IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received November 26, 2020, accepted December 19, 2020, date of publication December 25, 2020,
date of current version January 6, 2021.

*Digital Object Identifier 10.1109/ACCESS.2020.3047375*

# Auto-KPCA: A Two-Step Hybrid Feature Extraction Technique for Quantitative Structure–Activity Relationship Modeling

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

[1]Partnership and Investment Unit, College of Computer and Information Sciences, Princess Nourah bint Abdulrahman University, Riyadh 11564, Saudi Arabia
[2]Department of Information Technology, College of Computer and Information Sciences, King Saud University, Riyadh 11451, Saudi Arabia
[3]Department of Information Systems, College of Computer and Information Sciences, King Saud University, Riyadh 11431, Saudi Arabia

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

This research was funded by the Deanship of Scientific Research at Princess Nourah bint Abdulrahman University through the Fast-track Research Funding Program.

**ABSTRACT** Quantitative structure–activity relationship (QSAR) modeling is an established approach for drug discovery, but many QSAR datasets suffer from the curse of dimensionality, a challenge that is usually addressed by using dimensionality reduction techniques such as principal component analysis (PCA). However, although linear feature extraction techniques have low computational cost and can handle linear relationships between descriptors, they cannot handle the complex structures found in QSAR data. Hybridization of feature extraction techniques is an effective approach to address the challenges of high-dimensional datasets, and combining the benefits of at least two dimensionality reduction techniques has been successful in many fields. This paper proposes Auto-KPCA, a two-step hybrid feature extraction technique that leverages (i) the fast computational capability of kernel PCA (KPCA) and (ii) the performance of a deep generalized autoencoder in handling complex data structures. Based on classification accuracy, the proposed approach is compared to other feature extraction techniques on the same benchmark dataset. The capability of Auto-KPCA is then investigated further by testing four deep-learning classification models, namely a convolutional neural network, a recurrent neural network, a feedforward deep neural network, and long short-term memory. To the best of the authors' knowledge, this study is the first to investigate hybridization of KPCA and a deep generalized autoencoder in the context of QSAR. The reported results (i) provide invaluable insights regarding the behavior of different techniques in predicting class labels and (ii) demonstrate increased classification accuracy and noticeably decreased mean square error when compared with KPCA and autoencoders.

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

## I. INTRODUCTION

With the growing popularity and prominence of drug discovery and new drug design, an abundance of biological and chemical data is now available, and ligand-based virtual screening helps to search large libraries of chemical databases to identify new compounds [1]. Quantitative structure–activity relationship (QSAR) modeling is a way to identify a relationship between a molecule and its activities. This goal is achieved by (i) finding a representation of a given molecule's structure and (ii) building a model to investigate the relationship between that representation and the desired activity or property [2].

An essential task in QSAR is to analyze the biological activities of a chemical structure based on the information encoded in the molecular descriptors. A chemical structure can be described by a myriad of molecular descriptors, also called *features*, and thus QSAR datasets are characterized as being high dimensional [3]. However, despite the many encoded descriptors, only a few are representative and related

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

to the compound's structure [4]; unnecessary descriptors result in redundancy and noise, thereby leading to inadequate classification performance. Identifying the relevant descriptors related to a compound can be accomplished by means of feature selection or extraction methods [5]. Given the multi-dimensionality of QSAR datasets, dimensionality reduction techniques are an integral part of QSAR modeling [6], and many feature selection and extraction techniques have been used to address the challenges posed by QSAR high-dimensionality [4], [7], [8].

Research is underway into dimensionality reduction for QSAR data [8]–[10]. In a previous study of how feature extraction methods affect high-dimensional QSAR data, a dataset representing the problem of blood–brain barrier (BBB) permeability [7] was investigated, and five state-of-the-art feature extraction methods were examined, all associated with promising levels of performance in other domains [8]. These methods were (i) linear principal component analysis (PCA) [11], (ii) kernel PCA (KPCA) [12], (iii) deep generalized autoencoder (dGAE) [13], (iv) Gaussian random projection (GRP) [14], and (v) sparse random projection (SRP) [15]. The study showed that dGAE could separate data points, thereby indicating their ability to handle complex datasets, and PCA was found to demonstrate the best class separation when compared to the other feature extraction techniques.

Based on the observations made and results obtained in the aforementioned empirical comparative study [8], we propose herein Auto-KPCA, a two-step hybrid feature extraction technique for QSAR modeling. The proposed technique is based on KPCA and dGAE, and we reason that leveraging the capabilities of KPCA and dGAE could highlight useful ways of handling complex QSAR data and extracting more relevant features. Expanding on the feature extraction techniques examined by Alsenan *et al.* [8], the present paper contributes to QSAR research as follows. 1) It presents a novel hybrid feature extraction technique based on dGAE and KPCA, and to the best of our knowledge the proposed technique is yet to be tested in the QSAR context. KPCA is computationally more efficient than dGAE, whereas dGAE is better at handling complex data structures. We investigate the performance of the proposed hybrid technique in comparison with KPCA or dGAE alone and based on measuring the mean square error (MSE) to analyze the amount of lost information following feature extraction. 2) It analyzes the technique's ability to separate class labels by visualizing the data points using scatterplots. 3) It develops four deep learning (DL) classifiers and compares their performance before and after employing Auto-KPCA. 4) It compares the performance of the proposed hybrid technique with those of state-of-the-art techniques and similar approaches in the literature. The performance is assessed based on the classifier accuracy measures indicated by Idakwo *et al.* [16] and Sahithi *et al.* [17]. 5) It compares the running time of the Auto-KPCA hybrid technique with those of KPCA and autoencoders (AEs) separately.

The rest of this paper is organized as follows. In Section II, we review the literature on feature extraction techniques in QSAR. In Section III, we outline the present research methodology, including the experimental setup, the preprocessing, the proposed feature extraction technique, and the classification models. We present the experimental results in Section IV and our conclusions in Section V.

## II. LITERATURE REVIEW

High dimensionality is a major problem when building a classification model because it can result in feature noise, redundancy, and computational complexity. QSAR datasets can benefit from dimensionality reduction techniques to solve these problems, and the high dimensionality of QSAR datasets has encouraged the application of feature selection and extraction methods to handle this type of dataset.

In feature selection, a subset of features is kept while less-relevant features are discarded. The feature subset is chosen such that the essence of the original representation is retained. There are many types of feature selection methods, including filters, wrappers, and embedded/hybrid methods [4], [18]. Li *et al.* [19] used a feature selection method known as recursive feature elimination (RFE) to extract the most effective features in a QSAR dataset; they reported that the features selected by RFE contributed to the best-performing classification model. Castillo-Garit *et al.* [20] used wrapper feature selection for dimensionality reduction, and Brito-Sanchez *et al.* [21] used a feature selection method known as forward stepwise to select compounds. Danishuddin *et al.* [4] reviewed QSAR feature selection methods; they showed that using feature selection methods alone produces enhanced results, but they recommended including a feature extraction step to handle efficiently the complexity and high dimensionality of QSAR data [3].

Feature extraction works by transforming the original feature space into a new lower-dimensional one. The initial features undergo various operations to produce new features, thereby meaning that the new features cannot be associated easily with their original components. Many state-of-the-art feature extraction techniques have been used to deal with high-dimensional QSAR datasets, such as genetic algorithms (GAs) and partial least squares regression [22]–[24], ant colony optimization [25], $k$-means clustering [26], and PCA [11].

PCA has become widely popular in the context of QSAR dimensionality reduction [11]. Research suggests that PCA is less sensitive to noise when compared to other well-known feature extraction methods such as Isomap, locally linear embedding (LLE), and Hessian LLE (HLLE) [27], [28]. PCA has also been shown to outperform linear discriminant analysis when dealing with complex data structures [16], [29], [30].

With the growing success of DL techniques, AEs have been used for dimensionality reduction [31]. AEs can deal efficiently with nonlinear data [32]–[34] and can learn two-way mappings between high- and low-dimensional spaces [35].

This is considered highly advantageous compared to other techniques such as LLE, which is limited to one-way mapping and cannot extract features gradually, thereby meaning that the relationship between samples is not preserved properly because all features are extracted at once [31].

Dorronsoro *et al.* [36] and Guerra *et al.* [37] used an unsupervised artificial neural network (ANN) to extract descriptors for a classification model. Wang *et al.* [7] subjected their dataset to a combination of selection and extraction methods and compared the performance of the developed predictive model using different subsets of feature selection techniques; they selected variance threshold (VT), univariate feature selection (UFE), RFE, Pearson correlation coefficient, and PCA. Of the fingerprints used in their study, they reported that the Molecular Access System ones outperformed the rest. As the feature selection and extraction methods, they chose VT, UFE, and RFE to reduce the dimensionality to only 72 descriptors.

Recently, Alsenan *et al.* [8] tested five state-of-the-art feature extraction methods in the context of QSAR, namely (i) linear PCA [11], (ii) KPCA [12], (iii) dGAE [13], (iv) GRP [14], and (v) SRP [15].

That study shed light on the effectiveness of feature extraction techniques in QSAR to predict BBB permeability. Through visualization after projecting the data in a lower-dimensional space, Alsenan *et al.* concluded that dGAE demonstrated the best separation of instances, indicating that dGAE has great potential for extracting features from complex data. They also showed that PCA exhibited the best class separation and could identify the two classes better than could random projection or dGAE.

To address the challenges of high-dimensional datasets, much effort has been made to combine the benefits of dimensionality reduction techniques [16], [38]. Hybridization of dimensionality reduction techniques is usually done in at least two steps, with different methods applied usually in a pipeline. Linear discriminant analysis combined with a GA was used for gene selection to increase the prediction accuracy of many microarray gene expression datasets [39]. A GA was also combined with a support vector machine (GA-SVM) [40] and simulated annealing [41]. Susmi *et al.* [42] subjected a gene-expression leukemia dataset to PCA combined with canonical correlation analysis, and the hybridization was an improvement compared to using a single dimensionality-reduction technique. In a recent survey, Almugren and Alshamlan [43] highlighted many hybrid feature selection methods based on bio-inspired evolutionary methods. They applied the reviewed hybrid methods in the context of gene selection and cancer classification, and the hybrid methods included GA, ant colony optimization, bat algorithm, artificial bee colony, particle swarm optimization, and grasshopper optimization.

Encouraged by (i) the literature showing the benefits of hybridization approaches and (ii) the observations from the recent comparative study of feature extraction techniques [8], we investigate herein the proposed two-step hybrid feature extraction technique.

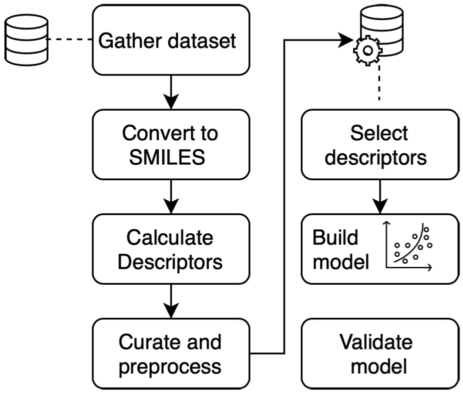

**FIGURE 1.** Quantitative structure–activity relationship (QSAR) modeling.

## III. METHODOLOGY

In this section, we outline the present research methodology. Winkler *et al.* [44] reviewed how to represent the relationship between a chemical structure and its properties and activities. The process involves gathering a dataset of chemical compounds, converting those compounds to SMILES (Simplified Molecular Input Line Entry Specification), calculating the molecular descriptors, curating and preprocessing the data, applying dimensionality reduction to select the relevant descriptors, and finally developing and validating the classification model. Fig. 1 shows the steps involved in QSAR modeling.

In this research, we propose Auto-KPCA, a novel two-step hybrid dimensionality reduction technique. According to Idakwo *et al.* [16], a common method for assessing the performance of a feature extraction technique is to compare classifier performance before and after feature extraction. Herein, we follow that approach by examining and comparing the performance of DL classification models before and after applying Auto-KPCA. The classification models are based on four DL algorithms, namely (i) a feedforward deep neural network (FFDNN), (ii) a convolutional neural network (CNN), (iii) a recurrent neural network (RNN), and (iv) long short-term memory (LSTM). Herein, we examine the proposed feature extraction technique by modeling a QSAR problem known as BBB permeability [45], and we summarize the present methodology as the following steps:

1) obtain a dataset of compounds encoded in unique SMILES representation with known logBB values;
2) calculate descriptors (features) representing the chemical and biological properties of compounds to generate a high-dimensional dataset;
3) curate and preprocess the high-dimensional dataset;
4) save the high-dimensional data as BBB-Dataset A;
5) apply Auto-KPCA to reduce the high dimensionality of BBB-Dataset A; the new dataset with the extracted features is saved as BBB-Dataset B;

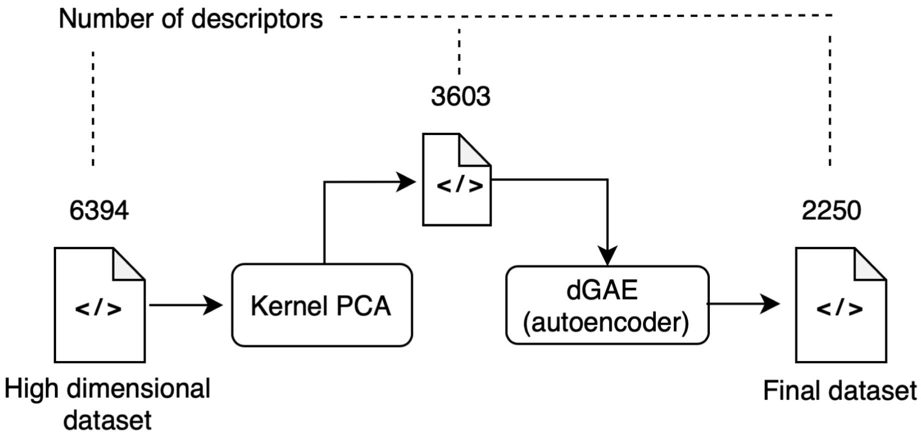

**FIGURE 2.** Auto-KPCA architecture.

6) develop and train the four DL classifiers to predict the logBB values;
7) test the DL classifiers on BBB-Datasets A and B;
8) compare the classifier performance with each dataset to obtain the capabilities of the proposed technique;
9) compare the classifier performance on BBB-Dataset B against other approaches in the literature.

The present methodology gives insights into the capabilities of the proposed feature extraction method in extracting valuable information. Because the proposed hybrid technique is based on two widely used feature extraction methods, namely KPCA and dGAE, it is important to assess the ability of each technique in retaining important features. Therefore, after the classification task, we calculate the MSE of the hybrid technique in comparison with KPCA and dGAE individually.

### A. EXPERIMENTAL SETUP

The high-dimensional dataset is for the well-known QSAR problem of BBB permeability. Compounds are classified as having either high permeability (BBB+) or low permeability (BBB−). To measure a compound's permeability to the central nervous system, a splitting threshold is chosen to separate the two class labels; we follow previous studies and split the compounds as $\log BB \geq -1$ for class BBB+ and $\log BB < -1$ for BBB− [19]. The benchmark dataset was acquired from Wang *et al.* [7] as the largest BBB dataset; it contains 1803 BBB+ compounds and 547 BBB− compounds (2350 in total).

The dataset is a list of compound instances encoded as SMILES along with the class labels. To generate the high-dimensional dataset, molecular descriptors (features) are calculated using chemical tools that are specialized for this task. We use the software tools AlvaDes [46] and OCHEM (Online Chemical Modeling Environment) [47] to calculate 1D, 2D, 3D and fingerprints for each compound. For the 2350 compounds, 6394 descriptors were calculated for this experiment, forming a high-dimensional QSAR dataset.

### B. PREPROCESSING

Machine learning and DL require several preprocessing steps to ensure data efficiency, with data cleaning, integrating, and transformation being important steps before any classification task. After generating the descriptors, the 1D, 2D, 3D descriptors and fingerprints were integrated into one dataset. To clean the dataset after calculating the descriptors, we  searched the dataset for records with no calculated descriptors or missing values. Eight compounds were found with no calculated descriptors and were dropped. As for records with some null values, we performed a "replace by the mean" operation by calculating the mean value for the descriptor in the entire dataset and substituting that for the null values. The descriptors had various value ranges, so we performed MinMax scaling to normalize the range of descriptors values in the dataset; this step transformed the values to be between 0 and 1.

### C. AUTO-KPCA

Auto-KPCA is a novel two-step hybrid dimensionality reduction technique that can be used to transform a high-dimensional QSAR dataset (BBB permeability in the present study) from a high-dimensional space to a low-dimensional space. The proposed approach is based on KPCA and dGAE, with the motivation for combining the two nonlinear techniques stemming from the desire to benefit from the fast computational capability of KPCA and the performance of dGAE in handling complex data structures. This architecture reduces the computational complexity of AEs, therefore the complete high-dimensional data are passed through KPCA first and then input into dGAE.

Fig. 2 shows the data flow of descriptors from a high-dimensional space to a low-dimensional space. The complete set of descriptors comprised 6394 descriptors and fingerprints, each of which was used as an input for KPCA. The objective of first passing the complete high-dimensional data to KPCA was to transform the data to a lower-dimensional space while achieving minimum information loss and faster computational processing compared with dGAE.

With KPCA, the kernel function for KPCA was set to a polynomial. The number of components returned by KPCA can be specified, but for our model we allowed KPCA to return all non-zero values in the space with reduced dimension. This was accomplished by not passing any value to KPCA in terms of the number of components (the parameter n_components), thereby allowing the model to retain all invaluable information. After passing the complete high-dimensional dataset to KPCA, 3603 descriptors were returned.

The retained descriptor set was then passed to dGAE, which was constructed using six encoder layers with a 30% dropout rate and five decoder layers with a 20% dropout rate. The first encoder layer of the model was the input layer, while the final encoder layer was the output to obtain the space with reduced dimension. The number of hidden units in the first encoder layer amounted to 3603, which corresponded to the number of features obtained from KPCA. The hyperparameter tuning of the model was set experimentally to Adam as an optimizer with a learning rate of 0.01, and the activation function was ReLU with an epoch size of 50 and a batch size of 64. Although the AE was trained using both the encoder and the decoder, the reduced dimensionality was obtained from the smallest hidden layer in the encoder, known as the "bottleneck." This layer transformed the input to the "latent space," which is the lowest space level in the architecture. dGAE extracted 2250 features in total.

The pseudocode of Auto-KPCA in which the latent space is computed is shown in Algorithm 1. The input values to Auto-KPCA are represented in lines 2–4 and 18–20, and the output values are represented in lines 6 and 23. The input of Auto-KPCA is a high-dimensional dataset $S$, which is processed in two steps. In step 1, the eigenvectors, eigenvalues, and principal components are computed by KPCA. The "for" loop in lines 11–14 represents the process of obtaining the final matrix and projecting the data to the low-dimensional space. Line 16 represents the output $Y$, which is a lower-dimensional dataset. In step 2, the output from KPCA is input to dGAE. We define $l$ hidden layers and $h$ neurons. The "for" loop in lines 24–29 trains the network and updates the weights to minimize the error. For all hidden layers, the activation function is computed and the output $\hat{Y}$ is constructed. The loss is calculated and backpropagated to fine-tune the weights. This process is repeated until the minimum error is reached. The final reduced dataset is produced by computing the latent space of the encoder.

**Algorithm 1** Two-Step Auto-KPCA

```
    Step 1: Kernel PCA (KPCA)
 1: Input:
 2: $S$: high-dimensional dataset
 3: $f$: kernel choice
 4: $d$: number of principal components
 5: Output:
 6: $Y$: low-dimensional dataset

 7: $K$ = compute kernel matrix $(S, f)$
 8: $K^{\sim}$ = normalize kernel matrix $(K)$
 9: $U$ = TopEigenVectors $(K^{\sim}, d)$
10: $\lambda$ = TopEigenValues $(K^{\sim}, d)$
11: for $i = 1$ to $m$ do
12:     for $r = 1$ to $d$ do
13:         $Y_{r,i} = \sum_{t=1}^{M} \frac{1}{\sqrt{\lambda_r}} \left(\mathrm{K}(x_i, x_t)\, U_{\mathrm{t,r}}\right)$
14:     end for
15: end for
16: Return $Y$

    Step 2: Deep generalized autoencoder (dGAE)
17: Input:
18: $Y$: lower-dimensional dataset
19: $h$: hidden units
20: $l$: hidden layers
21: Output:
22: $\hat{Y}$: decoder output
23: $Q$: latent space from encoder

24: for all $l$ do
25:     $a_i$ = hidden activation function
26:     $\hat{Y}$: = reconstruct output from hidden activation $(a_i)$
27:     $e$ = compute error gradient $(\hat{Y})$
28:     Backpropagate to update weights $(e)$
29: end for
30: $Q$ = compute latent space using encoder bottleneck
31: Return $Q$
```

## D. CLASSIFICATION MODELS

Four DL classification models were developed to assess how Auto-KPCA affects classifier performance. The main hypothesis is that Auto-KPCA can extract useful features that enhance classifier prediction. The hyperparameters for the DL models were set experimentally until the best model was realized. The activation function for the DL models was ReLU, with the Adam optimizer and a learning rate of 0.01. In each model, two batch optimization layers were used to normalize the output of each layer before passing it to the next layer. The number of epochs for this experiment was set to 100 and the batch size to 200. The models were validated with 10-fold validation, with the dataset split into 10 subsets. In each validation iteration, one subset was retained for testing while the rest were used for training. This process was repeated until each subset had been used once for testing. To ensure that no data leaked to the classifier and affected its performance or caused overfitting, each iteration was concluded with a clear_session function. Because BBB permeability is a binary classification problem, we used a sigmoid in the final output layer to make the class prediction between BBB+ and BBB−. The details of each DL model are given below.

1) **FFDNN:** The FFDNN model had three hidden layers (also called dense layers). The number of neurons in each hidden layer was determined by trial and error in a range between the numbers of neurons in the input and output layers [48].
2) **CNN:** The CNN model was developed by converting the dataset from 2D to 3D shape. Feature maps were calculated in each convolutional layer using filters. Four convolutional layers were constructed with a batch normalization layer between layers. The number of neurons for each hidden layer was 1024, 512, 512, 256, and 1.
3) **RNN:** The RNN model is known for solving sequential problems as well as problems with fixed input vectors like the problem in hand. RNN span (t) time back to predict the current time (t) allowing it to store previous representations. Four RNN layers were formed with the number of neurons set to 1024, 512, 512, and 256. We set the parameter `Return_sequence` to "true" to preserve the previous output.
4) **LSTM:** The LSTM model is proposed for the first time to solve the problem of BBB permeability. Although it contains infinite memory, RNN normally loops back 5–10 time steps. LSTM was proposed by Hochreiter *et al.* [49] as a solution to the short memory of RNN. To develop the LSTM model, we constructed six LSTM layers and two batch normalization layers. Similarly to the RNN model, `Return_sequence` was set to "true."

## IV. EXPERIMENTAL RESULTS

In this section, we present experimental results for the proposed two-step hybrid dimensionality reduction technique Auto-KPCA and compare them with those for the five feature extraction techniques investigated previously by Alsenan *et al.* [8]. We also discuss comparisons with other studies from the literature. First, we describe and explain the evaluation metrics used to compare the classification models, then we present the results of the four DL classifiers before and after applying Auto-KPCA [16], [17]. The MSE of Auto-KPCA is calculated and compared with that of KPCA and dGAE individually, and we use scatterplots to visualize how the proposed technique affects the shape of the dataset.

The original high-dimensional dataset comprised 6394 descriptors, which were transformed into 3603 extracted features using KPCA. In turn, the new set of descriptors was input to dGAE, where the reduced dimensions of 2250 features were extracted from the latent space in the AE (i.e., the "bottleneck"). The result was a low-dimensional dataset that was subsequently input into each of the DL models.

### A. EVALUATION MEASURES

The proposed Auto-KPCA and the other feature extraction techniques are assessed on the same benchmark data based on the classification accuracy in predicting class labels. The following five performance measures are used to assess the effectiveness of each model when given real data.

Accuracy is the percentage of compounds that are classified correctly, and it is calculated as

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN}. \tag{1}$$

Specificity is the percentage of BBB− compounds classified correctly out of the total number of BBB−, and it is calculated as

$$Specificity = \frac{TN}{TN + FP}. \tag{2}$$

Sensitivity is the percentage of BBB+ compounds classified correctly out of the total number of BBB+, and it is calculated as

$$Sensitivity = \frac{TP}{TP + FN}. \tag{3}$$

Measuring the correlation between actual and predicted labels in binary classification, the Matthews correlation coefficient (MCC) is commonly used in QSAR modeling, especially in imbalanced binary classification. It is calculated as

$$MCC = \frac{(TP \times TN)\text{-}(FP \times FN)}{\sqrt{(FP + TN)(FP + TP)(FN + TN)(FN + TP)}}. \tag{4}$$

In (1)–(4), *TP* is the true-positive rate of compounds classified correctly as BBB+, *TN* is the true-negative rate of compounds classified correctly as BBB−, *FP* is the false-positive rate of compounds classified mistakenly as BBB+, and *FN* is the false-negative rate of compounds classified mistakenly as BBB−.

Receiver operating characteristic (ROC) graphs are important for visualizing classifier performance and comparing different algorithms, showing *TP* in comparison to *FP*. The area under the ROC curve (AUC) is an important measure for assessing the performance of binary classification models [50], showing the classifier's ability to separate compounds classified as BBB+ or BBB−.

### B. RESULTS

To assess the performance of Auto-KPCA, we conduct four types of evaluation: 1) the hybrid technique is compared to the individual KPCA and dGAE techniques based on the classification accuracy of the FFDNN model; 2) the dataset is visualized in the low-dimensional space via visual encoding with scatterplots; 3) the proposed technique is tested via four DL classification models, namely FFDNN, CNN, RNN, and LSTM; to evaluate how the proposed feature extraction method affects high-dimensional QSAR data, the result of each classifier is considered before and after applying Auto-KPCA, and the best classification models with Auto-KPCA are compared with the literature based on the same benchmark dataset; 4) the running times are presented.

**TABLE 1.** Performance of dimensionality reduction techniques (ACC = overall accuracy, Sens = sensitivity, Spec = specificity, AUC = area under curve, MCC = Matthews correlation coefficient, RP = random projection, FFDNN = feedforward deep neural network, DR = dimensionality reduction).

| DR technique | Training set | | | Test set | | | | |
|---|---|---|---|---|---|---|---|---|
| | ACC [%] | Sens [%] | Spec [%] | ACC [%] | Sens [%] | Spec [%] | AUC [%] | MCC [%] |
| Auto-KPCA | 99.68 | 99.58 | 99.79 | 98.19 | 98.87 | 97.54 | 99.53 | 96.40 |
| dGAE | 97.19 | 96.90 | 97.48 | 93.67 | 94.01 | 92.72 | 97.85 | 86.71 |
| KPCA | 99.93 | 99.86 | 100 | 95.84 | 94.13 | 97.56 | 98.88 | 91.72 |

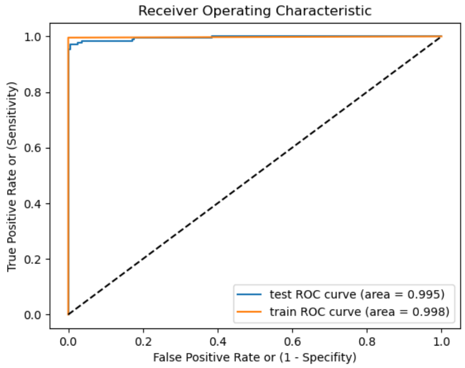

**FIGURE 3.** ROC of FFDNN classifier with Auto-KPCA.

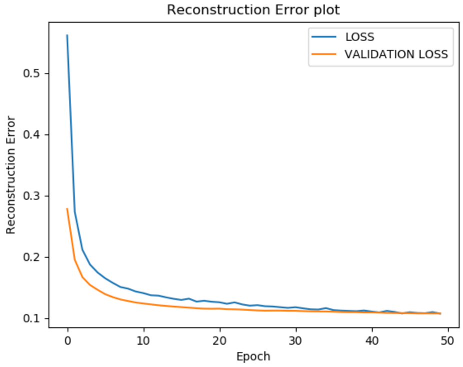

**FIGURE 4.** Classifier reconstruction error with Auto-KPCA.

### 1) AUTO-KPCA VERSUS dGAE AND KPCA

Table 1 lists the results for the proposed hybrid Auto-KPCA in comparison with those for dGAE and KPCA individually with the FFDNN model. When the FFDNN model was fitted for training, Auto-KPCA and KPCA demonstrated the best learning, in comparison with dGAE. In the testing set, Auto-KPCA outperformed the other techniques in terms of overall accuracy, sensitivity, AUC, and MCC. Auto-KPCA achieved an overall accuracy of 98.19% compared to 93.67% and 95.84% for dGAE and KPCA, respectively. Auto-KPCA achieved the best prediction of BBB-penetrating compounds by scoring 98.87% on the sensitivity measure.

Auto-KPCA achieved a specificity score of 97.54%, which is somewhat consistent with that obtained using KPCA. The AUC score for Auto-KPCA was 99.53%, which was the most accurate compared to the other techniques.

Fig. 3 shows the ROC graph of the FFDNN model using Auto-KPCA. The ROC graph illustrates $TP$ in comparison with $FP$, and the plotted data are toward the top left corner, thereby showing that the classifier can distinguish between sensitivity and specificity.

MCC is a classification measure that delivers a high score only if the results are consistently satisfactory with respect to all confusion-matrix scores (i.e., $TP$, $FN$, $TN$, and $FP$), thereby making it a highly reliable measure for classification tasks [51], [52]. In this experiment, the proposed technique yielded the highest MCC score compared to the other examined techniques. The best single technique scored 91.72% in the MCC measure, whereas the proposed Auto-KPCA achieved an MCC score of 96.40%.

Overall, the proposed two-step hybrid dimensionality reduction technique outperformed its single components in terms of improving the classifier's prediction ability across various accuracy measures.

The primary goal of training is to allow the classifier to learn data well enough to generalize to previously unseen data. This can be measured by evaluating the reconstruction error, which shows the classification loss during training and validation with respect to multiple epochs. When the reconstruction error is minimized between training and validation, it reflects the classifier's performance consistency on new data. The reconstruction error of the classifier with Auto-KPCA is shown in Fig. 4, where the training and validation lines converge. This reflects consistency in terms of accuracy, as well as overall success in terms of classifier training.

Comparing the accuracy measures of the classifier with the dimensionality reduction technique provides insights into how the proposed technique affects the classifier's prediction [16], showing its ability to extract important features that affect the classifier's learning process. Another way to measure the performance of Auto-KPCA is to use MSE, which for KPCA is calculated using the inverse-transform function. KPCA was applied first to reduce dimensionality, then the inverse-transform function was used to obtain the data loss between the actual and inverse-transformed data. In dGAE, training is performed based on the loss function; therefore, in this experiment we used the reconstruction loss calculated with MSE to find the error [31]. The error was calculated by comparing the predicted and actually obtained values during training. Table 2 presents the MSE score for each technique.

**TABLE 2. Mean squared error of Auto-KPCA compared with Kernel PCA and dGAE.**

| DR Technique | Mean squared error (MSE) |
|---|---|
| dGAE | 0.030155909674066887 |
| Kernel PCA | 5.331339499693642e-28 |
| Auto-KPCA | 9.932557459779348e-08 |

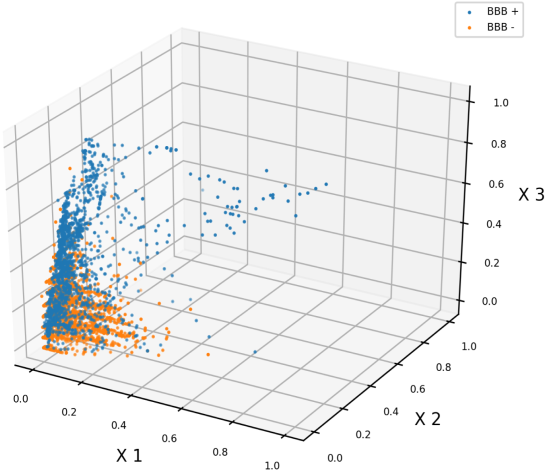

**FIGURE 5. Scatterplot visualization of data prior to diemsnioanlity reduction.**

For Auto-KPCA, the dataset was first passed to KPCA for training through the fit function, and the transformed dimensions were passed to dGAE for further feature reduction. When the lowest dimensions were reached with the AE, the output from the predict function was calculated using the decoder to retain the initial dimensions. The initial dimensions were passed back to KPCA through the inverse function to obtain the complete dataset projected back to the full dimension. Finally, the datasets were compared to obtain the MSE values.

To interpret MSE, we sought the technique with the lowest MSE. Table 2 presents the MSE scores obtained with KPCA, dGAE, and Auto-KPCA. Auto-KPCA obtained a lower MSE score compared to dGAE, which indicates that passing the high-dimensional data to KPCA first resulted in a decreased reconstruction loss with Auto-KPCA. Therefore, Auto-KPCA can extract meaningful features with minimal information loss.

### 2) DATA VISUALIZATION

To visualize the data transformation before and after applying Auto-KPCA, we use visual encoding with scatterplots. Fig. 5 shows the dataset before applying Auto-KPCA, where a positive correlation is noticeable (i.e., the data points move together in one direction). Fig. 6 visualizes the distribution of the data points after applying Auto-KPCA, and we note the following: (i) the positive and negative classes are separated noticeably across the graph axes; (ii) the level of overlap among data points is considerably lower than that in Fig. 6, thereby indicating that the proposed nonlinear technique can separate instances [53]; (iii) there is no indication of a positive correlation among data points.

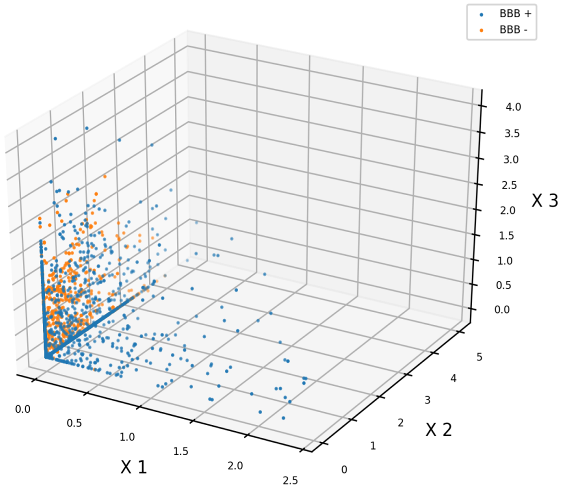

**FIGURE 6. Scatterplot visualization of Auto-KPCA.**

### 3) DEEP LEARNING MODELS RESULTS

Here, we show the performance results for the four DL models before and after applying Auto-KPCA. Table 3 summarizes the experimental results for Auto-KPCA, showing that the overall accuracy of the FFDNN model increased from 95.11% (before applying Auto-KPCA) to 98.19%. The sensitivity, AUC, and MCC values also improved to 98.78%, 99.53%, and 96.40%, respectively.

The CNN model exhibited a moderate improvement in terms of overall accuracy and sensitivity. However, the RNN model's overall accuracy increased from 96.53% to 97.22%. Meanwhile, the LSTM model showed the greatest performance improvement; in particular, before applying Auto-KPCA, the accuracy was 90.85%, which increased subsequently to 96.67%. This indicates that LSTM was sensitive to noise when dealing with the high-dimensional data.

Overall, the four models were associated with noticeable improvements in overall accuracy using Auto-KPCA when tested on an external unseen dataset. FFDNN and CNN achieved a perfect score, while the overall accuracy of RNN and LSTM improved from 90% to 95% and from 80% to 85%, respectively.

To compare our proposed models properly with other previous studies, it is important to find ones conducted on similar datasets. Table 4 lists the performance of Auto-KPCA in comparison with other feature extraction techniques using similar datasets and under similar environments [8], namely linear PCA [11], GRP [14], and SRP [15]. To investigate other classification models from the literature, we chose the studies by Wang *et al.* [7] and Yuan *et al.* [45] based on the following criteria: (i) conducted on a similar BBB permeability dataset

**TABLE 3. Performance evaluation of proposed deep learning (DL) models (ACC = overall accuracy, Sens = sensitivity, Spec = specificity, AUC = area under curve, MCC = Matthews correlation coefficient, DL = deep learning, DR = dimensionality reduction, Ext = external dataset accuracy).**

| Classification model | | Training | | | Testing | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| DR | DL algorithm | ACC [%] | Sens [%] | Spec [%] | ACC [%] | Sens [%] | Spec [%] | AUC [%] | MCC [%] | ACC$_{ext}$ [%] |
| No DR | FFDNN | 99.86 | 99.72 | 100 | 95.11 | 92.15 | 98.11 | 98.38 | 90.40 | 95 |
| | CNN | 99.55 | 99.36 | 99.75 | 96.81 | 95.00 | 98.11 | 98.01 | 93.14 | 95 |
| | RNN | 99.61 | 99.31 | 99.90 | 96.53 | 94.91 | 98.09 | 98.6 | 93.14 | 90 |
| | LSTM | 95.90 | 99.87 | 91.91 | 90.85 | 82.88 | 99.42 | 96.50 | 83.02 | 80 |
| With Auto-KPCA | FFDNN | 99.68 | 99.58 | 99.79 | 98.19 | 98.87 | 97.54 | 99.53 | 96.40 | 100 |
| | CNN | 99.87 | 99.88 | 99.86 | 96.91 | 96.10 | 96.84 | 98.6 | 92.85 | 100 |
| | RNN | 100 | 100 | 100 | 97.22 | 98.33 | 96.13 | 99.64 | 94.48 | 95 |
| | LSTM | 98.54 | 99.26 | 97.48 | 96.67 | 95.05 | 98.32 | 98.34 | 93.40 | 85 |

**TABLE 4. Best-performing DL models in comparison with other studies (ACC = overall accuracy, Sens = sensitivity, Spec = specificity, AUC = area under curve, MCC = Matthews correlation coefficient, DL = deep learning, DR = dimensionality reduction, MLP = multilayer perceptron neural network).**

| DR method | Dataset | Model | ACC [%] | Sens [%] | Spec [%] | AUC [%] | MCC [%] |
|---|---|---|---|---|---|---|---|
| **Auto-KPCA** | 2350 | FFDNN | **98.19** | 98.87 | **97.58** | 99.53 | **96.40** |
| **Auto-KPCA** | 2350 | RNN | 97.22 | 98.33 | 96.13 | **99.64** | 94.48 |
| VT+RFE+UFE [7] | 2350 | SVM | 91.0 | 93.3 | 83.8 | 94.0 | - |
| VT+RFE+UFE [7] | 2350+<br>92 (BBB+) | MLP | 96.6 | **99.0** | 83.3 | 91.0 | - |
| Manual subsets [45] | 1990 | SVM | 95.7 | 96.2 | 94.4 | | 0.894 |
| KPCA [8] | 2350 | FFDNN | 95.84 | 94.13 | 97.56 | 98.88 | 91.72 |
| PCA [8] | 2350 | FFDNN | 96.23 | 95.32 | 96.92 | 96.76 | 91.11 |
| dGAE [8] | 2350 | FFDNN | 93.67 | 94.01 | 92.72 | 97.85 | 86.71 |

based on passive diffusion; (ii) encompassed a larger BBB dataset; (iii) published in the past five years; (iv) achieved the highest accuracy in the literature on BBB permeability.

Choosing a larger dataset is essential for undertaking this study. Yuan *et al.* [45] noted that a smaller dataset leads to higher accuracy, which results from the classifier being specific to certain compounds and unable to generalize to a larger dataset. This is also partially the reason for choosing recent studies, given that recent ones could gather larger datasets as a result of ligand-based virtual screening to find new compounds. Finally, it is important to compare the proposed model against the state of the art from the literature.

Table 4 demonstrates previous efforts on the same benchmark data, including those of Wang *et al.* [7] and Yuan *et al.* [45]. Yuan *et al.* [45] shared a large dataset in which different subsets of descriptors and fingerprints were examined. Wang *et al.* [7] performed experiments using an SVM and a DL consensus model based on two feature selection methods, namely VT and RFE; their SVM model achieved an overall accuracy, sensitivity, and specificity of 91.0%, 93.3%, and 83.8%, respectively. Wang *et al.* [7] proposed a consensus model with a multilayer perceptron neural network using the same benchmark data, with an additional 92 compounds classified as BBB+ for validation. As shown in Table 4, their consensus model achieved an

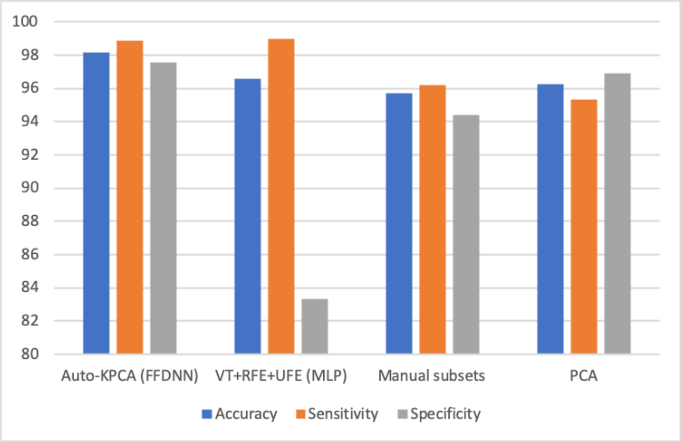

**FIGURE 7.** Performance trade-off between accuracy, sensitivity and specificity.

**TABLE 5.** Running times.

| Model | Running time for DR [s] | Running time for classification [s] |
|---|---|---|
| FFDNN alone | - | 83.24 |
| FFDNN with KPCA | 10.90 | 25.20 |
| FFDNN with dGAE | 155.41 | 178.13 |
| FFDNN with Auto-KPCA | 94.46 | 113.45 |

overall accuracy of 96.6%, a sensitivity of 99.0%, and a specificity of 83.3%. The proposed FFDNN, CNN, and RNN models outperformed Wang *et al.* [7] across all accuracy measures.

The SVM model due to Yuan *et al.* [45] achieved an overall accuracy of 95.7%, a sensitivity of 96.2%, a specificity of 94.4%, and an MCC of 89.4%. Although the overall accuracy was lower than that achieved by Wang *et al.* [7], the model due to Yuan *et al.* [45] showed greater consistency in the prediction of both class labels.

The best DL models with Auto-KPCA outperformed that due to Yuan *et al.* [45] in overall accuracy, sensitivity, and specificity. This improvement shows that the proposed feature extraction technique can select more effective features than those selected manually. In particular, the MCC of the proposed FFDNN and RNN models surpassed that of Yuan *et al.* [45] by 0.894%, which is a substantial margin. Auto-KPCA distinctly outperformed linear PCA, GRP, and SRP in overall accuracy, sensitivity, AUC, and MCC. Also, as mentioned in Section IV-B1, Auto-KPCA outperformed its individual components, and it outperformed linear PCA in all accuracy measures.

#### 4) RUNNING TIME

Table 5 summarizes the running time for each model during compilation in terms of performing dimensionality reduction, as well as the classification task. The running time for dGAE for a complete split was 155.41 s, and the complete classification task took 178.13 s.

When Auto-KPCA was applied, KPCA took 9.7 s to reduce the dimensions from 6394 to 3603. The transformed set of features was, in turn, input into dGAE for further reduction, where it took only 84.717 s to complete the dimensionality reduction task, and a total of 178.13 s per split for the classification. This time is less than that taken by dGAE, given that the complete running time for the classification task per split took only 113.45 s with Auto-KPCA.

## V. CONCLUSION

The present research constitutes a new approach to feature extraction techniques to reduce high dimensionality. Best QSAR practices were followed to compile the classification models based on the largest publicly available BBB permeability dataset.

This study initiated a line of research to address QSAR high-dimensionality problems by means of feature extraction methods. Considering the problem of encoding full sets of descriptors with dGAE, a novel two-step hybrid feature extraction technique was proposed named Auto-KPCA and based on dGAE and KPCA. We compared the proposed technique with renowned techniques that have achieved promising results using the same dataset. To investigate the capabilities of Auto-KPCA, four DL models were developed, and the accuracy of each model increased significantly after applying Auto-KPCA to the high-dimensional dataset.

The investigated feature extraction techniques represent merely a fraction of the techniques that have demonstrated adequate capability in dimensionality reduction. Therefore, future work should involve further evaluation of available feature extraction techniques to solve QSAR modeling problems. In addition, Auto-KPCA led to an enhancement in classifier performance, which indicates that the proposed technique can extract useful information from high-dimensional data. Therefore, it would be worthwhile to investigate the capabilities of Auto-KPCA in the context of other domains and research problems.

## REFERENCES

[1] A.-J. Banegas-Luna, J. P. Cerón-Carrasco, and H. Pérez-Sánchez, "A review of ligand-based virtual screening Web tools and screening algorithms in large molecular databases in the age of big data," *Future Medicinal Chem.*, vol. 10, no. 22, pp. 2641–2658, Nov. 2018.

[2] J. Gasteiger, "Of molecules and humans," *J. Medicinal Chem.*, vol. 49, no. 22, pp. 6429–6434, Nov. 2006.

[3] 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.

[4] 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.

[5] D. Storcheus, A. Rostamizadeh, and S. Kumar, "A survey of modern questions and challenges in feature extraction," in *Proc. NIPS Workshop Feature Extraction*, Montreal, QC, Canada, 2015, pp. 1–18.

[6] H. A. Gaspar, P. Sidorov, D. Horvath, I. I. Baskin, G. Marcou, and A. Varnek, "Generative topographic mapping approach to chemical space analysis," in *Proc. Frontiers Mol. Des. Chem. Inf. Sci.-Herman Skolnik Award Symp.*, Jürgen Bajorath. Washington, DC, USA: ACS Publications, 2016, pp. 211–241.

[7] 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.

[8] S. A. Alsenan, I. M. Al-Turaiki, and A. M. Hafez, "Feature extraction methods in quantitative Structure–Activity relationship modeling: A comparative study," *IEEE Access*, vol. 8, pp. 78737–78752, 2020.

[9] S. Alsenan, I. Al-Turaiki, and A. Hafez, "A recurrent neural network model to predict blood-brain barrier permeability," *Comput. Biol. Chem.*, vol. 89, Dec. 2020, Art. no. 107377.

[10] S. Alsenan, I. Al-Turaiki, and A. Hafez, "Autoencoder-based dimensionality reduction for QSAR modeling," in *Proc. 3rd Int. Conf. Comput. Appl. Inf. Secur. (ICCAIS)*, Mar. 2020, pp. 1–4.

[11] 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.

[12] 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.

[13] 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.

[14] J. Lin and D. Gunopulos, "Dimensionality reduction by random projection and latent semantic indexing," in *Proc. Text Mining Workshop, 3rd SIAM Int. Conf. Data Mining*, 2003, pp. 1–10.

[15] 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.

[16] 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, 2019, pp. 119–139.

[17] 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.

[18] P. Maykel 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.

[19] 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 bloodbrain barrier penetrating and nonpenetrating agents by statistical learning methods," *J. Chem. Inf. Model.*, vol. 45, no. 5, pp. 1376–1384, Sep. 2005.

[20] J. A. Castillo-Garit, G. M. Casanola-Martin, H. Le-Thi-Thu, H. Pham-The, and S. J. Barigye, "A simple method to predict blood-brain barrier permeability of Drug- like compounds using classification trees," *Medicinal Chem.*, vol. 13, no. 7, pp. 664–669, Oct. 2017.

[21] Y. Brito-Sánchez, Y. Marrero-Ponce, S. J. Barigye, I. Yaber-Goenaga, C. M. Pérez, H. Le-Thi-Thu, and A. Cherkasov, "Towards better BBB passage prediction using an extensive and curated data set," *Mol. Informat.*, vol. 34, no. 5, pp. 308–330, May 2015.

[22] 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.

[23] 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.

[24] N. Sukumar, G. Prabhu, and P. Saha, "Applications of genetic algorithms in QSAR/QSPR Modeling," *Applications of Metaheuristics in Process Engineering*, J. Valadi and P. Siarry, Eds. Cham, Switzerland: Springer, 2014, pp. 315–324.

[25] E. Bonabeau, M. Dorigo, and G. Theraulaz, "Inspiration for optimization from social insect behaviour," *Nature*, vol. 406, no. 6791, p. 39, 2000.

[26] 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.

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

[28] 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.

[29] 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, Apr. 2019.

[30] 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.

[31] 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. 364–371.

[32] Q. Hu, M. Feng, L. Lai, and J. Pei, "Prediction of drug-likeness using deep autoencoder neural networks," *Frontiers Genet.*, vol. 9, p. 585, Nov. 2018.

[33] 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, Oct. 2018.

[34] 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.

[35] J. Wang, H. He, and D. V. Prokhorov, "A folded neural network autoencoder for dimensionality reduction," *Procedia Comput. Sci.*, vol. 13, pp. 120–127, Jan. 2012.

[36] 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.

[37] 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.

[38] Z. M. Hira and D. F. Gillies, "A review of feature selection and feature extraction methods applied on microarray data," *Adv. Bioinf.*, vol. 2015, pp. 1–13, Jun. 2015.

[39] E. B. Huerta, B. Duval, and J.-K. Hao, "Gene selection for microarray data by a LDA-based genetic algorithm," in *Pattern Recognition in Bioinformatics* (Lecture Notes in Computer Science), M. Chetty, A. Ngom, and S. Ahmad, Eds. Berlin, Germany: Springer, 2008, pp. 250–261.

[40] M. Perez and T. Marwala, "Microarray data feature selection using hybrid genetic algorithm simulated annealing," in *Proc. IEEE 27th Conv. Electr. Electron. Engineers Isr.*, Nov. 2012, pp. 1–5.

[41] N. Revathy and R. Balasubramanian, "Ga-svm wrapper approach for gene ranking and classification using expressions of very few genes," *J. Theor. Appl. Inf. Technol.*, vol. 40, no. 2, pp. 113–119, 2012.

[42] S. J. Susmi, "Hybrid dimension reduction techniques with genetic algorithm and neural network for classifying leukemia gene expression data," *Indian J. Sci. Technol.*, vol. 9, no. 1, pp. 1–8, Jan. 2016.

[43] N. Almugren and H. Alshamlan, "A survey on hybrid feature selection methods in microarray gene expression data for cancer classification," *IEEE Access*, vol. 7, pp. 78533–78548, 2019.

[44] 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.

[45] 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.

[46] *Alvadesc Software for Molecular Descriptors Calculation*. New York, NY, USA: Springer, Oct. 2019.

[47] I. Sushko *et al.*, "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.

[48] K. G. Sheela and S. N. Deepa, "Review on methods to fix number of hidden neurons in neural networks," *Math. Problems Eng.*, vol. 2013, pp. 1–11, May 2013.

[49] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Comput.*, vol. 9, no. 8, pp. 1735–1780, 1997.

[50] 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.

[51] S. Boughorbel, F. Jarray, and M. El-Anbari, "Optimal classifier for imbalanced data using matthews correlation coefficient metric," *PLoS ONE*, vol. 12, no. 6, Jun. 2017, Art. no. e0177678.

[52] D. Ballabio, F. Grisoni, and R. Todeschini, "Multivariate comparison of classification performance measures," *Chemometric Intell. Lab. Syst.*, vol. 174, pp. 33–44, Mar. 2018.

[53] Y.-Y. Sun, M. K. Ng, and Z.-H. Zhou, "Multi-instance dimensionality reduction," in *Proc. 24th AAAI Conf. Artif. Intell.*, 2010, pp. 1–6.

**SHROOQ A. ALSENAN** (Member, IEEE) received the Ph.D. degree in information systems sciences from the College of Computer and Information Sciences, King Saud University. She works as a Researcher and the Head of the Partnership and Investment Unit with the College of Computer and Information Sciences, Princess Nourah Bint Abdulrahman University, and a member of two research groups. Her research interests include chemoinfomratics, bioinfomratics, and deep learning.

**ISRA M. AL-TURAIKI** is currently an Associate Professor of Computer Science with King Saud University. Her research interests include 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 include data analytics, big data, bio informatics, cloud computing, and artificial intelligence.