note that N data points can be mapped from:

$$R^d \rightarrow R^k$$

where X_{d*n} is the original matrix, and R_{k*d} is a random matrix. This is performed while preserving the distance between the data points in the original high-dimensional space and the points in the new sub-space [22]. A major advantage of RP compared to PCA and autoencoder is its simplicity and low computational costs [22]. An RP random matrix can be generated using Gaussian distribution, which projects a data matrix as (dkN). A lesser computational sparse distribution proposed by Achlioptas [23] reduced the complexity further with a distribution that projects the data matrix as (ckN), where c represents non-zero feature values. This simple ''sparse'' distribution can be expressed as follows:

$$\text{rij} = \sqrt{3} \begin{cases} +1p = 1/6 \\ 0p = 2/3 \\ -1p = 1/6 \end{cases} \tag{8}$$

where rij is an element of matrix R. This simplified distribution ensures that all zero variance values of rij would still produce a projection that satisfies the Johnson-Lindenstrauss lemma theorem [39] with less computational overhead, as all computations are performed on non-zero integer values [25]. Both GRP and SRP are investigated in this study.

VI. RESEARCH METHODOLOGY

One way to assess the effects of the proposed feature-extraction techniques is to test and compare their impact on classifier performance [73]. Our research methodology demonstrates the variations in performance and accuracy of the same classification model based on different data representations. We conducted a series of experiments with five feature-extraction techniques: KPCA, autoencoder, GRP, SRP, and PCA. PCA is one of the most used feature-extraction technique in QSAR modeling and BBB permeability [40], and it was the baseline with which the other proposed techniques were compared. We followed the proposed steps of [2] to solve the QSAR classification problem. A walk-through of these steps can be summarized as follows:

1) Descriptors (features) calculation
2) Dataset preprocessing and curation
3) Feature extraction technique
4) Classification model
5) Validation

We performed descriptor calculation under an experimental setup. Five different techniques were developed for the feature extraction and fed to the classifier. The performance of the feature-extraction techniques was measured based on the classifier performance.

A. EXPERIMENTAL SETUP

The predicted QSAR activity adopted for our experiment was the permeability of compounds to the BBB, which is a binary classification problem with two predicted class labels (BBB+) and (BBB−) [74]. Building a model using a benchmark dataset is crucial for measuring the performance of the classification model. The benchmark dataset was acquired from Wang *et al.* [13] and as composed of 2350 compounds and 6394 generated fingerprints, 1D, 2D, and 3D descriptors. The dataset was imbalanced with 1803 (BBB+) and 547 (BBB−) class labels. Th synthetic minority oversampling technique (SMOTE) was employed to resample the dataset to 1803 instances corresponding to class (BBB+) and 1803 instances corresponding to class (BBB−). Compounds in QSAR are encoded to a special numerical representation called Simplified Molecular Input Line Entry Specification (SMILES). SMILES is a one-line notation that describes molecules and encapsulates all the information related to the compound structure and activities. This includes the atomic number, bonds orders, branches, rings and so on [2]. Descriptors in this study were calculated using AlvaDes [75] and the Online Chemical Modeling Environment (OCHEM) [76]. AlvaDesc was used for calculating 1D and 2D descriptors and fingerprints (MACCS 16, and Hashed). OCHEM was used to calculate 3D descriptors and to obtain their atom coordinates and atomic partial charges. OCHEM utilizes the Chemistry Development Kit (CDK) [77] package for the calculation and manipulation of QSAR models and the calculation of their respective descriptors. Multiple 3D descriptors are supported by CDK, such as Weighted Holistic Invariant Molecular (WHIM), Charged Partial Surface Area (CPSA), Gravitational Index, Molecular Distance Edge (MDE), and Geometrical Shape Coefficients of the radius–diameter diagram [78]. The BALLOON optimizer was used to obtain the partial charge and atom coordinates of 3D descriptors. In total, 6394 descriptors and FPs were calculated and merged into a single file for preprocessing.

B. PREPROCESSING

To ensure that dimensionality-reduction was performed on a clean consistent dataset, the following preprocessing tasks were performed.

1) **Data cleaning:** In this step two primary tasks were required: handling records with no descriptor values and handling records with certain missing descriptors values. During the process of calculating descriptors, eight records had null values for all the descriptors.

This implies that the tools and optimizers failed to calculate their corresponding descriptors. These eight records were dropped completely, as it was impossible to compensate for all the descriptors values. The missing values were replaced by zero, as these compounds had a corresponding value for all descriptors. Therefore, mean value or imputation could not be used as a substitute.

2) **Data transformation (Scaling:)** Once the descriptors were calculated and the missing values were handled, our dataset had a varying range of values. Certain descriptors had zero or negative values, whereas others had values reached up to 10000. Scaling or feature scaling is a preprocessing technique used to normalize the range of features values in a given dataset [91]. For instance, instead of having values ranging between -10 and 10000, it can be scaled to a fixed range between 0 and 1. MinMax scaler is one of the well-known scalers, which transforms features by scaling each feature to a given range. It works as shown in the following equation:

$$X_{norm} = \frac{X - X_{min}}{X_{max} - X_{min}} \tag{9}$$

Min and max indicate the range of the feature. The values of 1D, 2D, and 3D descriptors were transformed using the MinMax scaler to a range between 0 and 1.

3) **Handling class imbalance:** A dataset is considered to be imbalanced if the classes are not equally represented [79]. Imbalanced distribution of classes when building machine learning models directly affects the performance of the model, especially for minority class accuracy measures such as (specificity). To address the challenge of an imbalanced dataset, resampling techniques are used. Wang *et al.* evaluated multiple resampling methods, including Random Undersampling (RUS) and the Synthetic Minority Oversampling Technique (SMOTE), Adaptive Synthetic Sampling Method (ADASYN), Evolutionary Neural Network (ENN), and Weight Loss Function (WLF). According to their study, SMOTE performed consistently well with a BBB dataset. Following the results reported by Wang *et al.* [13], we solve the imbalanced distribution of classes in the BBB permeability dataset by applying (SMOTE). SMOTE was first introduced by Chawla *et al.* [80]. It uses K-Nearest neighbors to add new instances to the minority class. In this experiment, the minority class is represented by the compounds with low permeability (BBB-). K-Nearest neighbors synthesize new data points (instances) by creating new ones between two original points that share similar features. The SMOTE resampling technique transformed the dataset from 1803 and 547 compounds in the positive and negative classes, respectively, to 1803 compounds in each class.

C. FEATURE EXTRACTION

Dimensionality reduction is a preprocessing step that occurs before building a classification model. Because of its significance to this study, we explain the feature-extraction experiment in a separate section. Five feature-extraction methods were employed: PCA, KPCA, autoencoder, GRP, and SRP. Our experiments were performed using the PyCharm python environment. All feature-extraction techniques were imported using scikit-learn. The imported dataset was the resampled dataset with 1803 (BBB−) instances and 1803 (BBB+) instances. The parameters choices were finalized to employ each technique as follows:

- **KPCA:** The radial basis function (RBF) kernel is a popular kernel function with KPCA. RBF was used to project the data in higher-dimensional space; consequently, it becomes linearly separable. For this experiment, we set the number of jobs to be 10 jobs running in parallel, which provided faster computing. We set the number of components to the default value.
- **PCA:** To construct a linear PCA, data were projected using a singular value decomposition to transform them to a low-dimensional space. The number of components was set to the default value.
- **GRP and SRP** For GRP construction, a Gaussian random matrix was used to reduce the high-dimensional data into a low-dimensional Euclidean space. We experimentally set the number of components to 3700, which represents the desired number of descriptors projected after applying GRP. The same number of components were used with SRP.
- **dGAE:** Five encocder layers and five decoder layers were used to construct the autoencoder model. During the experiment, running autoencoder with the original 6394 descriptors resulted in a significantly high number of computations, which caused a system shut down and incomplete execution. The experiment was repeated several times; however, it failed repeatedly. Eventually, we had to change the initial number of neurons corresponding to the descriptors from 6394 to 4500. The first encoder layer is the input layer, whereas the remaining four layers are hidden layers, with 4500, 4200, 4000, and 3750 neurons, respectively. The last encoder layer is the output layer, which produced the reduced dimension feature space. Two dropout layers were used with a dropout rate of 30%. The last encoder layer is the the bottleneck layer, with a latent representation of the compressed data. The decoder layers were only used for training the network. After training, the output of this network was set to the last encoder layer to extract the reduced features in the latent space. The autoencoder hyperparameter is similar to that of a deep learning model. A rectified linear unit (ReLU) activation function was applied. We experimented with both Adam and rectified Adam optimizers.

TABLE 1. Model hyperparameter for the deep learning classification model.

Hyperparameter	Value
DL Library	Keras
Number of hidden layers	3
Number of hidden layers nodes	256,128,64
Activation function	Input layer: ReLU Tanh
Batch size	200
Number of epochs	100
Optimizer	Adam
Regularization	2 layers of Batch Normalization
Scaler	MinMax scaler
Learning rate	0.01
Validation	Tenfold cross-validation
Loss	Binary crossentropy

The final set of features obtained by PCA and kernel PCA included 3606 features, whereas dGAE and RP methods reduced the number to 3000 features.

D. CLASSIFICATION MODEL

A deep learning classification model was developed with five layers: an input layer, three hidden layers, two batch normalization layers, and an output layer. The activation function for the proposed model was ReLU with the Adam optimizer with a learning rate of 0.01. Table 1 summarizes the hyperparameters for the classification model.

For the experimental comparison proposed in this paper, we iteratively executed the model six times, with different datasets in each run. The first run included the complete high-dimensional dataset with 6394 features (descriptors). The remaining five runs included each of the feature-extraction techniques. Because each technique transformed the dataset differently, the new dataset with the reduced number of features was input into the classification model to test and compare the classifier's performance with respect to each feature-extraction technique. The full model architecture is illustrated in Figure 5. Validation of the classification model was performed using K-fold cross-validation, which divides the training set into groups known as (folds). In every iteration, one fold of the training set is left out for testing, and the training is performed using the remaining folds. For our experiment, tenfold cross-validation was performed.

VII. EXPERIMENTAL RESULTS

To assess the model's performance, four primary measures were considered: accuracy, specificity, sensitivity, and AUC. The accuracy indicates the overall performance of the model pertaining to the true positive assessment [81]. However, it is not regarded as a good indication to the performance of the model, as it only considers the correctly classified instances regardless of the falsely classified ones. In addition, it is not considered a good measure for imbalanced datasets [82]. Specificity is the percentage of compounds that are correctly classified as (BBB−) by the model and sensitivity is the percentage of compounds that are correctly classified as (BBB+) by the model.

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{10}$$

$$Specificity = \frac{TN}{TN + FP} \tag{11}$$

$$Sensitivity = \frac{TP}{TP + FN} \tag{12}$$

Here, TP refers to true positive, which indicates the number of compounds that were correctly classified as "positive" or (BBB+); TN is true negative, which indicates the number of compounds that were correctly classified as (BBB−) by the classifier; FP is false positive, which indicates the number of compounds that were inaccurately classified as BBB+, and FN represents false negative, which is the number of compounds that were mistakenly classified as (BBB−). Receiver operating characteristics (ROC) graphs are important for visualizing classifier performance and comparing different algorithms. It shows the TP rates in comparison to FP rates [83].

A. RESULTS

Interpreting the performance of dimensionality-reduction techniques is essential to determine the technique that was able to model the problem most efficiently. Three main approaches were considered to compare the performance of the feature-extraction techniques: (i) comparing the accuracy measures of the classification models [73], (ii) designing scatter plots to visualize any existing correlation between features in the low-dimensional space [28], [29], and (iii) visualizing class separation with an external widely used Modified National Institute of Standards and Technology (MNIST) dataset. Evaluating the proposed dimensionality-reduction techniques using two datasets (i.e., the BBB permeability and MNIST datasets) helped us understand the consistency in the performance of each technique.

B. VALIDATION WITH CLASSIFICATION MODEL

Table 2 lists the results obtained using different feature-extraction techniques. After applying each technique, the dataset with the reduced dimensions was input into a feed forward deep neural network (FFDNN) classification model. The primary goal was to compare the classifier accuracy of each technique and obtain insights about the effect of each feature-extraction technique on a classification model. Table 2 presents the performance of each model after training and testing to reveal the model's overfitting problems. Overfitting can be detected when a model learns well during training, which yields high accuracy scores, but demonstrates significantly inferior performance on the testing set. In the case of the accuracy measures, the overall accuracy of the model does not precisely demonstrate the ability of the model to predict class labels in the classification problem. The AUC measure is important when assessing the performance of dimensionality-reduction techniques and binary

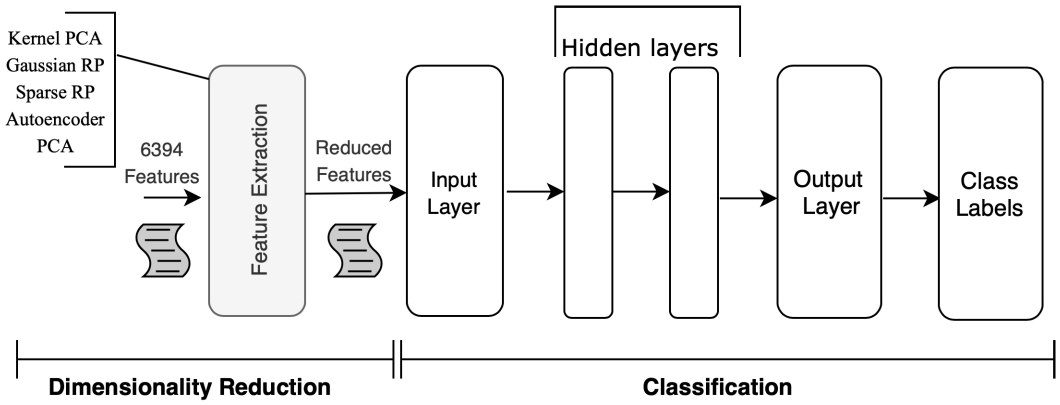


FIGURE 5. Classification model architecture.

TABLE 2. Performance of Dimensionality Reduction Techniques (ACC = Overall accuracy, Sens = Sensitivity scores, Spec = Specificity scores, AUC = Area under the curves, RP = Random Projection, FFDNN = Feed Forward Deep Neural Network), DR = Dimensionality Reduction, MCC = Matthew Correlation Coefficient.

Dimensionality Reduction Technique	Training set			Test set				
	ACC	Sens.	Spec.	ACC	Sens.	Spec.	AUC	MCC
dGAE (RAdam)	97.19	96.90	97.48	93.67	94.01	92.72	97.85	86.71
dGAE (Adam)	93.61	96.00	91.20	88.64	92.30	85.15	95.37	77.54
KPCA (RAdam)	99.93	99.86	100	95.84	94.13	97.56	98.88	91.72
KPCA (Adam)	100	100	100	94.98	94.31	97.56	98.64	91.99
Linear PCA	100	100	100	96.23	95.32	96.92	96.76	91.11
Gaussian RP	99.83	99.72	99.94	95.47	92.76	98.22	98.34	92.78
Sparse RP	99.86	99.75	99.97	95.25	92.55	97.92	98.29	90.63
FFDNN no DR	99.86	99.72	100	95.11	92.15	98.11	98.38	90.40

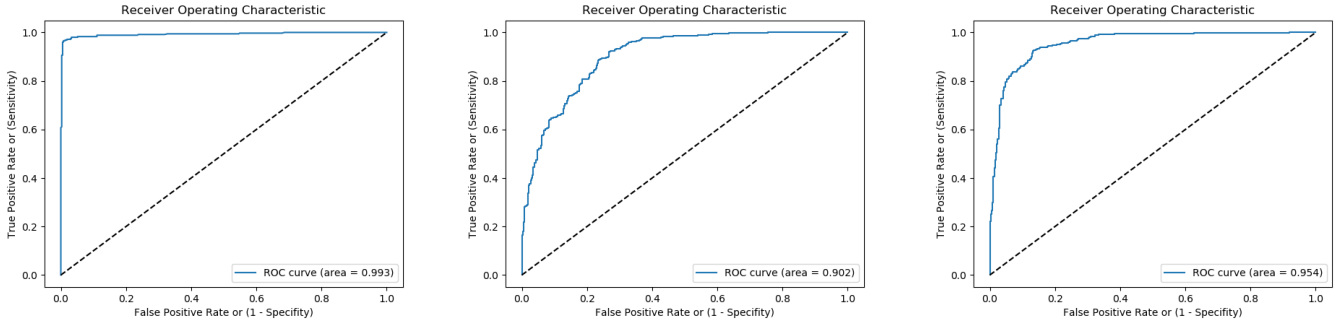
classification models [84]. AUC exposes the ability of the model to separate the two classes in a classification problem. In the BBB permeability context, it can distinguish the compounds that are able to penetrate the BBB [85], [86].

The main objective of this study was to investigate the performance of feature-extraction techniques on a high-dimensional QSAR dataset. We observed that the AUC scores of all feature-extraction techniques exceeded those obtained with PCA. The best AUC score was achieved with KPCA, which indicates that the non-linearity of KPCA was able to capture the structure of descriptors better than PCA and permitted a better distinction between classes. Figure 6(a) show the ROC graph of KPCA.

While observing the accuracy, sensitivity, and specificity, it was evident that employing a resampling technique caused the negative class ''specificity'' scores to increase dramatically. Consequently, it caused a minor decrease in sensitivity scores as well. Considering this, we observed that linear PCA achieved the highest overall accuracy score, but not in specificity or AUC scores. This indicates that the overall accuracy does not provide sufficient insight about the mistakenly classified compounds [87]. The specificity scores were unsatisfactory in the research studies reported in the literature, including the works that employed resampling techniques [13], [88]. In addition, many models that were developed with resampled datasets in a BBB permeability context achieved a high score in one measure but a relatively low score in another.

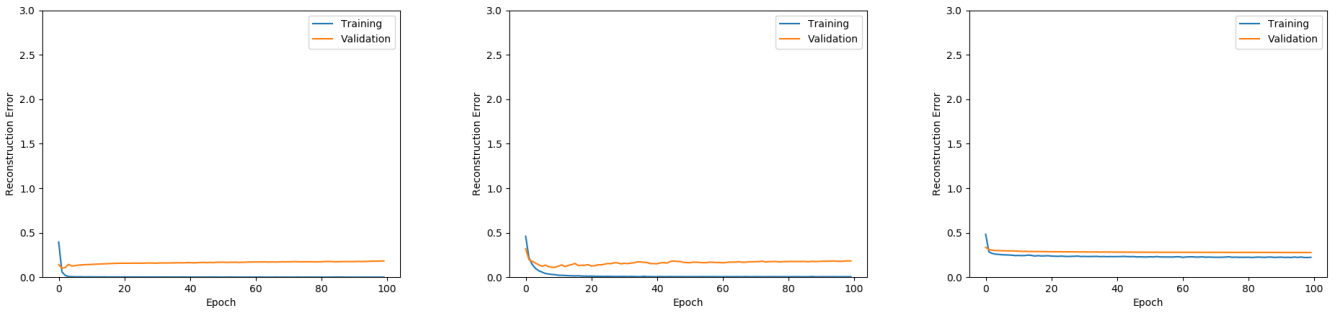
RP techniques demonstrated promising results proving that their simplicity and low computational cost were not obtained at the expense of specificity and AUC scores. GRP achieved the highest ''specificity'' score followed by SRP. Linear PCA and KPCA achieved the highest sensitivity scores representing the positive class.

Conversely, dGAE delivered average results when compared to the other techniques but still outperformed PCA in terms of AUC scores. However, the construction of the autoencoder model was responsible for this limitation, as the input layer had only 4300 neurons representing descriptors, which excluded 2094 descriptors that were not even considered. This initial drop in the number of descriptors ultimately caused a loss of important information and affected the prediction accuracy of the model. ROC plots of the autoencoder with both Adam and



((a)) ROC KPCA ((b)) ROC (dGAE) Adam ((c)) ROC (dGAE) RAdam

FIGURE 6. Adam vs. RAdam with dGAE.



((a)) Loss PCA ((b)) Loss RP ((c)) Loss dGAE

FIGURE 7. Reconstruction error.

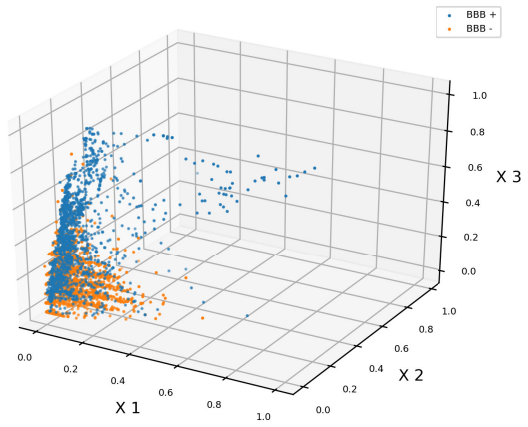
RAdam optimizers are shown in Figures 6(b) and 6(c). An apparent improvement in the performance of autoencoder in separating classes was achieved using the RAdam optimizer.

The primary goal of training a classifier is to minimize the reconstruction error between training and validation. Reconstruction error plots are useful to show the consistency in the performance of a classifier using an unseen dataset. Figure 7 illustrates the reconstruction errors of autoencoder (dGAE), PCA, and RP. Despite the low overall accuracy scores of autoencoder, it demonstrated minimal overfitting, as there was an obvious near convergence between the training and validation data. Considering the overall performance of all techniques, we concluded that KPCA and RP demonstrated superior performance corresponding to various accuracy measures in comparison with autoencoder. Although autoencoder did not achieve the highest accuracy, it showed some potential when considering that the initial number of descriptors was not equivalent to that of other techniques. In addition, it demonstrated minimal predisposition to overfitting. Thus, we can conclude that feature extraction with PCA is no longer the best option for handling high-dimensional classification problems in QSAR.

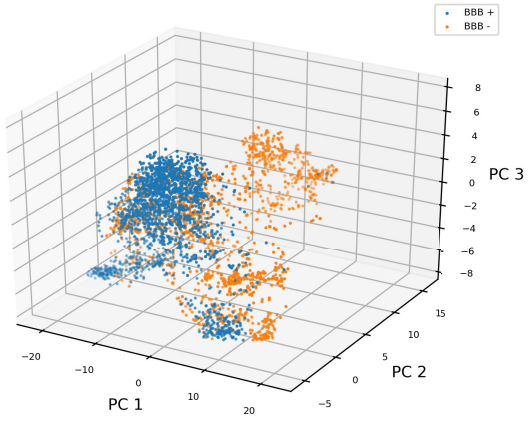
C. VALIDATION WITH A 3D SCATTERPLOT VISUALISATION

Visualizing the efficiency of a feature-extraction technique in identifying important features and separate classes provides invaluable insights. Scatter plots are a popular visual-encoding technique for observing the relationships between features. The relationship between data points were expressed through the plot axes representing data points pertaining to two or three features [28]. The positioning of data points between the axes indicates the value of each data point with respect to these features.

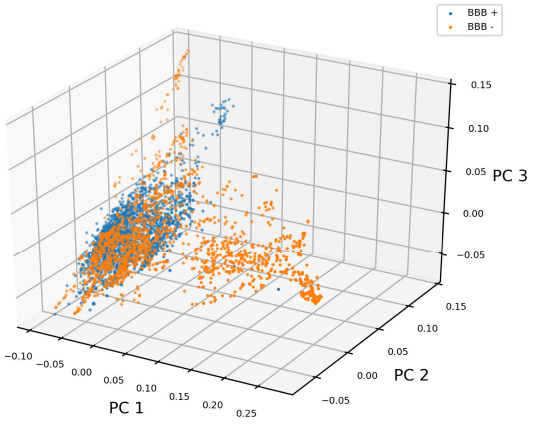
Scatter plots demonstrate the correlation between features (i.e., positive, negative, or no relationship at all). The motivation behind the visualization of a dataset is to detect the undesired existence of a positive relationship between features after employing dimensionality reduction. A positive relationship or "low variance" between descriptors indicates that the chosen features are correlated and hence have no implications on the data. For instance, if the bond count of a compound and its weight are positively correlated in the sense that their values change positively or negatively on the same scale, then the dimensionality-reduction technique failed to remove redundant data since one of these features can be deduced from the other [29]. In addition, scatter plots are useful for clustering a dataset into groups. A clear grouping indicates a good separation of classes. By applying this in our dataset, we search for separation in the two class labels (compounds that are classified as BBB+ or BBB−). Further, the variables in the low space were randomly picked and repeated for each technique to ensure overall consistency and to confirm that a certain observation was not specific to certain features.



((a)) Original dataset

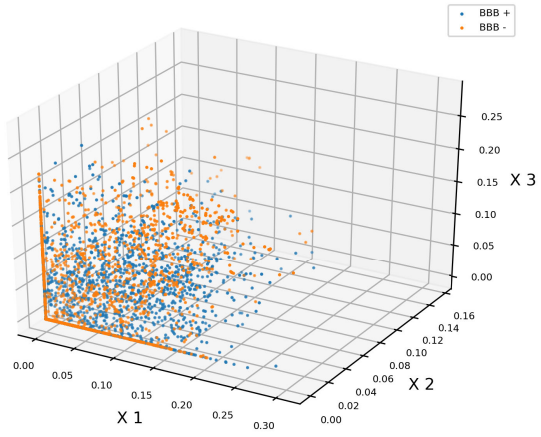


((b)) PCA

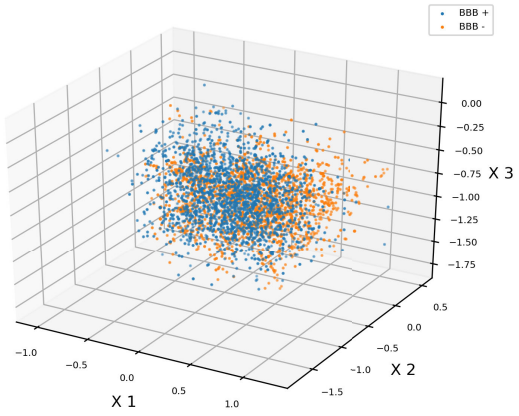


((c)) KPCA

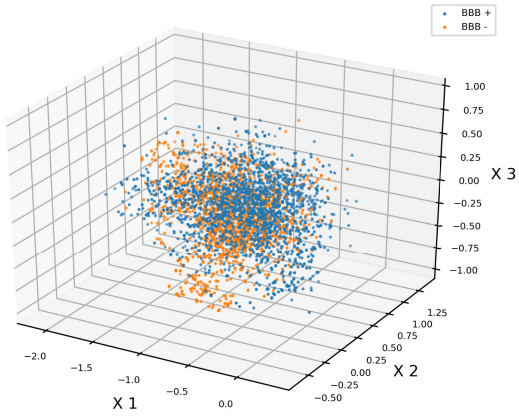
FIGURE 8. Three-Dimensional Scatter Plots of the Original Dataset, and After PCA and KPCA.



((a)) dGAE



((b)) Gaussian RP

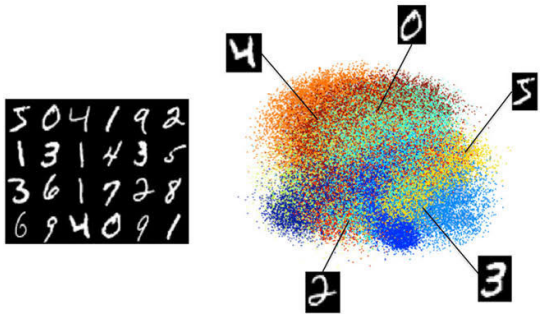


((c)) Sparse RP

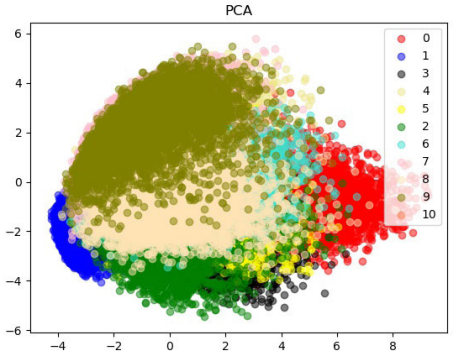
FIGURE 9. Three-Dimensional Scatter Plots of dGAE and Random Projection (RP).

1) VISUALIZATION WITH QSAR DATASET

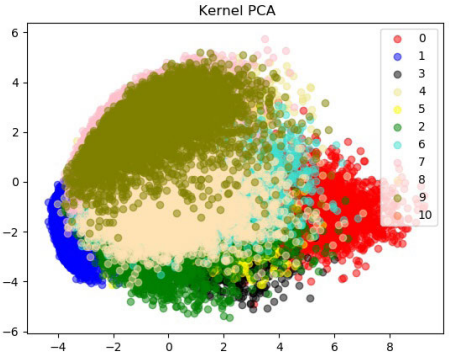
Designing scatter plots to visualize the positioning of data points with respect to features provides useful insights about each technique. The scatter plot of the original dataset, illustrated in Figure 8(a), shows a tight linear and positive correlation between the features. This indicates the existence of feature redundancies in the original high-dimensional dataset. Figure 8(b) and 8(c) illustrate the transformation of data points after applying PCA and KPCA. The scatter plots show the distribution of data points with respect to the top three principle components. PCA and KPCA arrange principle components according to the highest variance; hence, the first



((a)) MNIST dataset

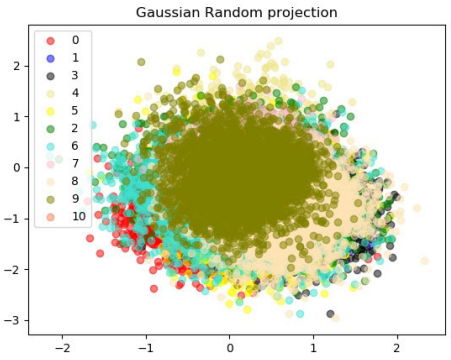


((b)) PCA with MNIST dataset

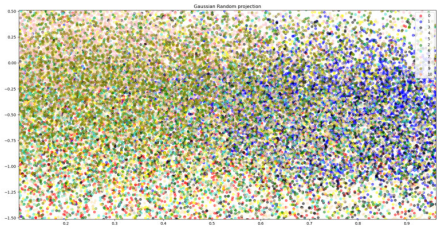


((c)) KPCA with MNIST dataset

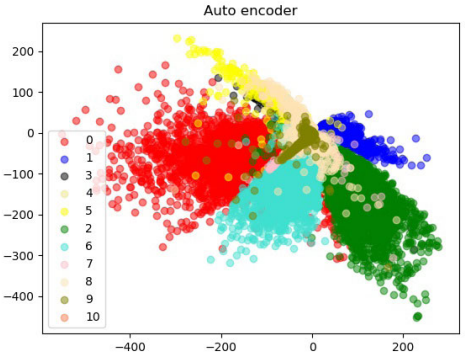
FIGURE 10. PCA and KPCA with Modified National Institute of Standards and Technology (MNIST).



((a)) RP with MNIST



((b)) Zoomed RP scatterplot showcasing no class separation



((c)) dGAE with MNIST dataset

FIGURE 11. RP and dGAE with MNIST.

three principle components were selected. Both techniques show low correlation between descriptors when compared with the original dataset. This observation is an indication that certain important descriptors with high variance were preserved. Both techniques demonstrated good separation between the class labels and a clear grouping of the positive class. Out of the five models, KPCA achieved the highest score on the area under the ROC curve (AUC). This indicates that KPCA achieved the best performance in distinguishing the two class labels as illustrated in 8(c).

Figure 9(b) and 9(c) illustrate the positioning of data points after employing GRP and SRP. Both RP techniques show no visible correlation between features, which indicates that RP retained most of the relevant features from the original dataset. Linking the high accuracy of RP in the negative class with the scatter plot images provided useful insights on the performance of this technique. The high specificity scores of RP demonstrated that RP was able to identify the negative class more precisely in comparison with dGAE and PCA. The scatter plots illustrated in Figure 9(b) and 9(c) show an obvious grouping of (BBB-) datapoints. The low computational cost of RP has no implications on its ability to extract useful features and separate class labels efficiently in a binary classification problem. The scatter plot demonstrating the dGAE performance is shown in Figure 9(a). An evident disjointedness between data points is illustrated, and no distinguish-

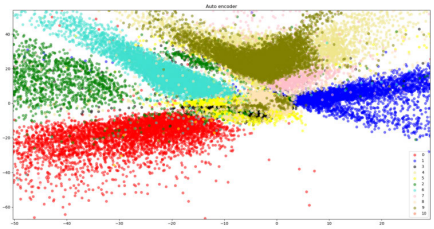


FIGURE 12. Zoomed dGAE scatter plot to showcase the separation of class labels.

able positive correlation was detected between the features. It can be noted that the data points are clearly separated with less overlapping when compared to other techniques. Although the two classes are not clearly grouped at this stage, the distance between data points indicates a clear separation of instances of different class labels [89]. This observation conforms with the high AUC score of 97.85 obtained by dGAE.

2) VISUALIZATION WITH MNIST DATASET

To test the consistency of the developed feature-extraction techniques, we explored their performance with the MNIST database. MNIST is a handwritten digit database that is widely used for image recognition problems [90]. Although MNIST is not a QSAR dataset, this experiment was useful for three primary reasons: 1) to detect any unusual patterns that contradict those obtained in the BBB permeability dataset; 2) to provide a broader understanding of the dimensionality-reduction technique in a multi-class classification problem; and 3) to verify the effectiveness of these techniques using a widely tested benchmark dataset. MNIST consists of 70000 image samples and 784 features representing image pixels. The dataset is split into 60000 and 10000 samples for training and testing, respectively. The vector value ranges from 0 to 1, which symbolizes the intensity of the pixel. The same parameters and experimental setup used in the BBB permeability experiment were applied to the MNIST dataset. The original distribution of MNIST data points before feature extraction is shown in 10(a), with obvious overlapping between the 11 class labels.

Figures 10(b) and 10(c) show the transformed dataset using PCA and KPCA. These techniques improved the class boundaries marginally, as the majority of classes are visible when compared to the original dataset. However, apparent overlapping of the class boundaries continues to exist. Classes representing the numbers 1 and 0, shown in blue and red, respectively, are separated better than the others. Most digit classes were identified using linear PCA and KPCA. However, when analyzing the RP scatter plot, it was not able to separate instances or class labels, as shown in Figure 11(a). There may be two reasons for this issue. First, the linearity of RP may have failed to handle a non-linear image recognition problem. Second, the method by which RP projects features from a high- to a low-dimensional space by maintaining the distances regardless of the data structure is a drawback that results in a loss of important information. A zoomed-in image of RP shows no visual class separation, as shown in Figure 11(b).

Of the five dimensionality-reduction techniques, autoencoder showed the best class separation, as illustrated in Figure 11(c) and 12. The apparent space between classes proves that autoencoder was able to extract the most useful information in the low-dimensional space.

It also confirms that the lower accuracy obtained with dGAE in the classification model, as shown in Table 2, was due to the missing features, which were not encoded due to the computational complexity of autoencoder. This motivates us to investigate a new method for reducing the high computational cost of autoencoder while still utilizing its capability to encode important hidden features in the transformed low-dimesional space.

VIII. CONCLUSION

In this paper, we provided a new outlook on dimensionality-reduction techniques in QSAR modeling. Based on previous studies that focused on feature-selection techniques in QSAR, we conducted the first experimental analysis of five feature-extraction techniques. In addition to reviewing feature-extraction methods in a high-dimensional QSAR dataset, we provided new insights about the ability of each technique to handle the negative class and separate the binary class labels more accurately when compared to the baseline (linear PCA). This study proved that, through the accurate transformation of feature space to a low-dimensional Euclidean space, extraction techniques could substantially increase the accuracy of the classifier for all class labels. Our research further showed that the performance and accuracy of the same classification model varied as a result of different data representations.

This paper introduced a new approach for comparing feature-extraction methods in QSAR. Further research is encouraged to investigate other feature-extraction techniques in a high-dimensional QSAR dataset. Investigation of autoencoder with higher computational capabilities could improve the performance of proposed approach. Alternatively, hybrid approaches should be considered to substantially decrease computational overhead. Although the developed models improved classification accuracy, a question remains with regard to reversing the transformation of features, as the obscureness of the transformed features is a persistent issue for researchers.

REFERENCES

[1] G. Camille Wermuth, B. Villoutreix, S. Grisoni, A. Olivier, and J.-P. Rocher, *Strategies in the Search for New Lead Compounds or Original Working Hypotheses*. San Diego, CA, USA: Academic, Jan. 2015, ch. 4, pp. 73–99.
[2] D. A. Winkler, "The role of quantitative structure–activity relationships (QSAR) in biomolecular discovery," *Briefings Bioinf.*, vol. 3, no. 1, pp. 73–86, Jan. 2002.
[3] Danishuddin and A. U. Khan, "Descriptors and their selection methods in QSAR analysis: Paradigm for drug design," *Drug Discovery Today*, vol. 21, no. 8, pp. 1291–1302, Aug. 2016.

[4] C. Schittenkopf, G. Deco, and W. Brauer, "Two strategies to avoid overfitting in feedforward networks," *Neural Netw.*, vol. 10, no. 3, pp. 505–516, Apr. 1997.

[5] Z. M. Hira and D. F. Gillies, "A review of feature selection and feature extraction methods applied on microarray data," *Adv. Bioinf.*, vol. 2015, 2015.

[6] L. Ladha and T. Deepa, "Feature selection methods and algorithms," *Int. J. Comput. Sci. Eng.*, vol. 3, no. 5, pp. 1787–1797, 2011.

[7] N. Mehrabi, F. Morstatter, N. Saxena, K. Lerman, and A. Galstyan, "A survey on bias and fairness in machine learning," 2019, *arXiv:1908.09635*. [Online]. Available: http://arxiv.org/abs/1908.09635

[8] L. van der Maaten and J. van den Herik, "Dimensionality reduction: A comparative review," *J. Mach. Learn. Res.*, vol. 10, nos. 66–13, p. 13, 2009.

[9] J. Ma, R. P. Sheridan, A. Liaw, G. E. Dahl, and V. Svetnik, "Deep neural nets as a method for quantitative structure–activity relationships," *J. Chem. Inf. Model.*, vol. 55, no. 2, pp. 263–274, 2015.

[10] Y. Yuan, F. Zheng, and C.-G. Zhan, "Improved prediction of blood–brain barrier permeability through machine learning with combined use of molecular property-based descriptors and fingerprints," *AAPS J.*, vol. 20, no. 3, p. 54, May 2018.

[11] R. Miao, L.-Y. Xia, H.-H. Chen, H.-H. Huang, and Y. Liang, "Improved classification of Blood-Brain-Barrier drugs using deep learning," *Sci. Rep.*, vol. 9, no. 1, pp. 1–11, Dec. 2019.

[12] H. Li, C. W. Yap, C. Y. Ung, Y. Xue, Z. W. Cao, and Y. Z. Chen, "Effect of selection of molecular descriptors on the prediction of blood–brain barrier penetrating and nonpenetrating agents by statistical learning methods," *J. Chem. Inf. Model.*, vol. 45, no. 5, pp. 1376–1384, Sep. 2005.

[13] Z. Wang, H. Yang, Z. Wu, T. Wang, W. Li, Y. Tang, and G. Liu, "In silico prediction of blood-brain barrier permeability of compounds by machine learning and resampling methods," *ChemMedChem*, vol. 13, no. 20, pp. 2189–2201, Oct. 2018.

[14] M. Storcheus, A. Rostamizadeh, and S. Kumar, "A survey of modern questions and challenges in feature extraction," in *Proc. Int. Workshop Feature Extraction, Modern Questions Challenges*, Montreal, QC, Canada, 2015, pp. 1–18.

[15] S. Khalid, T. Khalil, and S. Nasreen, "A survey of feature selection and feature extraction techniques in machine learning," in *Proc. Sci. Inf. Conf.*, Aug. 2014, pp. 372–378.

[16] Q. Meng, D. Catchpoole, D. Skillicom, and P. J. Kennedy, "Relational autoencoder for feature extraction," in *Proc. Int. Joint Conf. Neural Netw. (IJCNN)*, May 2017, pp. 341–371.

[17] M. Goodarzi, Y. V. Heyden, and S. Funar-Timofei, "Towards better understanding of feature-selection or reduction techniques for quantitative structure–activity relationship models," *TrAC Trends Anal. Chem.*, vol. 42, pp. 49–63, Jan. 2013.

[18] M. P. Sharma and R. P. Saxena, "A review on non linear dimensionality reduction techniques for face recognition," *Int. J. Recent Innov. Trends Comput. Commun.*, vol. 5, no. 7, pp. 195–200, 2017.

[19] H. Chen, O. Engkvist, Y. Wang, M. Olivecrona, and T. Blaschke, "The rise of deep learning in drug discovery," *Drug Discovery Today*, vol. 23, no. 6, pp. 1241–1250, Jun. 2018.

[20] M. Linting, J. J. Meulman, J. F. P. Groenen, and J. A. van der Koojj, "Nonlinear principal components analysis: Introduction and application," *Psychol. Methods*, vol. 12, no. 3, pp. 336–358, Sep. 2007.

[21] S. Pirhadi, F. Shiri, and J. B. Ghasemi, "Multivariate statistical analysis methods in QSAR," *RSC Adv.*, vol. 5, no. 127, pp. 104635–104665, 2015.

[22] J. Lin and D. Gunopulos, "Dimensionality reduction by random projection and latent semantic indexing," in *Proc. 3rd SIAM Int. Conf. Data Mining Text Mining Workshop*, 2003.

[23] D. Achlioptas, "Database-friendly random projections," in *Proc. 20th ACM SIGMOD-SIGACT-SIGART Symp. Princ. Database Syst. (PODS)*, 2001, pp. 274–281.

[24] W. Wang, Y. Huang, Y. Wang, and L. Wang, "Generalized autoencoder: A neural network framework for dimensionality reduction," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. Workshops*, Jun. 2014, pp. 490–497.

[25] E. Bingham and H. Mannila, "Random projection in dimensionality reduction: Applications to image and text data," in *Proc. 7th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining (KDD)*, 2001, pp. 245–250.

[26] G. Idakwo, J. Luttrell, IV, M. Chen, H. Hong, P. Gong, and C. Zhang, "A review of feature reduction methods for QSAR-based toxicity prediction," in *Advances in Computational Toxicology*. New York, NY, USA: Springer, pp. 119–139, 2019.

[27] M. Eklund, U. Norinder, S. Boyer, and L. Carlsson, "Choosing feature selection and learning algorithms in QSAR," *J. Chem. Inf. Model.*, vol. 54, no. 3, pp. 837–843, Mar. 2014.

[28] S. Liu, D. Maljovec, B. Wang, P.-T. Bremer, and V. Pascucci, "Visualizing high-dimensional data: Advances in the past decade," *IEEE Trans. Vis. Comput. Graphics*, vol. 23, no. 3, pp. 1249–1268, Mar. 2017.

[29] M. Sedlmair, T. Munzner, and M. Tory, "Empirical guidance on scatterplot and dimension reduction technique choices," *IEEE Trans. Vis. Comput. Graph.*, vol. 19, no. 12, pp. 2634–2643, Dec. 2013.

[30] U. Norinder and S. Boyer, "Binary classification of imbalanced datasets using conformal prediction," *J. Mol. Graph. Model.*, vol. 72, pp. 256–265, Mar. 2017.

[31] R. Concu and M. N. D. S. Cordeiro, "On the relevance of feature selection algorithms while developing non-linear QSARs," in *Ecotoxicological QSARs*. New York, NY, USA: Springer, 2020, pp. 177–194.

[32] M. Köppen, "The curse of dimensionality," in *Proc. 5th Online World Conf. Soft Comput. Ind. Appl. (WSC5)*, vol. 1, 2000, pp. 4–8.

[33] V. Bolón-Canedo, N. Sánchez-Maroño, and A. Alonso-Betanzos, *Feature Selection for High-Dimensional Data*. New York, NY, USA: Springer, 2015.

[34] M. P. Gonzalez, C. Teran, L. Saiz-Urra, and M. Teijeira, "Variable selection methods in QSAR: An overview," *Current Topics Med. Chem.*, vol. 8, no. 18, pp. 1606–1627, 2008.

[35] F. Grisoni, V. Consonni, and R. Todeschini, "Impact of molecular descriptors on computational models," in *Computational Chemogenomics*. New York, NY, USA: Springer, 2018, pp. 171–209.

[36] M. Hechinger, K. Leonhard, and W. Marquardt, "What is wrong with quantitative structure–property relations models based on three-dimensional descriptors?" *J. Chem. Inf. Model.*, vol. 52, no. 8, pp. 1984–1993, 2012.

[37] K. Pearson, "Principal components analysis," *Dublin Phil. Mag. J. Sci.*, vol. 6, no. 2, p. 559, 1901.

[38] R. O. Duda, P. E. Hart, and D. G. Stork, "Fisher's linear discriminant," in *Patent Classification and Scene Analysis*. Hoboken, NJ, USA: Wiley, 1973.

[39] W. B. Johnson and J. Lindenstrauss, "Extensions of Lipschitz mappings into a Hilbert space," *Contemp. Math.*, vol. 26, nos. 189–206, p. 1, 1984.

[40] C. Yoo and M. Shahlaei, "The applications of PCA in QSAR studies: A case study on CCR5 antagonists," *Chem. Biol. Drug Design*, vol. 91, no. 1, pp. 137–152, Jan. 2018.

[41] S. Negi, Y. Kumar, and V. M. Mishra, "Feature extraction and classification for EMG signals using linear discriminant analysis," in *Proc. 2nd Int. Conf. Adv. Comput., Commun., Autom. (ICACCA) (Fall)*, Sep. 2016, pp. 1–6.

[42] D.-W. Chen, R. Miao, W.-Q. Yang, Y. Liang, H.-H. Chen, L. Huang, C.-J. Deng, and N. Han, "A feature extraction method based on differential entropy and linear discriminant analysis for emotion recognition," *Sensors*, vol. 19, no. 7, p. 1631, 2019.

[43] X. Tan, L. Deng, Y. Yang, Q. Qu, and L. Wen, "Optimized regularized linear discriminant analysis for feature extraction in face recognition," *Evol. Intell.*, vol. 12, no. 1, pp. 73–82, Mar. 2019.

[44] A. M. Martinez and A. C. Kak, "PCA versus LDA," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 23, no. 2, pp. 228–233, Feb. 2001.

[45] W. Li, M. Coats, J. Zhang, and S. J. McKenna, "Comparative analysis of feature extraction methods for colorectal polyp images in optical projection tomography," in *Proc. MIUA*, 2013, pp. 67–82.

[46] J. P. Cunningham and Z. Ghahramani, "Linear dimensionality reduction: Survey, insights, and generalizations," *J. Mach. Learn. Res.*, vol. 16, pp. 2859–2900, 2015.

[47] S. T. Roweis, "Nonlinear dimensionality reduction by locally linear embedding," *Science*, vol. 290, no. 5500, pp. 2323–2326, Dec. 2000.

[48] J. B. Tenenbaum, "A global geometric framework for nonlinear dimensionality reduction," *Science*, vol. 290, no. 5500, pp. 2319–2323, Dec. 2000.

[49] D. L. Donoho and C. Grimes, "Hessian eigenmaps: Locally linear embedding techniques for high-dimensional data," *Proc. Nat. Acad. Sci. USA*, vol. 100, no. 10, pp. 5591–5596, May 2003.

[50] J. Chen and Y. Liu, "Locally linear embedding: A survey," *Artif. Intell. Rev.*, vol. 36, no. 1, pp. 29–48, Jun. 2011.

[51] F. S. Tsai, "Comparative study of dimensionality reduction techniques for data visualization," *J. Artif. Intell.*, vol. 3, no. 3, pp. 119–134, Mar. 2010.

[52] A. Akhbardeh and M. A. Jacobs, "Comparative analysis of nonlinear dimensionality reduction techniques for breast MRI segmentation," *Med. Phys.*, vol. 39, no. 4, pp. 2275–2289, Apr. 2012.

[53] P.-J. L'Heureux, J. Carreau, Y. Bengio, O. Delalleau, and S. Y. Yue, "Locally linear embedding for dimensionality reduction in QSAR," *J. Comput.-Aided Mol. Des.*, vol. 18, nos. 7–9, pp. 475–482, Jul. 2004.

[54] J. Wang, H. He, and D. V. Prokhorov, "A folded neural network autoencoder for dimensionality reduction," *Procedia Comput. Sci.*, vol. 13, pp. 120–127, 2012.
[55] Z. Liu, Z. Lai, W. Ou, K. Zhang, and R. Zheng, "Structured optimal graph based sparse feature extraction for semi-supervised learning," *Signal Process.*, vol. 170, May 2020, Art. no. 107456.
[56] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, "Learning internal representations by error propagation," La Jolla Inst. Cogn. Sci., California Univ. San Diego, San Diego, CA, USA, Tech. Rep. ICS-8506, 1985.
[57] I. Dorronsoro, A. Chana, M. I. Abasolo, A. Castro, C. Gil, M. Stud, and A. Martinez, "CODES/Neural network model: A useful tool for in silico prediction of oral absorption and blood-brain barrier permeability of structurally diverse drugs," *QSAR Combinat. Sci.*, vol. 23, no. 23, pp. 89–98, Apr. 2004.
[58] A. Guerra, J. A. Páez, and N. E. Campillo, "Artificial neural networks in ADMET modeling: Prediction of blood-brain barrier permeation," *QSAR Combinat. Sci.*, vol. 27, no. 5, pp. 586–594, May 2008.
[59] G. E. Hinton, "Reducing the dimensionality of data with neural networks," *Science*, vol. 313, no. 5786, pp. 504–507, Jul. 2006.
[60] Q. Hu, M. Feng, L. Lai, and J. Pei, "Prediction of drug-likeness using deep autoencoder neural networks," *Frontiers Genet.*, vol. 9, Nov. 2018.
[61] R. Gómez-Bombarelli, J. N. Wei, D. Duvenaud, J. M. Hernández-Lobato, B. Sánchez-Lengeling, D. Sheberla, J. Aguilera-Iparraguirre, T. D. Hirzel, R. P. Adams, and A. Aspuru-Guzik, "Automatic chemical design using a data-driven continuous representation of molecules," *ACS Central Sci.*, vol. 4, no. 2, pp. 268–276, Feb. 2018.
[62] E. Bjerrum and B. Sattarov, "Improving chemical autoencoder latent space and molecular de novo generation diversity with heteroencoders," *Biomolecules*, vol. 8, no. 4, p. 131, 2018.
[63] N. Sukumar, G. Prabhu, and P. Saha, "Applications of genetic algorithms in QSAR/QSPR modeling," in *Applications Metaheuristics in Process Engineering*, J. Valadi and P. Siarry, Eds. New York, NY, USA: Springer, 2014, pp. 315–324.
[64] L. Afzelius, C. M. Masimirembwa, A. Karlén, T. B. Andersson, and I. Zamora, "Discriminant and quantitative PLS analysis of competitive CYP2C9 inhibitors versus non-inhibitors using alignment independent GRIND descriptors," *J. Comput.-Aided Mol. Des.*, vol. 16, no. 7, pp. 443–458, 2002.
[65] K. Hasegawa, Y. Miyashita, and K. Funatsu, "GA strategy for variable selection in QSAR studies: GA-based PLS analysis of calcium channel antagonists," *J. Chem. Inf. Comput. Sci.*, vol. 37, no. 2, pp. 306–310, Mar. 1997.
[66] T.-H. Lin, H.-T. Li, and K.-C. Tsai, "Implementing the Fisher's discriminant ratio in ak-means clustering algorithm for feature selection and data set trimming," *J. Chem. Inf. Comput. Sci.*, vol. 44, no. 1, pp. 76–87, Jan. 2004.
[67] E. Bonabeau, M. Dorigo, and G. Theraulaz, "Inspiration for optimization from social insect behaviour," *Nature*, vol. 406, no. 6791, pp. 39–42, Jul. 2000.
[68] A. Tillander, "Classification models for high-dimensional data with sparsity patterns," Ph.D. dissertation, Dept. Statist., Stockholm Univ., Stockholm, Sweden, 2013.
[69] J. Cai, J. Luo, S. Wang, and S. Yang, "Feature selection in machine learning: A new perspective," *Neurocomputing*, vol. 300, pp. 70–79, Jul. 2018.
[70] K. Chrysostomou, S. Y. Chen, and X. Liu, "Combining multiple classifiers for wrapper feature selection," *Int. J. Data Mining, Model. Manage.*, vol. 1, no. 1, pp. 91–102, 2008.
[71] S. Solorio-Fernández, J. A. Carrasco-Ochoa, and J. F. Martínez-Trinidad, "A review of unsupervised feature selection methods," *Artif. Intell. Rev.*, vol. 53, no. 2, pp. 907–948, Feb. 2020.
[72] K. Q. Weinberger, F. Sha, and L. K. Saul, "Learning a kernel matrix for nonlinear dimensionality reduction," in *Proc. 21st Int. Conf. Mach. Learn. (ICML)*, 2004, p. 106.
[73] V. S. Sahithi and I. V. M. Krishna, "Performance evaluation of dimensionality reduction techniques on chris hyper spectral data for surface discrimination," *J. Geomatics*, vol. 10, no. 1, pp. 7–11, 2016.
[74] M. W. Bradbury, "The blood-brain barrier," *Experim. Physiol.*, vol. 78, no. 4, pp. 453–472, Jul. 1993.
[75] Alvascience Srl. *alvaDesc (Software for Molecular Descriptors Calculation)*. Accessed: Oct. 16, 2019. [Online]. Available: https://www.alvascience.com/alvadesc/
[76] I. Sushko, S. Novotarskyi, R. Körner, A. K. Pandey, M. Rupp, W. Teetz, S. Brandmaier, A. Abdelaziz, V. V. Prokopenko, and V. Y. Tanchuk, "Online chemical modeling environment (OCHEM): Web platform for data storage, model development and publishing of chemical information," *J. Comput.-Aided Mol. Des.*, vol. 25, no. 6, pp. 533–554, Jun. 2011.
[77] C. Steinbeck, Y. Han, S. Kuhn, O. Horlacher, E. Luttmann, and E. Willighagen, "The chemistry development kit (CDK): An open-source java library for chemo- and bioinformatics," *ChemInform*, vol. 34, no. 21, pp. 493–500, May 2003.
[78] Y. Chu and X. He, "Molegear: A java-based platform for evolutionary de nevo molecular design," *Molecules*, vol. 24, no. 7, p. 1444, 2019.
[79] M. Kubat and S. Matwin, "Addressing the curse of imbalanced training sets: One-sided selection," in *Proc. 4th Int. Conf. Mach. Learn.*, Nashville, TN, USA, vol. 97, Jul. 1997, pp. 179–186.
[80] N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer, "SMOTE: Synthetic minority over-sampling technique," *J. Artif. Intell. Res.*, vol. 16, pp. 321–357, Jun. 2002.
[81] W. Zhu, N. Zeng, and N. Wang, "Sensitivity, specificity, accuracy, associated confidence interval and ROC analysis with practical SAS implementations," in *Proc. NESUG Health Care Life Sci.*, Baltimore, MD, USA, vol. 19, 2010, p. 67.
[82] J. Akosa, "Predictive accuracy: A misleading performance measure for highly imbalanced data," in *Proc. SAS Global Forum*, 2017, pp. 2–5.
[83] T. Fawcett, "An introduction to ROC analysis," *Pattern Recognit. Lett.*, vol. 27, no. 8, pp. 861–874, Jun. 2006.
[84] C. X. Ling, J. Huang, and H. Zhang, "AUC: A statistically consistent and more discriminating measure than accuracy," in *Proc. IJCAI*, vol. 3, 2003, pp. 519–524.
[85] J. Huang and C. X. Ling, "Using AUC and accuracy in evaluating learning algorithms," *IEEE Trans. Knowl. Data Eng.*, vol. 17, no. 3, pp. 299–310, Mar. 2005.
[86] C. Cortes and M. Mohri, "AUC optimization vs. error rate minimization," in *Proc. Adv. Neural Inf. Process. Syst.*, 2004, pp. 313–320.
[87] B. L. Sturm, "Classification accuracy is not enough," *J. Intell. Inf. Syst.*, vol. 41, no. 3, pp. 371–406, Dec. 2013.
[88] I. F. Martins, L. A. Teixeira, L. Pinheiro, and O. A. Falcao, "A Bayesian approach to in silico blood-brain barrier penetration Mmdeling," *J. Chem. Inf. Model.*, vol. 52, no. 6, pp. 1686–1697, Jun. 2012.
[89] Y.-Y. Sun, M. K. Ng, and Z.-H. Zhou, "Multi-instance dimensionality reduction," in *Proc. 24th AAAI Conf. Artif. Intell.*, GA, USA, 2010.
[90] Y. LeCun, "Mnist handwritten digit database, Yann Lecun, Corinna Cortes and Chris Burges," Tech. Rep., 2013.
[91] P. Juszczak, D. Tax, and R. P. W. Duin, "Feature scaling in support vector data description," in *Proc. ASCI*. Citeseer, 2002, pp. 95–102.

SHROOQ A. ALSENAN (Member, IEEE) is currently pursuing the Ph.D. degree with the College of Computer and Information Sciences, King Saud University. She works as the Head of the National and International Collaborations at the Research Center, College of Computer and Information Sciences, Princess Nourah Bint Abdul Rahman University. She is also a member of two research groups. Her research interests include chemoinformatics, bioinformatics, NLP, and deep learning.

ISRA M. AL-TURAIKI is currently an Assistant Professor of computer science with King Saud University. Her research interests are in the areas of data mining, bioinformatics, and machine learning.

ALAAELDIN M. HAFEZ (Member, IEEE) received the Ph.D. degree from Case Western Reserve University, in 1989. He is currently a Professor of information systems with King Saud University. His current research interests are in the areas of data analytics, big data, bioinformatics, cloud computing, and artificial intelligence.