


RESEARCH

Open Access



Artificial intelligence paradigm for ligand-based virtual screening on the drug discovery of type 2 diabetes mellitus

Alhadi Bustamam^{1*} , Haris Hamzah¹, Nadya A. Husna¹, Sarah Syarofina¹, Nalendra Dwimantara¹, Arry Yanuar² and Devvi Sarwinda¹

*Correspondence: alhadi@sci.ui.ac.id
¹ Department of Mathematics, Faculty of Mathematics and Natural Sciences, Universitas Indonesia, Depok, Indonesia
Full list of author information is available at the end of the article

Abstract

Background: New dipeptidyl peptidase-4 (DPP-4) inhibitors need to be developed to be used as agents with low adverse effects for the treatment of type 2 diabetes mellitus. This study aims to build quantitative structure-activity relationship (QSAR) models using the artificial intelligence paradigm. Rotation Forest and Deep Neural Network (DNN) are used to predict QSAR models. We compared principal component analysis (PCA) with sparse PCA (SPCA) as methods for transforming Rotation Forest. K-modes clustering with Levenshtein distance was used for the selection method of molecules, and CatBoost was used for the feature selection method.

Results: The amount of the DPP-4 inhibitor molecules resulting from the selection process of molecules using K-Modes clustering algorithm is 1020 with logP range value of -1.6693 to 4.99044. Several fingerprint methods such as extended connectivity fingerprint and functional class fingerprint with diameters of 4 and 6 were used to construct four fingerprint datasets, ECFP_4, ECFP_6, FCFP_4, and FCFP_6. There are 1024 features from the four fingerprint datasets that are then selected using the CatBoost method. CatBoost can represent QSAR models with good performance for machine learning and deep learning methods respectively with evaluation metrics, such as Sensitivity, Specificity, Accuracy, and Matthew's correlation coefficient, all valued above 70% with a feature importance level of 60%, 70%, 80%, and 90%.

Conclusion: The K-modes clustering algorithm can produce a representative subset of DPP-4 inhibitor molecules. Feature selection in the fingerprint dataset using CatBoost is best used before making QSAR Classification and QSAR Regression models. QSAR Classification using Machine Learning and QSAR Classification using Deep Learning, each of which has an accuracy of above 70%. The QSAR RFC-PCA and QSAR RFR-PCA models performed better than QSAR RFC-SPCA and QSAR RFR-SPCA models because QSAR RFC-PCA and QSAR RFR-PCA models have more effective time than the QSAR RFC-SPCA and QSAR RFR-SPCA models.

Keywords: Quantitative structure-activity relationship, K-modes clustering, CatBoost, Rotation Forest, principal component analysis, Sparse principal component analysis, Deep neural network, Fingerprint

Background

Type 2 diabetes is one of the fastest-growing chronic diseases due to decreased insulin function and insulin secretion [1]. Recent treatment for type 2 diabetes mellitus has been developed to increase the effect of incretin, which can improve insulin function and secretion. DPP-4 inhibitors are drugs that work as agents that inhibit the dipeptidyl peptidase-4 enzyme, which can increase the effects of incretin [2]. However, the registered DPP-4 inhibitor drugs have adverse side effects, such as upper respiratory tract infections, pancreatitis, and risk of heart failure, if taken for a long time. Therefore, development of new DPP-4 inhibitors is needed.

In-silico methods apply the use of computers as a tool in drug discovery that can perform cost-efficiently compared to conventional methods, which are known to be time-consuming and high cost [3]. They offer simulations and calculations that can rationally reduce the number of proposed compounds and assist in studying drug interactions with targets to the toxic properties of compounds and their metabolites [4]. QSAR is the ligand-based virtual screening method that studies the relationship between the chemical structures and biological activities of the molecules that can be calculated to derive a model or equation that can be used to predict the activity of a compound [4–6].

Several studies have performed QSAR against DPP-4 inhibitors. [2] developed QSAR using naïve Bayesian and recursive partitioning approaches to predict new DPP-4 inhibitors using thousands of available DPP-4 inhibitor molecule data. Hermansyah (2019) developed the QSAR method using XGBoost, Random Forest, Support Vector Machine, Multiple Linear Regression, and Deep Learning techniques to find a new DPP-4 inhibitor hit compound. These studies have relatively high predictive accuracy, which is above 80%. However, both studies still carried out a random selection method of molecules in the data partitioning stage. According to Andrada et al. (2015) [7], a random selection of molecules can lead to a mismatch because all members of the validation data may be members of the same group, thereby resulting in a molecular set that is not representative of the real data. Thus, a method is needed that can produce a representative data set in the data partition stage [2, 7, 8].

The problems in QSAR, in general, include the problem of transforming molecules into feature vectors that can be used as input data (feature extraction) and building a high-performance QSAR model. Rational selection of molecules in the data preparation stage of QSAR modelling is believed to affect the performance improvement of the QSAR model compared to random molecular selection. One method that can be used to select a subset of molecules rationally is clustering [7, 9]. Molecules can be uniquely identified from chemical databases by molecular descriptors [10]. Molecular descriptors are derived by several algorithms that describe specific aspects of a compound [11]. Molecular descriptors of compound molecules can be extracted from the Simplified Molecular Input Line Entry System (SMILES) format into a molecular fingerprint. SMILES is a general method that facilitates the representation and manipulation of molecular structures using computers with text strings and symbols, such as for single bonds, \bar{f} for double bonds, and # for triple bonds [9, 12, 13].

A fingerprint is a numerical representation concept of a particular structure or feature of a molecule that combines the presence or absence of different molecular substructures in a molecule into one molecular descriptor. Fingerprints can be classified into

non-hashed and hashed. A non-hashed fingerprint, also known as a structural key, is based on a predefined substructure dictionary, so there is a unique mapping between the position of the bit vector and a particular substructure. A hashed fingerprint is defined as a method that generates substructure set for a molecule that is converted into a fixed-length vector of bits. Molecular fingerprint data in the form of vectors containing bit strings with the numbers 0 or 1 can be considered as data with categorical features because they state the presence or absence of a particular molecular substructure in the fingerprint pattern of a molecule [11, 14]. In this research, the extended circular fingerprint (ECFP) and functional class fingerprint (FCFP) methods were used with diameters 4 and 6 [15], respectively, and a bit vector length of 1024, where the bit vector length represents the features of the data.

Clustering is a method that can be used to group or divide data by distance metrics that have been determined in high dimensional space. K-modes clustering is a clustering method that employs a similar procedure to K-means clustering. This method uses distance measures to handle categorical objects, replaces the average calculation method on the cluster with data mode, and uses frequency-based methods to update the data mode in the clustering process to minimise the clustering cost function [16]. Levenshtein distance is a distance method that can be used in the K-modes clustering as a dissimilarity function. It aims at measuring the similarity of two strings in the form of a sequence, where the order of the elements in one string is considered necessary using insertion, deletion, and substitution operations [17–19].

The artificial intelligence paradigm has been widely used to model QSAR, especially in the development of machine learning and deep learning methods. These methods have been successful in various applications of bioinformatics, including hypoxia detection in Cardiotocography (CTG) signals [20], prediction of protein interactions using amino acid sequences [21], biclustering method implementation on gene expression data of carcinoma tumour [22], protein interaction networks of schizophrenia's risk factor candidate genes [23], and Ebola virus phylogenetic analysis [24]. Rotation Forest is a machine learning method that uses PCA in forming a rotation matrix to rotate data sets that are then compiled into a decision tree. DNN is a deep learning method consisting of an input layer, several hidden layers, and an output layer. Several studies have been conducted to compare machine learning and deep learning methods in the drug discovery process. Similar performances of the two methods have been found with random data selection, but different performances can occur when data selection is made rationally [7, 25, 26]. Feature selection is one of the important steps for processing and analysing machine learning methods effectively. This technique can reduce the number of features of data while maintaining the same or even better learning performance by selecting a feature subset [27]. Important features are often useful for studying the relative importance or contribution of each feature in predicting targets [28]. CatBoost is a gradient boosting algorithm developed by [29]. The benefit of using the gradient boosting method is that it is relatively easy to determine the important score for each feature.

The analysis of the QSAR method is generally categorised into QSAR regression and QSAR classification [30]. The QSAR classification method is used to predict the active compound from the database, and then the QSAR regression method is used to study the activity value [30]. This study aims to build QSAR models using an artificial

intelligence paradigm, especially for the QSAR classification and regression models, to design a new DPP-4 inhibitor candidate for the treatment of type 2 diabetes. Machine learning and deep learning methods are used, including the Rotation Forest and DNN methods, respectively, where in the machine learning methods we compared PCA with sparse PCA (SPCA) as transformation methods in the framework of Rotation Forest. The K-modes clustering with Levenshtein distance method was used for the DPP-4 inhibitor molecular selection method and CatBoost was used for the feature selection method.

There are several novelties in this research. The first novelty in this study is the use of the K-modes clustering method using Levenshtein distance. The clustering analysis was carried on DPP-4 inhibitors through the fingerprints obtained by the ECFP and FCFP methods. The determination of the number of clusters in the K-modes clustering algorithm was evaluated by applying the Silhouette Coefficient method. Selection of DPP-4 inhibitor molecules was carried out based on the clustering results by taking one molecule from each cluster with the lowest logP value that is less than five, according to Lipinski's Rule of 5 [31]. The second novelty is the use of SPCA as a rotation matrix in the Rotation Forest method. The SPCA method is often used to perform dimensional reduction. In this study, SPCA was applied in the Rotation Forest method to transform feature data variables into new variables within an independent loading vector-matrix, in which all of the principal components (PCs) in the SPCA loading matrix were retained, to build QSAR models. CatBoost as a feature selection method for ECFP and FCFP fingerprint methods is a new thing in ligand-based virtual screening research. Therefore, this study will compare and analyse the performance of the QSAR model as a ligand-based virtual screening method with and without using the CatBoost feature selection method.

Methodology

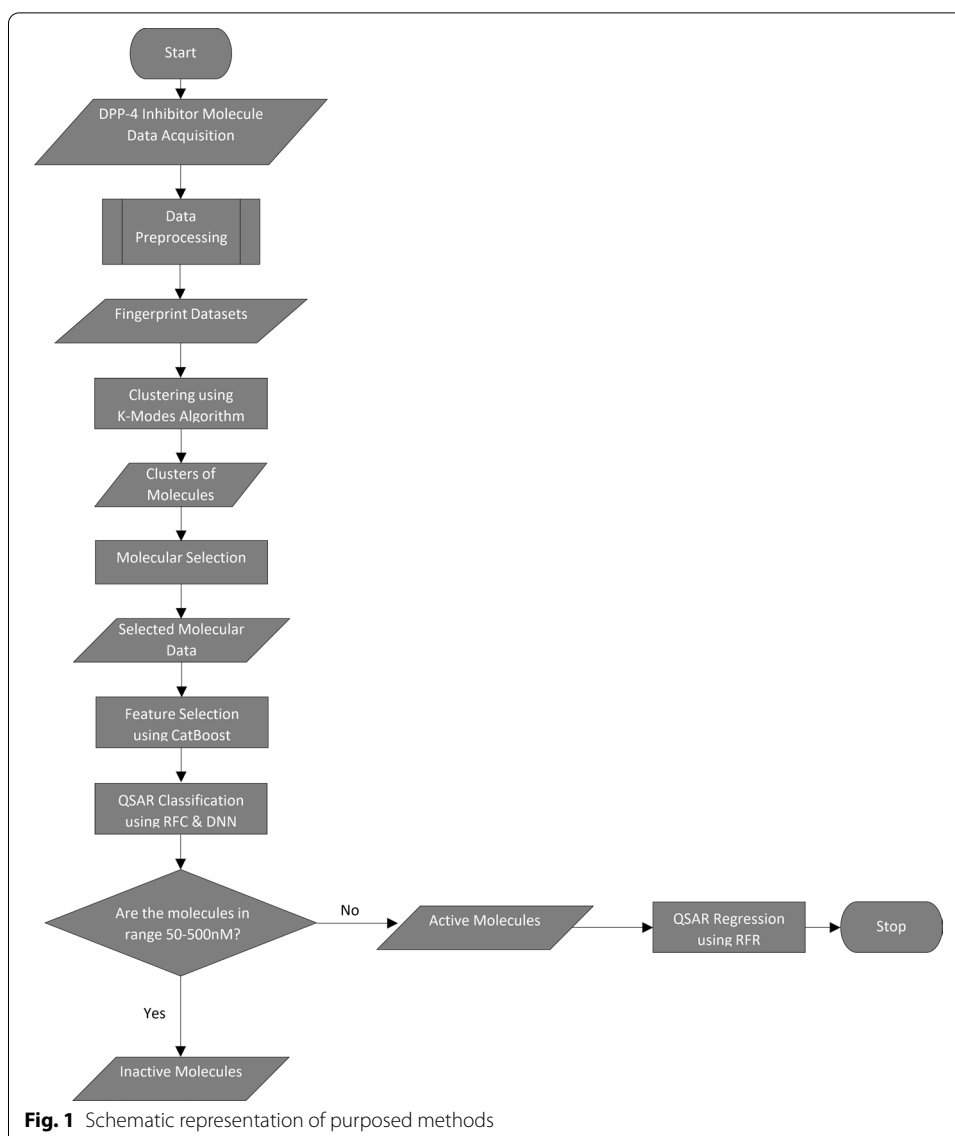
The schematic representation of proposed methods is shown in Fig. 1.

Data source and acquisition

The data used in this study is 4661 DPP-4 inhibitor molecular data in the form of Canonical SMILES which contains information on the molecular structure of the DPP-4 inhibitor with the type of biological activity IC₅₀. The data were obtained from <https://www.ebi.ac.uk/> site accessed in July 2019 through several stages, such as selecting the DPP-4 inhibitor and then selecting the biological activity value of IC₅₀. The IC₅₀ values have been converted to molar units pIC₅₀ (defined as log₁₀ IC₅₀).

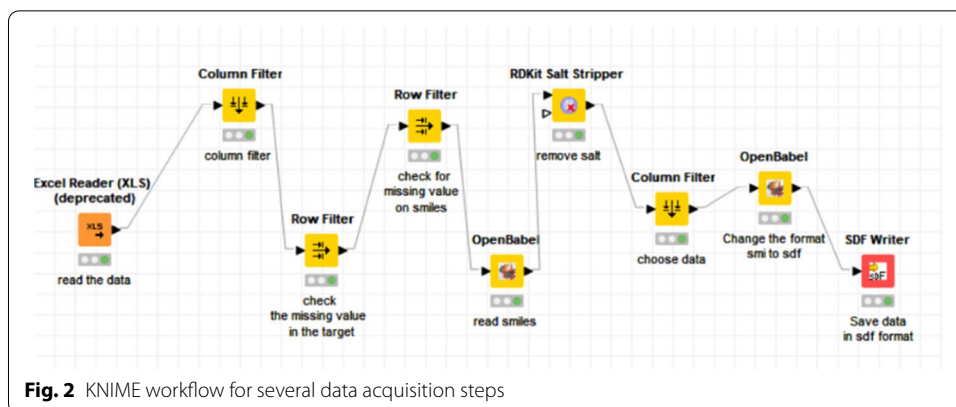
Data preprocessing

In the data preprocessing stage, there are three things that can be done, including data cleaning, data conversion, and feature extraction. The data were cleaned with the following criteria: (1) only selecting human DPP-4 inhibitor; (2) only selecting DPP-4 enzyme; (3) select data with biological activity value IC₅₀; (4) eliminating duplicate compounds. The IC₅₀ value is in the interval of 0.1 to 98,000 nM. In this study, compounds that have a value between 50 and 500 nM are defined as grey compounds, so they need to be removed. Besides, compounds having an IC₅₀ value below 50 nM are defined as active compounds and an IC₅₀ value above 500 nM as inactive compounds.



Several data cleaning steps, such as removing incomplete data rows in each column, reading Canonical SMILES columns with the OpenBabel library, removing duplicate molecular structures, and removing salt in the Canonical SMILES column using the RDKit Salt Stripper library, were executed using the KNIME software version 3.7, as shown in Fig. 2. KNIME nodes can do a wide set of functions for many different tasks such as read and write data files, data processing, statistical analysis, data mining, and graphical visualization. At KNIME there are RDKit and CDK nodes that are very complete for cheminformatics applications, from reading data in various formats to converting molecules from 2D to 3D [32].

At the data conversion step, the data is converted from CSV to SDF format. The feature extraction process carried out is calculating the molecular descriptors using several fingerprinting methods [11]. In this research, the RDKit packages which are implemented in the Python 3.6 programming language are used for the calculation of molecular



descriptors. The molecular fingerprints used are ECFP_4, ECFP_6, FCFP_4 and FCFP_6 fingerprints. The bit length of the ECFP_4, ECFP_6, FCFP_4 and FCFP_6 fingerprints is 1024 bits. The formation of the QSAR regression and QSAR classification models will divide the dataset into a training dataset by 80% and a test dataset by 20%. Then 80% of the training dataset will be randomly divided into k groups (folds). The number of folds used is five folds ($k = 5$).

Data clustering using K-modes

K-modes clustering is a clustering method that has a similar procedure to K-means clustering by expanding the paradigm for grouping categorical data by making several modifications, such as using simple dissimilarity measures to handle categorical objects, replacing the average calculation method on the cluster with the data mode; and using frequency-based methods to update the data mode to solve problems in the K-means clustering algorithm [16]. The K-modes clustering algorithm used in this study is described as follows:

1. Determine the number of clusters (k).
2. Determine the initial k mode for each cluster.
3. Calculate the distance between each data to the mode based on the dimension of dissimilarity.
4. Group objects against the cluster in the closest mode.
5. After all objects have been grouped into k clusters, recalculate the dimensions of the dissimilarity with respect to the current mode.
6. If there are objects whose closest mode belongs to another cluster, group the objects against that cluster and update the mode of both clusters.
7. Repeat steps 3 through 6 until no objects have moved clusters after all data has been tested.

In this study, the determination of the number of clusters was carried out by calculating the cluster evaluation value using the Silhouette Coefficient method. Silhouette studies the separation distance between the clusters generated in the clustering process which aims to measure the closeness of each object in a cluster to objects in other clusters.

The Silhouette values range between -1 and $+1$, with values close to $+1$ indicating the model with the best separation between clusters [33].

The metric used to measure distance, or a measure of dissimilarity, in the clustering algorithm in this study is the Levenshtein distance. There are three main parts in the Levenshtein distance calculation algorithm: initializing the distance matrix, calculating the distance matrix, and returning the value from the distance matrix with the largest value as a result of the Levenshtein distance. In this study, similarity strings were used in the grouping of DPP-4 inhibitor molecules through a comparison of bit vector strings, each of which contained a string of 0 or 1, from each molecule obtained based on the fingerprinting method, ECFP and FCFP [17–19].

Molecular selection

The selection of DPP-4 inhibitor molecules is made by taking one molecule from each cluster obtained from the results of clustering. Molecules are selected based on the lowest $\log P$ value and 'Lipinski's Rule of 5' rule, i.e. the $\log P$ value cannot be more than 5.

In this study, the calculation of the $\log P$ value was carried out based on the atomic-based approach method, as proposed by Crippen and Wildman in the RDKit module (1999). In this proposed method, the $\log P$ value is given by adding up the contributions of each atom, as given in Eq. 1, where n_i is the number of atoms of the i th atomic type and a_i is the contribution coefficient of the i th atomic type [34–37].

$$\log P = \sum n_i a_i \quad (1)$$

Before calculating the $\log P$ value, the molecular column can be added or displayed on the dataset first. After the $\log P$ value is obtained for each molecule, the dataset is then sorted according to the $\log P$ value. In this study, selection of molecules was carried out based on the lowest $\log P$ value of the molecules from each cluster, so that one molecule from each cluster would be selected, which was obtained from the clustering process with the lowest $\log P$ value.

Feature selection

CatBoost is an implementation of gradient boosting, which uses a binary decision tree as a basic prediction. Two important things that were introduced by CatBoost were the implementation of ordered boosting, namely permutation-based alternatives to classic algorithms, and algorithms for processing categorical features [29]. CatBoost divides the dataset into random permutations and applies ordered boosting to the random permutations. The advantage of using a gradient boosting decision tree is that it is relatively easy to take essential values for each attribute after the tree is built.

Using the CatBoost library in the Python programming language, prediction values change used to obtain essential features. For each feature, the prediction values change shows the average change in predictions if the feature values change—the more significant the importance, the greater the average change to the predicted value. The leaf pairs being compared have split values in different nodes to the leaf path. If it meets the splitting criteria, the object goes to the tree's left side; otherwise, it goes the other way. The following Eqs. 2 and 3 determines the significant feature value.

$$FI = \sum_{trees, leaf_{SF}} (v_1 - avr)^2 c_1 + (v_2 - avr)^2 c_2 \quad (2)$$

$$avr = \frac{v_1 c_1 + v_2 c_2}{c_1 + c_2} \quad (3)$$

Deep learning

Deep learning is a sub-field of machine learning that uses ANN algorithms, which are inspired by the structure and function of the human brain. DNN is a deep learning method that has been used since 2012 by Dahl et al. in the 'Merck Molecular Activity Challenge' to predict biomolecular targets in one drug [38]. The basic neural network model can be described in M liner combination with input variables x_1, \dots, x_D as follows [39].

Data goes to z_j neuron

$$a_j = \sum_{i=1}^D w_{ji}^{(1)} x_i + w_{j_0}^{(1)}, j = 1, \dots, M, \quad (4)$$

where $w_{ji}^{(1)}$ is denoted as the weight parameter and $w_{j_0}^{(1)}$ as the bias parameter.

Data out of z_j neuron

$$z_j = h(a_j) \quad (5)$$

Data goes to y_k neuron

$$b_k = \sum_{i=1}^M w_{kj}^{(2)} z_j + w_{k_0}^{(2)}, j = 1, \dots, M, \quad (6)$$

where $w_{kj}^{(2)}$ is denoted as the weight parameter and $w_{k_0}^{(2)}$ as the bias parameter.

Data out of y_k neuron

$$y_k = l(b_k). \quad (7)$$

For binary classification problems, each activation unit is transformed using the sigmoid logistic function so that:

$$\sigma(x) = \frac{1}{1 + \exp(-x)} \quad (8)$$

The overall neural network function becomes

$$y_k(x, w) = l \left(\sum_{j=0}^M w_{kj}^{(2)} h \left(\sum_{i=0}^D w_{ji}^{(1)} x_i + w_{j_0}^{(1)} \right) \right) + w_{k_0}^{(2)} \quad (9)$$

The classification problem is mathematically formulated as an optimisation problem where the objective function or loss function is the cross-entropy between the target vector and the predicted results. Given the dataset set $\{x_n, t_n\}_{(i=1)}^N$, where $x \in \mathbb{R}^m$ is the input vector and t_n is the target vector, cross-entropy is defined as [39]

$$\mathcal{L} = \frac{-1}{N} \sum_{i=1}^N [t_n \ln(y_n) + (1 - t_n) \ln(1 - y_n)]. \tag{10}$$

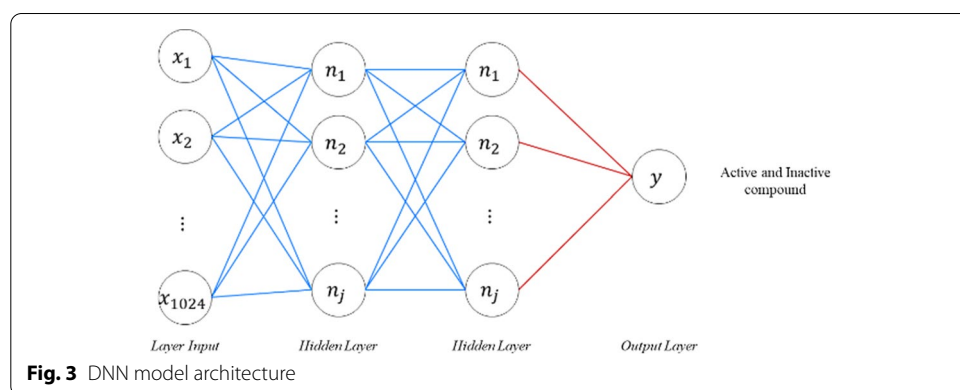
Furthermore, the best training procedure is known as the backpropagation algorithm, which is implemented using stochastic gradient descent [40]. The idea of the backpropagation algorithm is to fix errors from the output layer to the input layer with the chain rule. A simple approach to using gradient information is to select an update of the weights to form small steps towards a negative gradient, that is,

$$w^{(\tau+1)} = w^{(\tau)} - \eta \nabla E(w^{(\tau)}), \tag{11}$$

where the learning rate is used as a parameter to determine the speed of model learning process, $w^{(\tau+1)}$ is denoted as the new weight parameter and $w^{(\tau)}$ the weight before updating. At this stage, we evaluate $\nabla E(x) = 0$, which is used to solve optimisation problems such as stochastic gradient descent iteratively. Deep neural network architecture in this study consists of an input layer, 3 hidden layers, and an output layer. In this study, the selection of 3 hidden layers was carried out based on the research of [41], where the hidden layers are chosen between 2 and 5, while [42] chose 4 hidden layers for the DNN model architecture. Other hyperparameters used in this study include initialization of weights with random values normally distributed, the activation function used in each hidden layer is RELU, and the sigmoid function in the output layer [42], Adam’s optimizer is used as an optimization method in updating weights, using a dropout rate of 0.2 on the input layer and 0.5 in the hidden layer [43], the batch size is chosen was 32. The epoch used in the learning model process was 30 by applying the early stopping technique. The architecture of the DNN model in this research is illustrated in Fig. 3.

Rotation forest

Let $y = [y_1, y_2, \dots, y_n]^T$ be the set of class labels or response variables from the set w_1, w_2 . The decision tree in the ensemble is denoted by D_1, D_2, \dots, D_L and the set of independent variables in X is denoted by F . There are two parameters in this method: the number of decision trees denoted by L , and the number of original variables separator denoted by K . These two parameters have an essential role in determining the success of the Rotation Forest method. The first step in this method is to choose the number of



decision trees (L) used. Then, based on [44] to build each decision tree D_i for $i = 1, \dots, L$ will be determined by the following steps:

- 1 Randomly divide the set of independent variables F into K subsets. To increase the probability of high diversity in each tree, select disjoint subset so that each subset of feature contains $M = p/K$ features.
- 2 Let $F_{i,j}$ be the j -th feature subset for the training dataset D_i . Note $X_{i,j}$ as a data set with the variable set $F_{i,j}$, where $j = 1, 2, 3, \dots, K$. Randomly select a nonempty class subset and draw a bootstrap object sample of 75% of the total observations where $X_{i,j}^*$ is the bootstrapped data.
- 3 Apply PCA analysis to $X_{i,j}^*$. Use all PC coefficients from PCA and save as $a_{i,j}^{(1)}, a_{i,j}^{(2)}, \dots, a_{i,j}^{(M_j)}$ into matrix $C_{i,j}$ of $M_j \times 1$.
- 4 Principal Component Analysis (PCA) analysis on $X_{i,j}^*$. Use all principal component coefficients from PCA and save as $a_{i,j}^{(1)}, a_{i,j}^{(2)}, \dots, a_{i,j}^{(M_j)}$ into matrix $C_{i,j}$ of $M_j \times 1$.
- 5 Rearrange $C_{i,j}$ into a rotation matrix R_i of $p \times p$ with the coefficients obtained.
- 6 Rearrange the columns of the matrix R_i so that they correspond to the original variable subset F to construct the D_i tree. Then, state the rotational matrix composed of R_i^a .
- 7 Use XR_i^a as the training data cluster to build the D_i decision tree.
- 8 Estimate in each D the number of L trees followed by calculating the majority voting.

This research will use the Rotation Forest classification (PCA), which is developed using an algorithm that is under the Rotation Forest algorithm proposed by Rodriguez et al. (2006) and the Rotation Forest regressor proposed by Zhang et al. (2008) [44, 45].

Rotation Forest models that use PCA for classification and regression algorithms are called QSAR RFC-PCA and QSAR RFR-PCA respectively. Meanwhile, for models that use SPCA, they are called QSAR RFC-SPCA and QSAR RFR-SPCA. The differences between models that use PCA and SPCA is in the third step of Rotation Forest algorithm, in which SPCA analysis on $X_{i,j}^*$ is performed, and all the main component coefficients of the sparse loading are used.

QSAR model building

The main focus on this current work is to implement efficient modeling and well-defined models of QSAR. To accomplish this objective, it was necessary to solve two challenging issues in QSAR modelling. The first one is to carry out rational molecular selection to obtain a representative molecular subset and molecular descriptor selection to predict inhibitory concentration of DPP-4 inhibitor molecules. The second one is to make a modeling workflow as model validation, so that the result can be unbiasedly evaluated. A schematic of QSAR modeling workflow is shown in Fig. 1. This workflow starts with DPP-4 inhibitor molecules data acquisition and data preprocessing. After this step, the feature selection is carried out to identify an optimized non redundant descriptor that can lead to best models. Finally, when the descriptors are determined, it can be used to develop the QSAR Classification and QSAR Regression models. The QSAR Classification model building was executed using the Rotation Forest Classifier and DNN algorithms. QSAR Classification with the Rotation Forest Classifier algorithm uses 2 types of

matrix rotation methods, namely PCA and SPCA. Each of these models is called QSAR RFC-PCA and QSAR RFC-SPCA. Meanwhile, QSAR Classification with the DNN algorithm is called QSAR DNN. The QSAR Regression model building was executed using the Rotation Forest Regressor algorithm. QSAR Regression with the Rotation Forest Regressor algorithm also uses PCA and SPCA as the rotation matrix methods. Each of these models is called QSAR RFR-PCA and QSAR RFR-SPCA.

Evaluation

To determine the QSAR classification model, a confusion matrix that is used to indicate the number of observations was predicted correctly or not is required [46]. There are four parameters in this method, namely the true positive, false positive, false negative, and true negative.

Based on these parameters, the classification model evaluation metrics can calculate performance evaluation; such as Sensitivity, Specificity, Accuracy, and Matthews correlation coefficient (MCC), as explained as follows.

$$\text{sensitivity} = \frac{TP}{TP + FN}, \quad (12)$$

$$\text{specificity} = \frac{TN}{TN + FP}, \quad (13)$$

$$\text{accuracy} = \frac{TP + TN}{TP + FP + FN + TN}, \quad (14)$$

$$\text{MCC} = \frac{TP \times TN - FP \times FN}{\sqrt{(TP + FP)(TP + FN)(TN + FP)(TN + FN)}}. \quad (15)$$

To assess the performance of the QSAR regression model obtained, the coefficient of determination R^2 of root mean square error (RMSE) can be used. This criterion is determined using the formulas (15) and (16).

$$\sqrt{\frac{\sum (Y_i - \hat{Y}_i)^2}{n}} \quad (16)$$

$$1 - \frac{\sum (Y_i - \hat{Y}_i)^2}{\sum (Y_i - \bar{Y})^2}. \quad (17)$$

Results

Molecular selection result

Clustering procedure is used to assist in the molecular selection process. Meanwhile, The K-modes clustering algorithm is used to classify DPP-4 inhibitor compounds based on molecular fingerprints that can help the process of selecting molecules rationally. Levenshtein distance, which is used as a measure of dissimilarity in the K-modes clustering algorithm, is used to measure the closeness or similarity of the

DPP-4 inhibitor compound's molecules through a string comparison of molecular bit fingerprint vectors. The process of selecting DPP-4 inhibitor molecules was carried out based on the logP value criterion in the Lipinski rule after obtaining the results of grouping the DPP-4 inhibitor molecules in the clustering process using the K-modes clustering algorithm with Levenshtein distance.

Based on the results of grouping four datasets of molecular fingerprint for DPP-4 inhibitor compounds with the ECFP and FCFP methods, namely ECFP_4, ECFP_6, FCFP_4, and FCFP_6, the number of clusters from each dataset were obtained from the lowest being FCFP_4 = 1261, FCFP_6 = 1261, ECFP_4 = 1263, and ECFP_6 = 1265. The FCFP_4 dataset obtained the highest silhouette coefficient value (0.2574).

Based on the calculation of the logP value for 2053 DPP-4 inhibitor molecules, the logP value was obtained in the range of -1.6693 to 4.99044 , meaning that the logP value has met the criterion of Lipinski's rule that the logP value of a compound molecule must be less than 5. All fingerprint datasets labelled for each cluster are sorted according to the lowest logP value. After all the datasets are sorted, one molecule with the lowest logP value is taken from each cluster in each fingerprint dataset. 1263 molecules are obtained for the ECFP_4, 1265 for the ECFP_6, 1261 for the FCFP_4, and 1261 for the FCFP_6. Based on the selected molecules in all fingerprint datasets, 1020 molecules of the same DPP-4 inhibitor compound were obtained from all datasets. Those representative molecules can be further used for QSAR modelling.

Features selection result

This section will explain the results of selecting features using the CatBoost method. In assigning a value to essential features, each feature extraction dataset is used as an input vector at the learning stage of the CatBoost method. In this study, the CatBoost model learning was performed on each feature extraction data using the Python programming language with the CatBoost library. CatBoost models are generated, then there is an essential value in each feature. The feature that has the highest importance value indicates that it contributes the most in predicting the target. The performance results of the CatBoost model in predicting active and inactive compounds of DPP-4 inhibitors are presented in Table 1.

Table 1 informs that the accuracy of the CatBoost model results is above 0.840. Besides, the CatBoost model also has a balanced value between sensitivity, specificity, and MCC values. As a result, the recommended features as the essential features come from a good model. The following are the number of essential features for each

Table 1 CatBoost model performance

Datasets	Testing dataset			
	Sensitivity	Specificity	Accuracy	MCC
ECFP_4	0.899	0.784	0.843	0.689
ECFP_6	0.912	0.848	0.882	0.763
FCFP_4	0.889	0.870	0.880	0.759
FCFP_6	0.908	0.817	0.863	0.729

dataset, where the sum of the feature values qualifies the proportion values of 60%, 70%, 80%, and 90%.

There are 1024 features from the feature extraction results that are then selected using the CatBoost method. Fig. 2 explains that from the four fingerprint datasets, FCFP_4 received the least number of important features, by 55, with a proportion of 60%. In comparison, ECFP_4 received the most number of important features, by 280, with a ratio of 90%. Furthermore, these features will be used as input feature vectors to build the QSAR classification model using the Rotation Forest and DNN methods.

QSAR classification models using machine learning

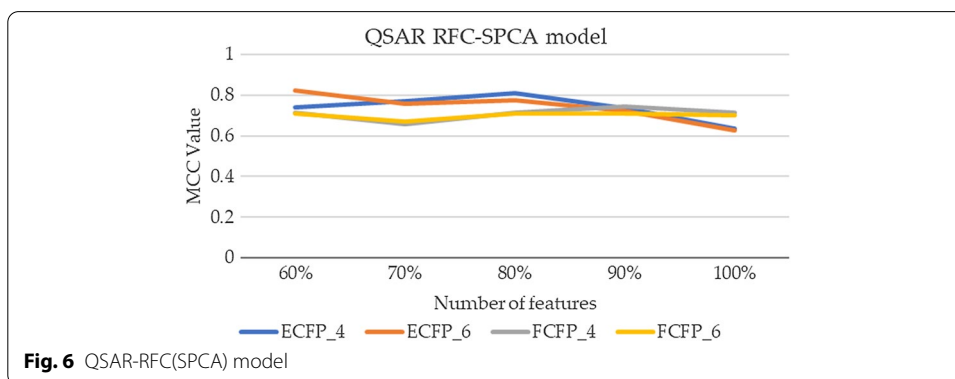
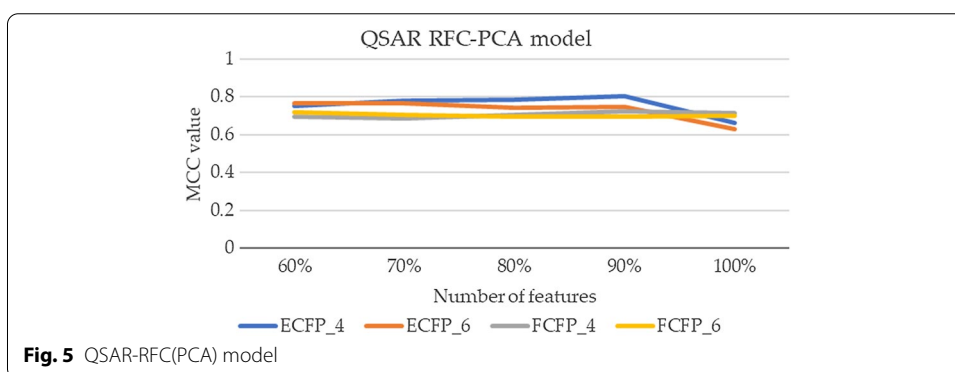
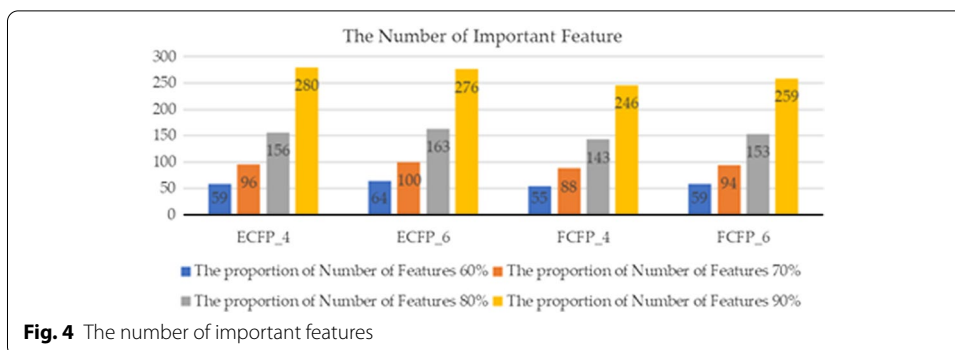
This section will explain the QSAR model in the machine learning approach using the Rotation Forest method to predict DPP-4 inhibitors into active and inactive compounds with different numbers of features based on the results of feature selection in the CatBoost method using a proportion of feature importance value of 60%, 70%, 80%, and 90% and without doing performing feature selection (100% feature importance). The QSAR classification model building using the machine learning approach is called QSAR-RFC(PCA) and QSAR-RFC(SPCA). Furthermore, it will be observed whether using some of the features in the feature extraction dataset results in an increase or decrease in the performance of the Rotation Forest model. The performance results of the QSAR-RFC(PCA) and QSAR-RFC(SPCA) models by selecting or not selecting features using the CatBoost method are presented in Figs. 5 and 6, respectively.

The feature selection with a proportion of 60% increased the MCC value in the feature extraction datasets ECFP_4, ECFP_6 and FCFP_6. Still, it did not increase the MCC value in the FCFP_4 feature extraction dataset, because there was a difference in the decrease in the performance of the MCC value by 0.018 in the QSAR-RFC(PCA) models. On the other hand, in the QSAR-RFC(SPCA) model, the MCC value increased in the feature extraction dataset of ECFP_4, ECFP_6, FCFP_4 and FCFP_6. Thus, to predict the DPP-4 inhibitors into active or inactive compounds using the QSAR-RFC(PCA) model combined with the FCFP_4 feature extraction, it is better to use all of the features to get a good model.

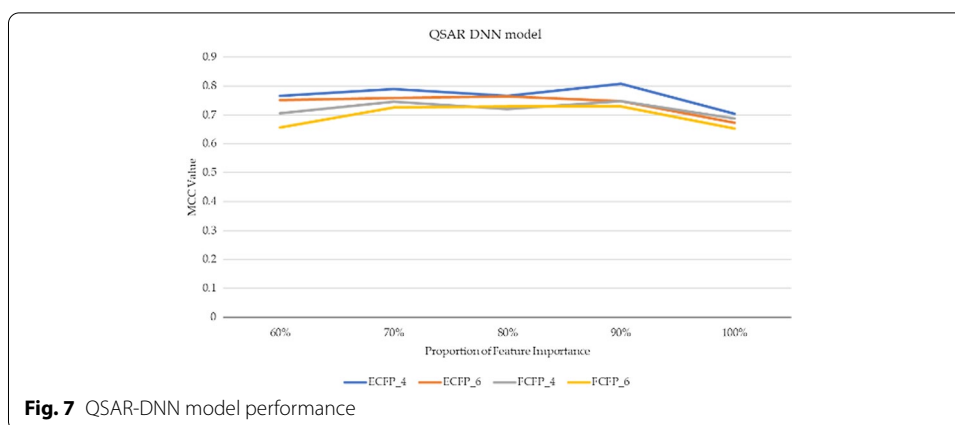
QSAR classification models using deep learning

This section will explain the QSAR model in the deep learning approach using the DNN method to predict DPP-4 inhibitors into active and inactive compounds with different numbers of features based on the results of feature selection in the CatBoost method utilising a proportion of feature importance value of 60%, 70%, 80%, and 90% and without making feature selection (100% feature importance). QSAR classification model building using a deep learning approach is called QSAR-DNN. Furthermore, it will be observed whether using some of the features in the feature extraction dataset results in an increase or decrease of the performance of the DNN model. The performance results of the QSAR-DNN by selecting or not feature using the CatBoost method are presented in Figs. 5 and 7.

The feature selection with a proportion of 60%, 70%, 80% and 90% increased the MCC value in the feature extraction dataset using QSAR-DNN models. For example, Fig. 4 shows that only by selecting features with a proportion of 60% in the ECFP_4



feature extraction, the QSAR-DNN can obtain an MCC value of 0.766; this is better than using all the features where the MCC value is 0.704. There are several features in the feature extraction of ECFP and FCFP that do not represent the molecular structure of the DPP-4 inhibitor properly, so these features need to be removed. The best QSAR-DNN in this study is the DNN model combined with ECFP_4 as a feature extraction method with a feature selection proportion of 90%. This model has sensitivity, specificity, accuracy and MCC values of 0.905, 0.903, 0.904 and 0.807, respectively.



Comparison of the QSAR classification model using machine learning and deep learning approach

The machine learning and deep learning methods used in this study are expected to predict each DPP-4 inhibitor molecular structure, whether it is an active or inactive compound. This section will compare the QSAR classification model using the machine learning approach, namely the QSAR-RFC(PCA) and QSAR-RFC(SPCA), with the QSAR classification model using deep learning approach, namely, QSAR-DNN. Based on the results obtained in this study, the QSAR model in the machine learning approach, the QSAR-RFC(SPCA) model combined with the ECFP_6 feature extraction and using the feature important value proportion of 60% of the features obtained an MCC value of 0.824, which is the highest MCC value. Meanwhile, the values for sensitivity, specificity and accuracy were 0.887, 0.938 and 0.911, respectively, and a running time of 11.622 s. Furthermore, the QSAR model, in the deep learning approach, has the best model, namely the QSAR-DNN model combined with ECFP_4. This model obtained the highest MCC value of 0.807 on ECFP_4 with a feature selection proportion of 90%, with the sensitivity, specificity and accuracy values of the QSAR-DNN model being 0.905, 0.903 and 0.904, respectively, and a running time of 2.046 s.

Balanced sensitivity, specificity and accuracy values are good criteria for a model [46], so that the model can distinguish between active and inactive compounds accurately. The QSAR classification model using the machine learning and deep learning approaches has shown a balanced performance between sensitivity, specificity and accuracy values. In this study, the author will not only observe the balance of sensitivity, specificity and accuracy values but also use the MCC value to compare the best performance of the QSAR classification model, because the MCC value is a more representative measurement metric when compared to accuracy [47]. The QSAR classification model using machine learning has the highest MCC value of 0.824, but its running time is longer than that of the DNN model.

QSAR regression

This section discusses the QSAR regression model using a Rotation Forest with a PCA rotation matrix and SPCA rotation matrix. This QSAR regression model uses

598 active molecules, with the lowest activity value of the test molecule being 48.9 nM ($pIC_{50} = 7.31$) and the highest 0.064 nM ($pIC_{50} = 10.19$). The machine learning approach using the Rotation Forest regression method will predict the value of DPP-4 inhibitor activity with several different features, based on the results of feature selection using the CatBoost Regressor method with the proportion of important feature values of 60%, 70%, 80%, and 90% and without making feature selection (100%). Furthermore, it will be observed whether using some of the features in the feature extraction dataset will increase or decrease the performance of the Rotation Forest regression model. In this study, the QSAR Rotation Forest regression model uses a PCA rotation matrix called QSAR-RFR(PCA), and the QSAR Rotation Forest regression model uses an SPCA rotation matrix called QSAR-RFR(SPCA). The performance results of the QSAR-RFR(PCA) model using the CatBoost feature selection method and without using the feature selection are presented in Tables 2, 3, 4 and 5.

Based on the simulation results, on the ECFP_4, ECFP_6, FCFP_4 and FCFP_6 datasets, the optimum features for the QSAR-RFR(PCA) model are 90%, because they have an R^2 value of 0.344, 0.345, 0.306 and 0.362 in each dataset. Meanwhile, without performing feature selection, the values for R^2 are 0.313, 0.278, 0.19 and 0.125 for each dataset, which means that the R^2 value is lower than the highest performance that can be achieved with the feature selection process. The performance results of

Table 2 QSAR-RFR(PCA) simulation results on ECFP_4 dataset

Number of features	R^2 value	Running time
60%	0.317	1.33
70%	0.299	2.00
80%	0.286	2.89
90%	0.344	4.28
100% (without feature selection)	0.313	16.30

Table 3 QSAR-RFR(PCA) simulation results on ECFP_6 dataset

Number of features	R^2 value	Running time
60%	0.309	1.42
70%	0.3405	2.02
80%	0.334	2.95
90%	0.345	4.49
100% (without feature selection)	0.278	15.38

Table 4 QSAR-RFR(PCA) simulation results on FCFP_4 dataset

Number of features	R^2 value	Running time
60%	0.186	1.18
70%	0.297	1.56
80%	0.1422	2.42
90%	0.306	3.64
100% (without feature selection)	0.19	15.68

Table 5 QSAR-RFR(PCA) simulation results on FCFP_6 dataset

Number of features	R ² value	Running time
60%	0.311	1.03
70%	0.353	1.47
80%	0.307	2.28
90%	0.362	3.61
100% (without feature selection)	0.125	13.75

Table 6 QSAR-RFR(SPCA) simulation results on ECFP_4 dataset

Number of features	R ² value	Running time
60%	0.352	10.84
70%	0.299	14.74
80%	0.285	22.15
90%	0.366	32.11
100% (without feature selection)	0.303	81.61

Table 7 QSAR-RFR(SPCA) simulation results on ECFP_6 dataset

Number of features	R ² value	Running time
60%	0.321	12.87
70%	0.384	17.02
80%	0.231	22.87
90%	0.408	33.1
100% (without feature selection)	0.233	87.61

Table 8 QSAR-RFR(SPCA) simulation results on FCFP_4 dataset

Number of features	R ² value	Running time
60%	0.334	8.29
70%	0.271	12.13
80%	0.245	17.44
90%	0.272	24.49
100% (without feature selection)	0.174	80.27

the QSAR-RFR(SPCA) model using the CatBoost feature selection method and without using the feature selection are presented in Tables 6, 7, 8 and 9.

Based on the simulation results, on the ECFP_4 and ECFP_6 datasets, the optimum features for the QSAR-RFR (SPCA) model are 90%, because they have R² values of 0.366 and 0.408 in each dataset, respectively. In the FCFP_4 dataset, the optimum number of features for the QSAR-RFR(SPCA) model is 60%, because it has an R² value of 0.334.

In the FCFP_6 dataset, on the other hand, the optimum number of features for the QSAR-RFR (SPCA) model is 70%, because it has an R² value of 0.353. Meanwhile,

Table 9 QSAR-RFR(SPCA) simulation results on FCFP_6 dataset

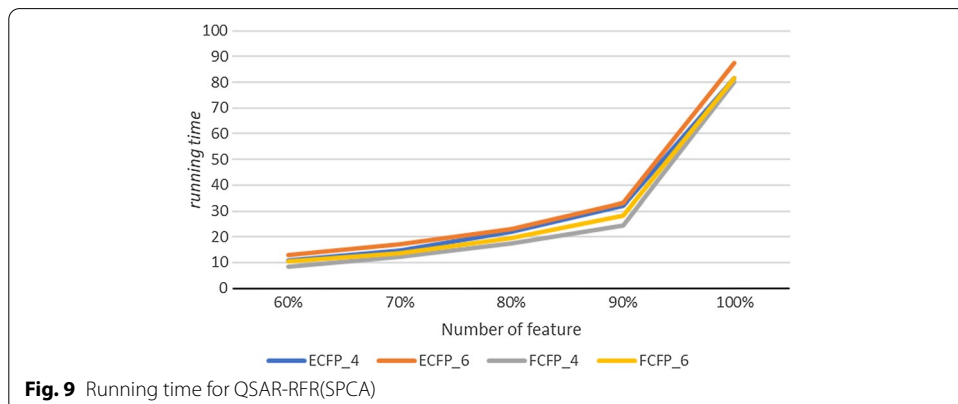
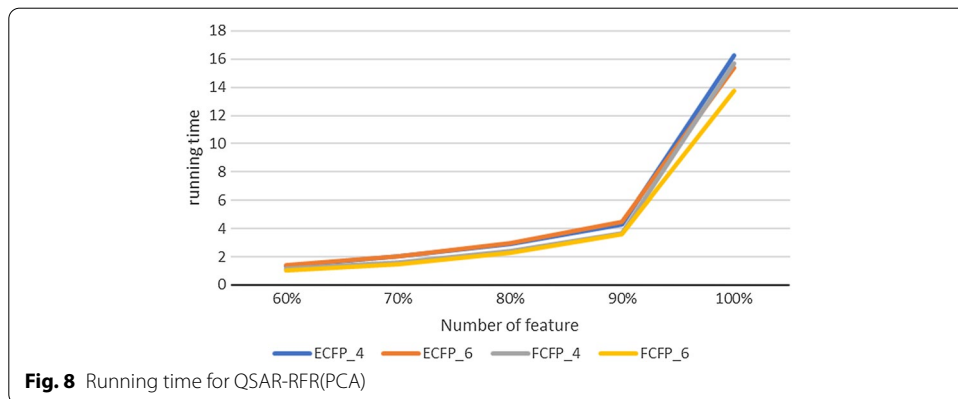
Number of features	R ² value	Running time
60%	0.299	10.66
70%	0.353	13.56
80%	0.301	19.57
90%	0.303	28.32
100% (without feature selection)	0.273	81.75

without performing feature selection, the values for R^2 are 0.303, 0.233, 0.174 and 0.273 for each datasets ECFP_4, ECFP_6, FCFP_4 and FCFP_6, meaning that the R^2 value is lower than the highest performance that can be achieved with the feature selection process.

Figs. 8 and 9 explain that the more features used, the longer the computation time required. From this, it can be observed that the importance of the feature selection process increases the value of R^2 and reduces the computation time in the QSAR-RFR(PCA) and QSAR-RFR(SPCA) models.

Conclusions

New dipeptidyl peptidase-4 (DPP-4) inhibitors need to be developed to be used as agents with low adverse effects for the treatment of type 2 diabetes mellitus. This study aims to build QSAR classification and regression models using machine



learning and deep learning algorithms. A representative subset of DPP-4 inhibitor molecules was resulted by applying the K-modes clustering algorithm with Levenshtein distance in the clustering process and analysing the selection of DPP-4 inhibitor molecules based on the logP value criteria of Lipinski's Rule of 5. Two thousand and fifty-three DPP-4 inhibitor molecules were obtained from the ChEMBL website. Clustering was carried out on the molecular fingerprint of DPP-4 inhibitors obtained from the SMILES feature. Several methods, such as ECFP diameter 4 and 6 and FCFP diameter 4 and 6, were used to construct four fingerprint datasets. From the selection process of molecules, 1020 representative DPP-4 inhibitor molecules are produced.

Before constructing the QSAR classification and regression model, feature selection was carried out using CatBoost. The results of feature selection using CatBoost are proven can improve the performance of the QSAR model, which requires a running time that is not too long. It is indicated by the results of the performance of QSAR-RFC, QSAR-RFR, and QSAR-DNN that have improved compared to using all features. The QSAR-RFC(PCA) method performed best in terms of predicting active and inactive DPP-4 inhibitors using ECFP_4, with an accuracy, sensitivity, specificity and MCC of 0.902, 0.914, 0.887 and 0.802, respectively, at 90% features with a running time of 4.307 s. The QSAR-RFC(SPCA) model using ECFP_6 has good performance, obtaining the corresponding values of 0.911, 0.887, 0.938 and 0.824, respectively, at 60% features with a running time of 11.622 s. The best QSAR-DNN model performance using ECFP_4 resulted in accuracy, sensitivity, specificity and MCC of 0.904, 0.905, 0.903 and 0.807, respectively, at 60% features with a running time of 2.046 s.

From these observations, the PCA rotation matrix's use to predict the activity value of the DPP-4 inhibitor IC50 is good enough. Based on the explanation above, in the QSAR regression model, the use of the PCA rotation matrix on the rotation Rotation Forest method obtains a more efficient computation time than modifying the rotation matrix with SPCA. The ECFP_4 dataset of the QSAR-RFR(PCA) model was able to get an R^2 value of 0.344 with a computation time of 4.28 s, while the QSAR-RFR(SPCA) model obtained an R^2 value of 0.366 with a computation time of 32.11 s. It explains why the SPCA rotation matrix increases the R^2 value by 0.022 but requires a very long computation time.

Abbreviations

DPP-4: Dipeptidyl peptidase-4; QSAR: Quantitative Structure-Activity Relationship; PCA: Principal Component Analysis; SPCA: Sparse Principal Component Analysis; ECFP: Extended Connectivity Fingerprint; FCFP: Functional Class Fingerprint; MCC: Matthew's Correlation Coefficient; RFC: Rotation Forest classification; RFR: Rotation Forest regression; DNN: Deep Neural Network; SMILES: Simplified Molecular Input Line Entry System; IC50: Half maximal inhibitory concentration; KNIME: The Konstanz Information Miner.

Acknowledgements

This research was supported by PUTI Q1 2020 grant from Universitas Indonesia with contract number NKB-1381/UN2.RST/HKP05.00/2020. The authors appreciate colleagues from the Directorate General of Higher Education (BRIN/DIKTI), the Directorate of Research and Community Engagement Universitas Indonesia, and Data Science Center Universitas Indonesia who contributed insights and expertise to advance this research in innumerable ways. We also would like to thank all anonymous reviewers for their constructive advice.

Authors' contributions

All authors contributed to the final version of the manuscript. AB led the research by conceiving of the presented idea. AB encouraged HH, NH, SS, and ND to investigate and with AY and DS supervised the findings of this work. HH, NH, SS, and ND performed the computations of both Rotation Forest (PCA) and Rotation Forest (SPCA) for QSAR classification and regression modellings. All authors read and approved the final manuscript.

Funding

This work was fully funded by PUTI Q1 2020 grant from Universitas Indonesia with contract number NKB-1381/UN2.RST/HKP05.00/2020. The funding body did not play any roles in the design of the study and collection, analysis, and interpretation of data and in writing the manuscript.

Availability of data and materials

The datasets generated and/or analyzed during the current study are available in the ChEMBL repository website, https://www.ebi.ac.uk/chembl/target_report_card/CHEMBL284/.

Declarations

Ethics approval and consent to participate

Not applicable

Consent for publication

Not applicable

Competing interests

The authors declare that they have no competing interests.

Author details

¹Department of Mathematics, Faculty of Mathematics and Natural Sciences, Universitas Indonesia, Depok, Indonesia.

²Faculty of Pharmacy, Universitas Indonesia, Gedung A Rumpun Ilmu Kesehatan Lantai 1, Depok, Indonesia.

Received: 13 January 2021 Accepted: 10 May 2021

Published online: 26 May 2021

References

1. World Health Organization. WHO: classification of diabetes mellitus. Geneva: World Health Organization; 2019. p. 36.
2. Cai J, Li C, Liu Z, Du J, Ye J, Gu Q, Xu J. Predicting DPP-IV inhibitors with machine learning approaches. *J Comput Aided Mol Des*. 2017;31(4):393–402. <https://doi.org/10.1007/s10822-017-0009-6>.
3. Lo Y-C, Rensi SE, Torng W, Altman RB. Machine learning in chemoinformatics and drug discovery. *Drug Discov Today*. 2018;23(8):1538–46. <https://doi.org/10.1016/j.drudis.2018.05.010>.
4. Geldenhuys WJ, Gaasch KE, Watson M, Allen DD, Van der Schyf CJ. Optimizing the use of open-source software applications in drug discovery. *Drug Discovery Today*. 2006;11(3-4):127–32. [https://doi.org/10.1016/s1359-6446\(05\)2018;23\(8\):1538-46](https://doi.org/10.1016/s1359-6446(05)2018;23(8):1538-46).
5. Patel BD, Ghate MD. Recent approaches to medicinal chemistry and therapeutic potential of dipeptidyl peptidase-4 (DPP-4) inhibitors. *Eur J Med Chem*. 2014;74:574–605. <https://doi.org/10.1016/j.ejmech.2013.12.038>.
6. Dearden JC. The history and development of quantitative structure-activity relationships (QSARs). *IJQSPR*. 2016;1(1):1–44. <https://doi.org/10.4018/ijqspr.2016010101>.
7. Andrada MF, Vega-Hissi EG, Estrada MR, Garro Martinez JC. Application of k-means clustering, linear discriminant analysis and multivariate linear regression for the development of a predictive QSAR model on 5-lipoxygenase inhibitors. *Chemometr Intell Lab Syst*. 2015;143:122–9. <https://doi.org/10.1016/j.chemolab.2015.03.001>.
8. Suhartanto H, Li X, Burrage K, Yanuar A, Bustamam A, Hilman M, Wibisono A. The development of integrated computing platform to improve user satisfaction and cost efficiency of in silico drug discovery activities. *Int J Adv Comput Tech* 2014;6(2):11–20.
9. Ramsundar B, Eastman P, Walters P, Pande V. Deep learning for the life sciences applying deep learning to genomics, microscopy, drug discovery, and more. 1st ed. Boston: O'Reilly; 2019. p. 238.
10. Rosselló F, Valiente G. Chemical graphs, chemical reaction graphs, and chemical graph transformation. *Electron Notes Theor Comput Sci*. 2005;127(1):157–66. <https://doi.org/10.1016/j.entcs.2004.12.033>.
11. Faulon JL, Bender A. Handbook of chemoinformatics algorithms. 1st ed. London: Chapman & Hall/CRC, Taylor & Francis Group; 2010. p. 454.
12. O'Donnell TJ. Design and use of relational databases in chemistry. 1st ed. Boca Raton: CRC Press; 2008. p. 224.
13. Weininger D. SMILES, a chemical language and information system: 1: introduction to methodology and encoding rules. *J Chem Inf Comput Sci*. 1988;28(1):31–6. <https://doi.org/10.1021/ci00057a005>.
14. Chackalamannil S, Rotella D, Ward S. Comprehensive medicinal chemistry III. 3rd ed. Amsterdam: Elsevier Ltd.; 2017. p. 4536.
15. Cereto-Massagué A, Ojeda MJ, Valls C, Mulero M, Garcia-Vallvé S, Pujadas G. Molecular fingerprint similarity search in virtual screening. *Methods*. 2015;71(C):58–63. <https://doi.org/10.1016/j.ymeth.2014.08.005>.
16. Huang Z. Extensions to the k-Means algorithm for clustering large data sets with categorical values. *Data Min Knowl Discov*. 1998;2(1998):283–304. <https://doi.org/10.1023/A:1009769707641>.
17. Khandare S, Gawade S, Turkar V. Design and development of e-farm with S.C.H.E.M.E. 2017 International Conference on Recent Innovations in Signal Processing and Embedded Systems (RISE). <https://doi.org/10.1109/risep.2017.8378223>.
18. Jurafsky D, Martin JH, Norvig P, Russell S. Speech and language processing: an introduction to natural language processing, computational linguistics, and speech recognition. 3rd ed. Stanford: Stanford University; 2019. p. 613.
19. Levenshtein VI. Binary codes capable of correcting deletions, insertions, and reversals. *Cybern Control Theory*. 1966;10(8):845–58.

20. Riskyana Dewi Intan P, Anwar Ma'sum MA, Alfiany N, Jatmiko W, Kekalih A, Bustamam A. Ensemble learning versus deep learning for Hypoxia detection in CTG signal. 2019 International Workshop on Big Data and Information Security, IWBSIS, 2019; 57–62 (2019). <https://doi.org/10.1109/IWBSIS.2019.8935796>
21. Bustamam A, Musti MIS, Hartomo S, Aprilia S, Tampubolon PP, Lestari D. Performance of rotation forest ensemble classifier and feature extractor in predicting protein interactions using amino acid sequences. *BMC Genom.* 2019;20(Suppl 9):1–13. <https://doi.org/10.1186/s12864-019-6304-y>.
22. Ardaneswari G, Bustamam A, Siswantining T. Implementation of parallel k-means algorithm for two-phase method biclustering in Carcinoma tumor gene expression data. *AIP Conference Proceedings.* 2017;1825. <https://doi.org/10.1063/1.4978973>.
23. Ginanjar R, Bustamam A, Tasman H. Implementation of regularized markov clustering algorithm on protein interaction networks of 2016. *ICACSI.* 2016;1(6):297–302.
24. Muradi H, Bustamam A, Lestari D. Application of hierarchical clustering ordered partitioning and collapsing hybrid in Ebola Virus phylogenetic analysis. *ICACSI 2015 - 2015 International Conference on Advanced Computer Science and Information Systems, Proceedings,* 2016;317–323. <https://doi.org/10.1109/ICACSI.2015.7415183>
25. Jing Y, Bian Y, Hu Z, Wang L, Sean X-Q, Chemical C, Screening G, Biology S. Paradigm for drug discovery in the big data era. *Aaps J.* 2018;20(3):1–22. <https://doi.org/10.1208/s12248-018-0210-0.Deep>.
26. Lenselink EB, Ten Dijke N, Bongers B, Papadatos G, Van Vlijmen HWT, Kowalczyk W, Ijzerman AP, Van Westen GJP. Beyond the hype: deep neural networks outperform established methods using a ChEMBL bioactivity benchmark set. *J Cheminform.* 2017;9(1):1–14. <https://doi.org/10.1186/s13321-017-0232-0>.
27. Rao H, Shi X, Rodrigue AK, Feng J, Xia Y, Elhoseny M, Yuan X, Gu L. Feature selection based on artificial bee colony and gradient boosting decision tree. *Appl Soft Comput J.* 2019;74:634–42. <https://doi.org/10.1016/j.asoc.2018.10.036>.
28. Hastie T, Tibshirani R, Friedman J. *The elements of statistical learning: data mining, inference, and prediction,* 2008; p. 764.
29. Prokhorenkova, L. O., Gusev, G., Vorobev, A., Dorogush, A. V. & Gulin, A. CatBoost: unbiased boosting with categorical features.. In S. Bengio, H. M. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi & R. Garnett (eds.), *NeurIPS* (p/pp. 6639-6649). 2018
30. Roy, Kunal & Kar, Supratik & Das, Rudra. (2015). A primer on QSAR/QSPR modeling: fundamental concepts. <https://doi.org/10.1007/978-3-319-17281-1>.
31. Lipinski CA, Lombardo F, Dominy BW, Feeney PJ. Experimental and computational approaches to estimate solubility and permeability in drug discovery and development settings. *Adv Drug Deliv Rev.* 2012;64:4–17. <https://doi.org/10.1016/j.addr.2012.09.019>.
32. Sydow D, Wichmann M, Rodríguez-Guerra J, Goldmann D, Landrum G, Volkamer A. Teachopencadd-knime: a teaching platform for computer-aided drug design using knime workflows. *J Chem Inf Model.* 2019;59(10):4083–6. <https://doi.org/10.1021/acs.jcim.9b00662>.
33. Rousseeuw PJ. Silhouettes: a graphical aid to the interpretation and validation of cluster analysis. *J Comput Appl Math.* 1987;20:53–65. [https://doi.org/10.1016/0377-0427\(87\)90125-7](https://doi.org/10.1016/0377-0427(87)90125-7).
34. Ghose AK, Crippen GM. Atomic physicochemical parameters for three-dimensional-structure-directed quantitative structure-activity relationships. 2. modeling dispersive and hydrophobic interactions. *J Chem Inf Comput Sci.* 1987;27(1):21–35. <https://doi.org/10.1021/ci00053a005>.
35. Rogers D, Hahn M. Extended-Connectivity Fingerprints. *Journal of Chemical Information and Modeling.* 2010;50(5):742–54. <https://doi.org/10.1021/ci100050t>.
36. Leach AR, Gillet VJ. *An introduction to chemoinformatics.* Revised. Dordrecht: Springer; 2007. p. 255.
37. Wildman SA, Crippen GM. Prediction of physicochemical parameters by atomic contributions. *J Chem Inf Comput Sci.* 1999;39(5):868–73. <https://doi.org/10.1021/ci990307l>.
38. Dahl GE, Jaitly N, Salakhutdinov R (2014). Multi-task Neural Networks for QSAR Predictions. *CoRR*, abs/1406.1231.
39. Bishop CM. *Pattern recognition and machine learning.* 1st ed. Singapore: Springer; 2006. p. 803.
40. Ma YA, Chen T, Fox EB. A complete recipe for stochastic gradient MCMC. *Advances in Neural Information Processing Systems.* 2015;29:17–2925. [arXiv:1506.04696](https://arxiv.org/abs/1506.04696).
41. Ghasemi F, Mehridehnavi A, Fassihi A, Pérez-Sánchez H. Deep neural network in qsar studies using deep belief network. *Appl Soft Comput.* 2018;62:251–8. <https://doi.org/10.1016/j.asoc.2017.09.040>.
42. Ma J, Sheridan RP, Liaw A, Dahl GE, Svetnik V. Deep neural nets as a method for quantitative structure-activity relationships. *J Chem Inf Model.* 2015;55(2):263–74. <https://doi.org/10.1021/ci500747n>.
43. Srivastava N, Hinton G, Krizhevsky A, Sutskever I, Salakhutdinov R. Dropout: a simple way to prevent neural networks from overfitting. *J Mach Learn Res.* 2014;15(56):1929–58.
44. Rodríguez JJ, Kuncheva LI, Alonso CJ. Rotation forest: a new classifier ensemble method. *IEEE Trans Pattern Anal Mach Intell.* 2006;28(10):1619–30. <https://doi.org/10.1109/TPAMI.2006.211>.
45. Zhang CX, Zhang JS, Wang GW. An empirical study of using rotation forest to improve regressors. *Appl Math Comput.* 2008;195(2):618–29. <https://doi.org/10.1016/j.amc.2007.05.010>.
46. Rokach L, Maimon O. *Data Mining with Decision Trees - Theory and Applications* (Vol. 69). WorldScientific; 2007. ISBN: 978-981-4474-18-4
47. Chicco D, Jurman G. The advantages of the Matthews correlation coefficient (MCC) over F1 score and accuracy in binary classification evaluation. *BMC Genom.* 2020;21(1):1–13. <https://doi.org/10.1186/s12864-019-6413-7>.

Publisher's Note

Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.