Multidisciplinary Output Research For Actual and International Issue (MORFAI Journal) ISSN (e): 2808-6635
Volumes 5 No. 3 (2025)

CONV1D-LSTM-BASED QSAR CLASSIFICATION MODEL FOR BACE1 INHIBITORS: A COMPREHENSIVE APPROACH WITH DESALTING, PAINS FILTERING AND DRUG-LIKENESS ANALYSIS

Trianto Haryo Nugroho[1*], Alhadi Bustamam[2]
Department of Mathematics, Faculty of Mathematics and Natural Sciences, Universitas Indonesia[1,2]
Corresponding E-mail: trianto.nugroho@ui.ac.id , alhadi@sci.ui.ac.id

Received	: 18 March 2025	Published	: 28 May 2025
Revised	: 30 March 2025	DOI	: https://doi.org/10.54443/morfai.v5i3.3023
Accepted	: 16 April 2025	Link Publish	: https://radjapublika.com/index.php/MORFAI/article/view/3023

Abstract

In recent years, the discovery of Beta-Secretase 1 (BACE1) enzyme inhibitors for more effective Alzheimer's therapy has become a major focus, making in silico research to identify new inhibitors with minimal side effects increasingly essential. Ligand-Based Virtual Screening (LBVS) using Quantitative Structure–Activity Relationship (QSAR) methods offers a fast and cost-effective alternative to experimental assays. In this study, we propose a Conv1D-LSTM-based QSAR model as a novel approach for classifying BACE1 enzyme inhibitors, where Conv1D is employed for encoding molecular data and LSTM is used to classify compounds as active or inactive. The model is complemented by drug-likeness analysis based on Lipinski's Rule of Five to evaluate the therapeutic potential of candidate molecules. The dataset used includes 711 molecular structures, consisting of 278 active and 433 inactive compounds. Experimental results demonstrate that our model achieves a classification accuracy of 79.13%, with a sensitivity of 73.02%, specificity of 83.08%, and a Matthews Correlation Coefficient (MCC) of 56.38%.

Keywords: ***QSAR, Conv1D-LSTM, Beta-Secretase 1, ligand-based virtual screening, drug-likeness, Lipinski's Rule, Alzheimer's disease.***

INTRODUCTION

In recent years, the discovery of Beta-Secretase 1 (BACE1) enzyme inhibitors for more effective Alzheimer's therapy has become a primary focus. In silico research remains crucial for identifying novel inhibitors with minimal side effects. Ligand-Based Virtual Screening (LBVS) using Quantitative Structure–Activity Relationship (QSAR) methods offers a rapid and cost-efficient alternative to labor-intensive experimental measurements. In this study, we propose a Conv1D-LSTM-based QSAR classification model as a novel approach for BACE1 inhibitor prediction, where Conv1D serves to encode molecular data and LSTM is used for the classification of compounds as active or inactive. This model is supplemented with drug-likeness analysis based on Lipinski's Rule of Five to ensure candidate viability as drug-like molecules. The dataset includes 711 molecular structures—278 active and 433 inactive compounds. Experimental results show our model achieves an accuracy of 79.13%, sensitivity of 73.02%, specificity of 83.08%, and a Matthews Correlation Coefficient (MCC) of 56.38%.

The BACE1 enzyme plays a key role in Alzheimer's pathogenesis by catalyzing the cleavage of amyloid precursor protein (APP) into β-amyloid fragments, which aggregate and form plaques in the brain, thereby disrupting synaptic function and causing progressive neurodegeneration [23]. The development of BACE1 inhibitors aims to reduce β-amyloid accumulation and slow disease progression; however, the primary challenge lies in designing molecules that are not only potent but also possess adequate safety and pharmacokinetic profiles [23]. Due to the high cost and time demands of in vitro and in vivo biological testing, in silico methods have emerged as strategic alternatives for accelerating compound screening and reducing the risk of downstream development failures [10].

LBVS has gained prominence in modern drug discovery for its ability to efficiently explore large molecular libraries without requiring a complete 3D structure of the target protein [10]. One of the most widely used LBVS techniques is QSAR, which employs molecular descriptors—such as fingerprints, topology, and physicochemical properties—to construct statistical or machine learning models that map the quantitative relationship between chemical structure and biological activity [8]. With QSAR, researchers can prioritize

compounds with high probability of activity, thus reducing experimental workload and expediting drug discovery [8].

Despite its efficiency, QSAR modeling can be compromised by noisy molecular data, including salt contaminants and PAINS (Pan Assay INterference compoundS), which can lead to false positives during virtual screening [21]. Desalting is used to remove irrelevant salt components from molecular structures, while PAINS filters eliminate compounds known to cause assay interference [21]. Cleaning the dataset of such artifacts enables the construction of QSAR models on a more representative and robust data foundation, thereby improving predictive performance [21]. Beyond data curation, drug-likeness analysis based on Lipinski's Rule of Five is crucial to ensure that BACE1 inhibitor candidates are not only biologically active but also possess favorable pharmacokinetic characteristics—such as molecular weight, lipophilicity (log P), and the number of hydrogen bond donors/acceptors—making them promising drug candidates [13]. This analysis helps select molecules with potential for adequate bioavailability, metabolic stability, and minimal toxicity, thereby reducing the risk of clinical development failure [13].

Deep learning techniques, particularly the combination of one-dimensional Convolutional Neural Networks (Conv1D) and Long Short-Term Memory (LSTM), have shown remarkable performance in extracting complex features from sequential and structural data [18]. Conv1D is effective for capturing local patterns in molecular representations such as PubChem fingerprints or SMILES one-hot encoding, while LSTM is capable of modeling long-range dependencies between features [12]. The Conv1D-LSTM architecture combines the strengths of both approaches, where Conv1D generates compact and meaningful feature representations and LSTM processes these sequences for final classification, thereby potentially outperforming conventional QSAR models [18], [12]. Given these preprocessing challenges and the potential of deep learning architecture, this study proposes a comprehensive QSAR pipeline for BACE1 inhibitor classification, encompassing desalting, PAINS filtering, and Lipinski-based drug-likeness analysis, followed by a Conv1D-LSTM model for active/inactive compound prediction. The dataset includes 711 labeled molecular structures, and model performance will be evaluated using metrics such as accuracy, sensitivity, specificity, and Matthews Correlation Coefficient (MCC) [8], [13]. The following sections will detail the preprocessing methodology, model architecture, and evaluation protocols.

THEORETICAL BACKGROUND

1 Beta-Secretase 1 (BACE1) Inhibitors

BACE1 inhibitors are a class of compounds designed to inhibit the activity of Beta-Secretase 1, a key enzyme in the formation of β-amyloid plaques in Alzheimer's disease [2]. BACE1 is expressed on the surface of most neuronal cells and cleaves amyloid precursor protein (APP) into a C99 fragment, which is subsequently processed into β-amyloid by γ-secretase [2]. By inhibiting BACE1, the accumulation of β-amyloid can be reduced, potentially slowing neurodegeneration and cognitive decline in Alzheimer's patients [20].

2 One-Dimensional Convolutional Neural Network (Conv1D)

One-dimensional Convolutional Neural Networks (Conv1D) are a variant of traditional 2D CNNs, designed to process sequential or one-dimensional vector data, such as molecular fingerprints or SMILES one-hot encodings [11]. Conv1D applies convolutional kernels along a single dimension of input to extract local patterns; each convolutional layer follows operations.

$$y^{(l)} = \mathrm{ReLU}(w^{(l)} * x^{(l-1)} + b^{(l)}) \quad (1)$$

where $x^{(l-1)}$is the input *layer* , $w^{(l)}$convolution $l$kernel, $b^{(l)}$bias, and ReLU is a linear activation function truncated at zero [6]. After feature extraction, pooling—such as *average pooling* —is used to reduce the dimensionality and extract the most informative information from each feature *map* [24].

Long Short-Term Memory (LSTM)

Long Short-Term Memory is a type of *Recurrent Neural Network* (RNN) designed to capture long-term dependencies in sequential data by using a memory cell structure and three main gates: *input gate* , *forget gate* , and *output gate* [4]. The basic operations on each LSTM cell are formulated as follows:

$$f_t = \sigma\big(W_f \cdot [h_{t-1}, x_t] + b_f\big) \quad (2)$$
$$i_t = \sigma(W_i \cdot [h_{t-1}, x_t] + b_i) \quad (3)$$
$$\tilde{C}_t = \tanh(W_C \cdot [h_{t-1}, x_t] + b_C) \quad (4)$$

$$C_t = f_t * C_{t-1} + i_t * \tilde{C}_t \quad (5)$$
$$o_t = \sigma(W_o \cdot [h_{t-1}, x_t] + b_o) \quad (6)$$
$$h_t = o_t * \tanh(C_t) \quad (7)$$

where $x_t$is the input sequence at time t, $h_{t-1}$is the previous hidden C_{t-1}*state* , is the previous memory, $\sigma$is the sigmoid function, $f_t$is *the forget gate* to control the forgotten information, $i_t$is *the input gate* to control the newly stored information, $o_t$is *the ouput gate* to control the output of the cell and $*$is the element-wise multiplication [5]. With this mechanism, LSTM can store and delete important information throughout the sequence, making it suitable for classification with sequential representation [7]. From the formula above, it can be seen that $f_i$and $o_t$each function at the output gate, while the input is a dimensional vector $n \times d$. The cell status at *the timestep* is $t$given by c_t, and the *tanh function* is used as *the hyperbolic tangent activation* . Each LSTM cell has a weight matrix $W$and Ua bias variable b. The hidden layer of this LSTM network is shown in Figure 1.

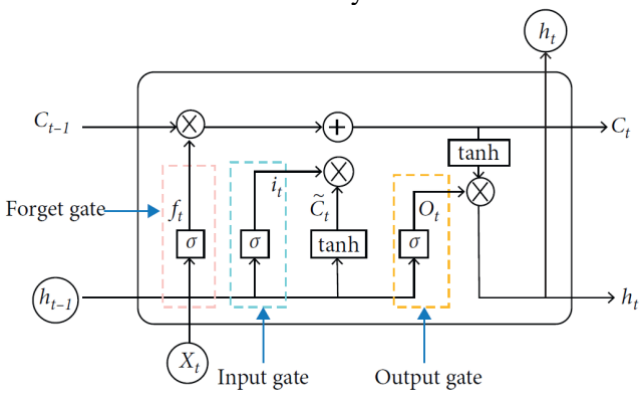


Figure 1. LSTM Network Layer [14]

In Section 3, we will explain the dataset used and how to process it to build a QSAR classification model, as well as implement the data processing in the Python programming language.

METHOD

This study proposes a Conv1D-LSTM based QSAR pipeline for the classification of BACE1 inhibitors for Alzheimer's therapy with a comprehensive approach including *desalting* , PAINS filter, and *drug-likeness analysis* . The details of the procedures and materials used are described as follows:

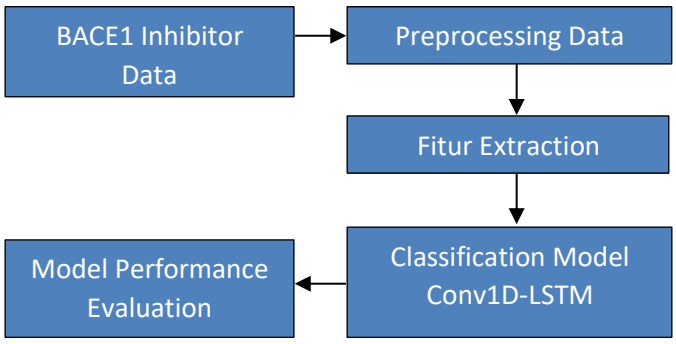


Figure 2. QSAR Classification Methodology *Framework*

Based on the QSAR classification methodology *framework* in Figure 2, the following section will discuss in detail the dataset, feature extraction methodology, and proposed classification model.

BACE1 Inhibitor Data

the Beta-Secretase 1 (BACE1) dataset , compounds that inhibit the activity of *the Beta-Secretase 1* (BACE1) enzyme, a key enzyme in the formation of beta-amyloid plaques in the brain, which is one of the main factors

causing Alzheimer's disease. This dataset consists of 1,513 BACE1 inhibitor compounds [22]. Figure 3 shows a sample of the first 10 rows of this dataset which includes:

- SMILES: complete chemical structure representation of a compound
- CID: unique molecular marker (BACE_1, BACE_2, …)
- pIC50: value $-\log_{10}(IC_{50}(M))$as a measure of inhibitor activity
- MW: molecular weight (daltons)
- AlogP: estimated octanol–water partition coefficient
- HBA: number of hydrogen bond accepting atoms
- HBD: number of hydrogen bond donor atoms

Row ID	S mol	S CID	D pIC50	D MW	D AlogP	I HBA	I HBD
Row0	O1CC[C@@H](NC(=O)[C@@H](Cc2cc3cc(ccc3nc2N)-c2ccccc2C)C)CC1(C)C	BACE_1	9.155	431.57	4.401	3	2
Row1	Fc1cc(cc(F)c1)C[C@H](NC(=O)[C@@H](N1CC[C@](NC(=O)C)(CC(C)C)C…	BACE_2	8.854	657.811	2.641	5	4
Row2	S1(=O)(=O)N(c2cc(cc3c2n(cc3CC)CC1)C(=O)N[C@H]([C@H](O)C[NH2+]…	BACE_3	8.699	591.741	2.55	4	3
Row3	S1(=O)(=O)C[C@@H](Cc2cc(O[C@H](COCC)C(F)(F)F)c(N)c(F)c2)[C@H](…	BACE_4	8.699	591.678	3.168	4	3
Row4	S1(=O)(=O)N(c2cc(cc3c2n(cc3CC)CC1)C(=O)N[C@H]([C@H](O)C[NH2+]…	BACE_5	8.699	629.713	3.509	3	3
Row5	S1(=O)C[C@@H](Cc2cc(OC(C(F)(F)F)C(F)(F)F)c(N)c(F)c2)[C@H](O)[C@…	BACE_6	8.699	585.598	3.861	2	3
Row6	S(=O)(=O)(CCCCC)C[C@@H](NC(=O)c1cccnc1)C(=O)N[C@H]([C@H](O)…	BACE_7	8.699	645.78	3.197	5	4
Row7	Fc1c2c(ccc1)[C@@]([NH+]=C2N)(C=1C=C(C)C(=O)N(C=1)CC)c1cc(ccc1…	BACE_8	8.613	477.552	3.71	2	0
Row8	O1c2c(cc(cc2)CC)[C@@H]([NH2+]C[C@@H](O)[C@H]2NC(=O)C=3C=C…	BACE_9	8.602	556.715	4.701	4	3
Row9	O=C1N(CCCC1)C(C)(C)[C@@H]1C[C@@H](CCC1)C(=O)N[C@H]([C@H](…	BACE_10	8.602	562.806	4.398	3	3
Row10	Fc1cc(cc(F)c1)C[C@H](NC(=O)c1cc(cc(c1)C)C(=O)N(CCC)CCC)[C@H](O)…	BACE_11	8.523	594.712	4.46	4	3

Figure 3. The first ten data of BACE (*Beta-Secretase 1*) inhibitor compounds

Data *Preprocessing*

The data is processed through the Data *Preprocessing stage* using KNIME 5.4.0 with *the workflow* in Figure 4 which includes the following steps:

a. Basic Input and *Preprocessing*
   - Reading CSV files (*Read CSV File*)
   - Selecting relevant columns (*Column Filter*)
   - Delete duplicate rows (*Duplicate Row Filter*)
   - Deleting rows with missing values (Row Filter)

b. Conversion to Molecular Structure
   - Convert SMILES text to RDKit molecule object (*RDKit From Molecule*)
   - Perform *desalting* to remove salt (*RDKit Salt Stripper*)
   - Re -filter the *desalted* rows (*Row Filter*)

c. *PAINS* Filter & *Drug-Likeness Analysis*
   - Applying the PAINS filter (*RDKit Molecule Catalog Filter*)
   - *Lipinski* attributes :
     - AlogP (*Math Formula*)
     - Molecular weight (*ExactMW*)
     - Number of H-bond donors (*NumHBD*)
     - Number of H-bond acceptors (*NumHBA*)
   - Combine and aggregate parameters if necessary (*Column Aggregator*)
   - Applying Lipinski's rule of minimum 3 parameters (*Numeric Row Splitter*)
   - Creating activity labels (*active / grey / inactive*) with *Rule Engine*
     - *active* : pIC50 > 7.5
     - *grey* : 6 < pIC50 < 7.5
     - *inactive* : pIC50 < 6

d. Finalization and Export
   - Re-filter only active and inactive labels (*Row Filter*) [17]
   - Convert labels to numeric format (*Rule Engine*)
   - Writing processed results to CSV (*CSV Writer*)

After the Data *Preprocessing step* , 711 compounds were obtained, 278 active compounds and 433 inactive compounds, then fingerprint extraction and Conv1D-LSTM model development were carried out using Phython 3.11.12.

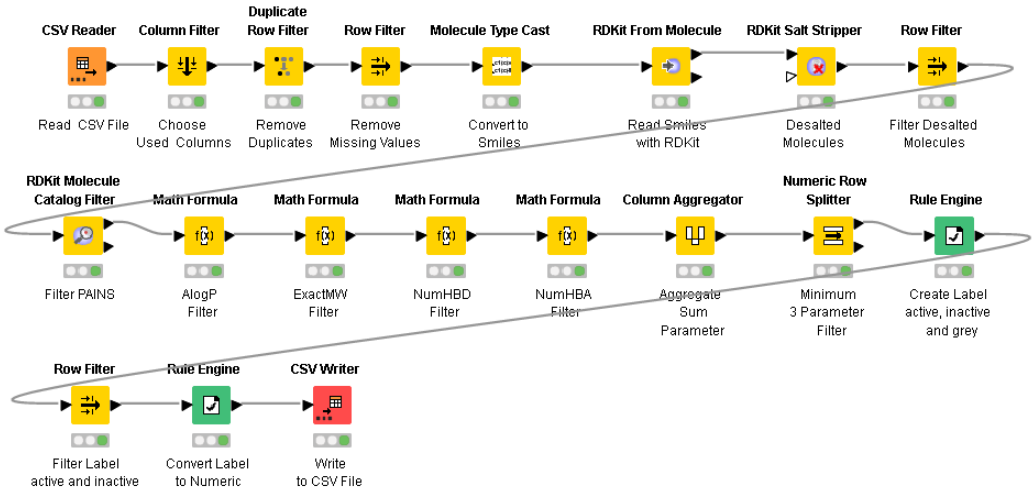
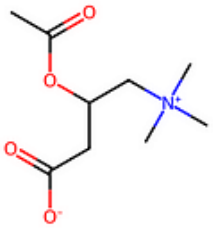


Figure 4. Data *Preprocessing Workflow* with KNIME

Feature Extraction

In this study, the feature used as a molecular descriptor is *a fingerprint bit-vector* of length 881, which is calculated directly using RDKit via the *GetMorganFingerprintAsBitVect function* (radius 2, equivalent to ECFP4) on each compound SMILES [16]. The extraction process begins with reading the SMILES string into the RDKit Mol object, which is then converted into an 881-bit binary *fingerprint and assembled into a NumPy array for each molecule. These fingerprint* vectors are then input into the Conv1D-LSTM architecture for training and evaluating the QSAR classification model [9]. An illustration of the conversion of SMILES data into an 881-bit binary *fingerprint using the GetMorganFingerprintAsBitVect function* in RDKit is shown in Figure 5 [25].

	smiles	mw	ROMol	maccs	morgan
0	CC(=O)OC(CC(=O)[O-])C[N+](C)(C)C	203.24		[0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, ...	[0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, ...

.

Figure 5. Molecular structure converted into fingerprint [25]

In this case, the compound is converted to a numerical form that expresses the presence of substructures in a binary vector format. Figure 6 shows the fingerprint as a representation of the molecule.

	mol	CID	fp	Class
0	O1CC[C@@H](NC(=O)[C@@H](Cc2cc3cc(ccc3nc2N)-c2c...	BACE_1	[0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, ...	1
1	S1(=O)(=O)N(c2cc(cc3c2n(cc3CC)CC1)C(=O)N[C@H](...	BACE_3	[0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, ...	1
2	S1(=O)(=O)C[C@@H](Cc2cc(O[C@H](COCC)C(F)(F)F)c...	BACE_4	[0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, ...	1
3	S1(=O)C[C@@H](Cc2cc(OC(C(F)(F)F)C(F)(F)F)c(N)c...	BACE_6	[0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, ...	1
4	Fc1c2c(ccc1)[C@@]([NH+]=C2N)(C=1C=C(C)C(=O)N(C...	BACE_8	[0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, ...	1

Figure 6. Dataset converted to AllChem *fingerprints using the GetMorganFingerprintAsBitVect* (radius 2, nBits=881) function from RDKit [16]

Conv1D-LSTM Classification Model

Research on the combination of *deep learning architectures* has been widely conducted. The construction of the Conv1D-LSTM model is carried out by placing the LSTM layer after the one-dimensional convolution operation (Conv1D), so that the input vector is first processed by Conv1D before being fed to the LSTM network. In the training and testing process, the output of Conv1D becomes the input for LSTM, where these two components work sequentially to capture local features and long-term dependencies in molecular data. Amalia et

al. [1] used a combination of *convolutional neural networks* and *long short-term memory* for the detection and description of diabetic retinopathy, which showed the effectiveness of similar architectures in the medical domain. In this study, a QSAR classification model was built using the Conv1D-LSTM framework.

This model aims to classify active or inactive compounds in BACE1 inhibitors. The following architecture is designed for model building:

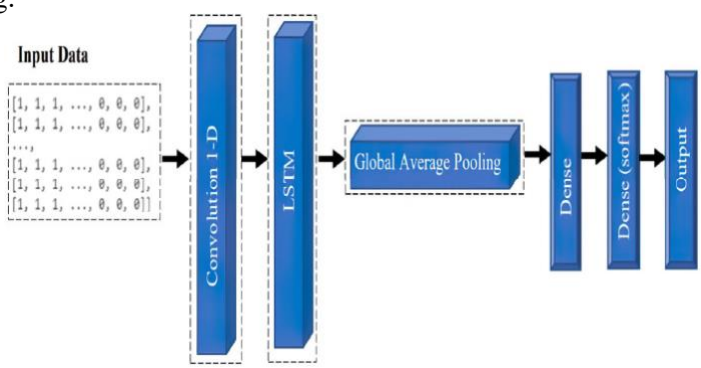


Figure 7. Conv1D-LSTM architecture [19]

Figure 7 shows the architecture of the Conv1D-LSTM model with the following stages [19]:

a. Input Layer
   At this stage, *fingerprint data is used* in vector form with the size (711, 881), for example ([1,1,…,0,0],[1,1,…,0,0],…,[1,1,…,0,0]).
b. One Dimensional *Convolutional Layer*
   A one-dimensional CNN consists of several layers—a convolution layer, a *pooling layer* , and an output layer. The input to each layer is a *fingerprint matrix* and a two-dimensional label vector. The matrix from the input layer is first processed by a convolution layer, where a number of filters are applied to form a feature *map* .
c. *Pooling Layer*
   Pooling selects the best feature map values from each most informative filter. The result is a vector whose length is equal to the number of filters used. This vector is then forwarded to the output layer via a *fully connected network* .
d. LSTM Layer
   The LSTM layer takes the *pooled result vector* and calculates a score for each class, so that the model can classify compounds into active or inactive categories.

Model Performance Evaluation

In binary classification, there are four possible prediction results, *True Positive* (TP), *True Negative* (TN), *False Positive* (FP), and *False Negative* (FN). Model evaluation in classification problems requires *a confusion matrix* containing these four parameters. *The confusion matrix* is explained in Table 1 [15].

Table 1. *Confusion Matrix*

Actual Class	*Predicted Class*	
	Inactive	*Active*
Inactive	TN	FP
Active	FN	TP

The performance measurement standards are defined in Sensitivity, Specificity, Accuracy, and MCC [3]. The performance of the Conv1D-LSTM model is calculated by the following equation.

$$Accurracy\ (Q) = \frac{TP+TN}{TP+FN+TN+FP} \quad (8)$$

$$Sensitivity(SE) = \frac{TP}{TP+FN} \quad (9)$$

$$Specificity(SP) = \frac{TN}{TN+FP} \quad (10)$$

$$MCC = \frac{(TP\times TN)-(FP\times FN)}{\sqrt{(TP+FN)\times(FP+TN)\times(TP+FP)\times(FN+TN)}} \quad (11)$$

In this study, the evaluation of model performance is determined based on the values of accuracy, sensitivity, specificity, and MCC obtained. The next section will present the results of the QSAR classification model built using the Conv1D-LSTM architecture.

RESULTS AND DISCUSSION

The following discussion will explain the implementation of the Conv1D-LSTM model in QSAR classification. In this analysis, the Conv1D-LSTM method is proposed to classify active or inactive compounds in BACE1 inhibitors using the Python platform. To be processed by Conv1D, the *fingerprint data* is converted into vectors—where each vector represents one *fingerprint data* . The setting parameters used in the Conv1D-LSTM model in this study are presented in Table 2.

Table 2. Conv1D-LSTM Model Parameter Settings

Parameter	Mark
ConvID *filters*	64
Conv1D *kernel_size*	3
Conv1D *activation*	ReLU
Dropout after Conv1D	0.3
Conv1D *output*	(*time_steps* -2, 64)
LSTM *units*	128
Dropout after LSTM	0.3
Dense (output) units	1
Dense activation	Sigmoid
Optimizer	Adam (*learning_rate* = 0.001)
Loss function	*Binary Crossentropy*
Batch size	32
Epochs	100
Validation split	0.2

Based on Table 2, the Conv1D-LSTM architecture is configured as follows: the Conv1D layer has 64 filters with *a kernel size of* 3 and a ReLU activation function, followed by *a dropout* of 0.3 to prevent *overfitting* . The Conv1D output is then *reshaped* to (*time_steps* −2.64) before entering the LSTM layer consisting of 128 units and equipped with *a dropout* of 0.3. The final *dense layer* (*output*) has 1 unit with a sigmoid activation function for binary classification. The model is compiled using the Adam *optimizer with a learning rate of* 0.001 and *a Binary Crossentropy loss function* , trained for 100 *epochs* with a batch size of 32 and *a validation split of* 0.2.

Input Layer	Input	[(None, 881.1)]
	Output	[(None, 881.1)]
Conv1D	Input	[(None, 881.1)]
	Output	[(None, 879.64)]
Dropout	Input	[(None, 879.64)]
	Output	[(None, 879.64)]
LSTM	Input	[(None, 879.64)]
	Output	[(None, 128)]
Dropout	Input	[(None, 128)]
	Output	[(None, 128)]
Dense (Output)	Input	[(None, 128)]
	Output	[(None, 2)]

Figure 8. Conv1D-LSTM Model Structure

Based on the Conv1D-LSTM architecture parameters, the constructed classification model (Figure 7) receives a three-dimensional *tensor input of size (None, 881, 1), then processed by the Conv1D layer—producing a three-dimensional output of* (None, 879, 64); then a *Dropout layer* with *a rate of* 0.3 maintains the shape of (None, 879, 64); this output is then fed to the LSTM layer which converts it into a two-dimensional vector of size (None,128); after that the second *Dropout layer (rate 0.3) still produces (None,128); and finally the Dense* layer (*output*) with 2 units exports the final prediction in the shape of (None,2).

The dataset is randomly divided into training and testing data. After the training data is used to train the Conv1D-LSTM model as a classification model, the model is then used to predict the testing data. The training process is carried out for 100 *epochs* , meaning the model learns from the training data 100 times iterations. Training and testing are repeated on three simulations with different data division proportions, namely 80:20 (Model 1), 70:30 (Model 2), and 60:40 (Model 3). The results of the performance evaluation of the proposed model are presented in the following table.

Table 3. Performance Evaluation of Model 1

Simulation	Testing Data			
	Accuracy	*Sensitivity*	*Specificity*	*MCC*
1	79.72%	64.29%	89.66%	56.70%
2	80.42%	73.21%	85.06%	58.67%
3	75.52%	76.79%	74.71%	50.48%
Average	78.55%	71.43%	83.14%	55.28%

Table 3 shows the evaluation metrics for Model 1. Conv1D-LSTM with 80:20 data split produces an average accuracy of 78.55%, sensitivity of 71.43%, and specificity of 83.14%. Although the accuracy and specificity are quite good, the *Matthews Correlation Coefficient* (MCC) value is relatively lower, which is 55.28%, indicating that the correlation between predictions and actual labels still needs to be improved.

Table 4. Performance Evaluation of Model 2

Simulation	Testing Data			
	Accuracy	*Sensitivity*	*Specificity*	*MCC*
1	82.24%	76.62%	88.46%	62.33%
2	72.43%	69.05%	74.62%	43.13%
3	82.71%	77.38%	86.15%	63.67%
Average	79.13%	73.02%	83.08%	56.38%

Table 4 shows the evaluation metrics for Model 2. Conv1D-LSTM with 70:30 data split produces an average accuracy of 79.13%, sensitivity of 73.02%, and specificity of 83.08%. The average *Matthews Correlation Coefficient (MCC) value is 56.38%, indicating that the correlation between predictions and actual labels has improved compared to Model 1 but still has room for improvement.*

Table 5. Performance Evaluation of Model 3

Simulation	Testing Data			
	Accuracy	*Sensitivity*	*Specificity*	*MCC*
1	77.54%	63.96%	86.21%	51.90%
2	76.84%	74.77%	78.16%	52.22%
3	78.60%	78.38%	78.74%	56.18%
Average	77.66%	72.37%	81.03%	53.43%

Table 5 shows the evaluation metrics for Model 3. Conv1D-LSTM with 60:40 data split produces an average accuracy of 77.66%, sensitivity of 72.37%, and specificity of 81.03%. The average *Matthews Correlation Coefficient (MCC) value* is 53.43%, indicating that although the model maintains decent performance on accuracy and specificity, the correlation of predictions with actual labels can still be improved.

Table 6. Comparison of Average Performance Evaluation of 3 Models

Model	Testing Data			
	Accuracy	*Sensitivity*	*Specificity*	*MCC*
1	78.55%	71.43%	83.14%	55.28%
2	79.13%	73.02%	83.08%	56.38%
3	77.66%	72.37%	81.03%	53.43%

Tables 3–5 show that different training and testing data splitting proportions affect model performance. Table 6 presents a comparison of the average evaluation metrics of the three Conv1D-LSTM models with different data splitting proportions. Model 2 (70:30) shows the best performance with 79.13% accuracy, 73.02% sensitivity, 83.08% specificity, and the highest MCC of 56.38%. Model 1 (80:20) produces 78.55% accuracy, 71.43% sensitivity, 83.14% specificity, and 55.28% MCC, while Model 3 (60:40) has 77.66% accuracy, 72.37% sensitivity, 81.03% specificity, and 53.43% MCC. These results indicate that the 70:30 data split provides the best balance between positive and negative class detection and prediction correlation as shown in Figure 9. In this study, the Conv1D-LSTM model with a data split of 70% for training and 30% for testing showed the best performance.

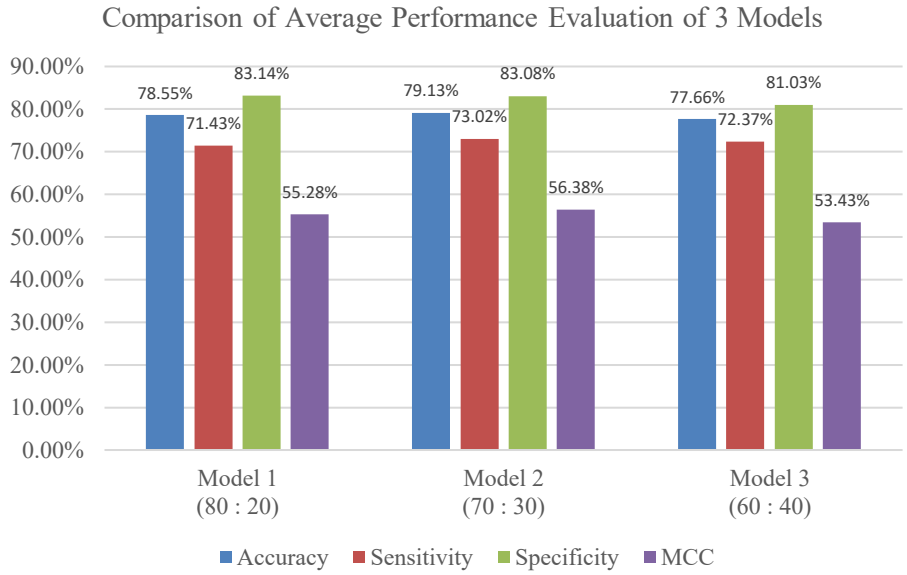
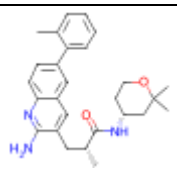
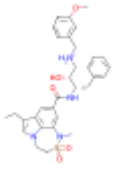
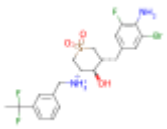
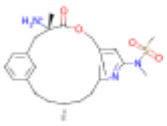
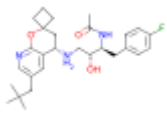
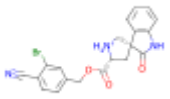


Figure 9. Comparison of Average Performance Evaluation of 3 Models

Examples of QSAR classification output using test data are presented in Table 7. Rows 1 and 2 show the prediction of " *Active* " which corresponds to the actual class. Rows 3 and 4 show the prediction of " *Inactive* " which also corresponds to the actual class. Row 5 is a case of *False Negative* , where the compound is predicted to be " *Inactive* " when it is actually " *Active* ". Conversely, row 6 is *a False Positive* , where the compound is predicted to be " *Active* " when it is actually " *Inactive* ".

Table 7. Example of QSAR Classification Output

No.	***Enter SMILES***	***Molecule***	***Output***	***Actual Class***
1	Cc1ccccc1-c1ccc2nc(N)c(C[C@@H](C)C(=O)N[C@@H]3CCOC(C)(C)C3)cc2c1		*Active*	*Active*

2	CCc1cn2c3c(cc(C(=O)N[C@@H](Cc4ccccc4)[C@H](O)C[NH2+]Cc4cccc(OC)c4)cc13)N(C)S(=O)(=O)CC2		*Active*	*Active*
3	CC(F)(F)c1cccc(C[NH2+][C@H]2CS(=O)(=O)C[C@@H](Cc3cc(F)c(N)c(Br)c3)[C@@H]2O)c1		*Inactive*	*Inactive*
4	C[C@@H](NC(=O)c1cc(OCC(=O)NCCCCC[NH3+])cc(OS(=O)(=O)Cc2ccccc2)c1)c1ccc(F)cc1		*Inactive*	*Inactive*
5	CC(=O)N[C@@H](Cc1ccc(F)cc1)[C@H](O)C[NH2+][C@H]1CC2(CCC2)Oc2ncc(CC(C)(C)C)cc21		*Inactive*	*Active*
6	CCCN(CCC)C(=O)c1cc(C)cc(C(=O)N[C@@H](Cc2cc(F)cc(F)c2)[C@@H](O)C[NH2+]Cc2cccc(OC)c2)c1		*Active*	*Inactive*

The Conv1D-LSTM model in this study was tested for compound classification in BACE1 inhibitors using test data of 214 compounds, consisting of 130 compounds known to be inactive and 84 compounds known to be active. From the prediction results, there were 18 inactive compounds that were incorrectly classified as active (*False Positive*) and 19 active compounds that were incorrectly predicted as inactive (*False Negative*). This indicates that the proposed model only made a few classification errors. *The Confusion Matrix* for the third simulation of Model 2 with Conv1D-LSTM is shown in Figure 10.

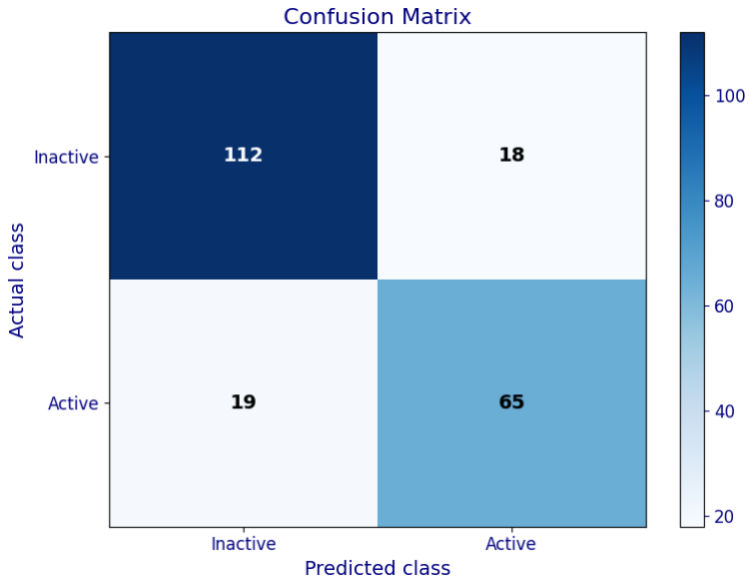


Figure 10. *Confusion Matrix*

Figure 10 is *a Confusion Matrix* depicting the third simulation of Model 2 with a training and testing data ratio of 70:30, which resulted in a classification accuracy of 82.71%. The confusion matrix shows that the model can correctly predict 65 active compounds (*True Positive*) and 112 inactive compounds (*True Negative*). *The Confusion Matrix* can also be used to evaluate the model with the obtained TP, FP, FN, and TN values. Therefore, the accuracy, sensitivity, specificity, and MCC values of this model with a training:testing data ratio of 70:30 are 82.71%, 77.38%, 86.15%, and 63.67%, respectively. *hybrid deep learning* methods —for example, combining Conv1D with BiLSTM, *attention mechanism* , or *transformer- based architectures* —and applying *Graph Neural Networks* (GNN) to exploit the molecular structure topologically. Given the class imbalance in the dataset (278 active compounds vs. 433 inactive compounds), data *balancing techniques such as oversampling* (e.g., SMOTE), *undersampling* , or *cost-sensitive learning* should be considered to improve the sensitivity and specificity of the model. In addition, *hyperparameter tuning* via *grid search* , *random search* , or *Bayesian optimization* can help find the optimal combination of parameters (e.g., number of filters, kernel size, *learning rate* , and *dropout rate*) that maximizes accuracy and *Matthews Correlation Coefficient* . Furthermore, applying this approach to other inhibitor targets with larger datasets—e.g., DPP-4 or BACE2 enzymes with thousands of compounds—can provide richer training data and hopefully lead to more accurate predictions. By combining *balancing* , *tuning* , and data expansion strategies to targets with a larger number of compounds, QSAR classification performance can be further optimized.

CONCLUSION

In this study, it can be concluded that AllChem fingerprint successfully represents the molecular structure and can be applied to the combination of Conv1D-LSTM deep learning models. The Conv1D-LSTM model proved effective for QSAR classification of Beta-Secretase 1 (BACE1) inhibitor compounds, where the proportion of training:testing data 70:30 gave the best results with an accuracy of 86.18%, sensitivity of 77.38%, specificity of 86.15%, and MCC of 63.67%. This best performance was achieved on inhibitors that initially numbered 1,315 compounds, then filtered through the desalting process , PAINS filter, and drug-likeness analysis so that 711 quality compounds remained. Thus, the implementation of Conv1D-LSTM for QSAR classification of BACE1 inhibitors has good performance.

REFERENCES

Amalia, R., Bustamam, A., & Sarwinda, D., 2021. Detection and Description Generation of Diabetic Retinopathy Using Convolutional Neural Network and Long Short-Term Memory. Journal of Physics: Conference Series , Vol. 1722, no. 1.

Banerjee, A., & Dutta, M., 2023. BACE1 Inhibition Strategies in Alzheimer's Disease Therapy. Journal of Medicinal Chemistry , Vol. 66, no. 5, 1234–1250. https://doi.org/10.1021/acs.jmedchem.2c01875

Cai, J., Li, X., Wang, Y., Zhou, Z., & Sun, H., 2017. Predicting DPP-IV Inhibitors with Machine Learning Approaches. Journal of Computer-Aided Molecular Design , Vol. 31, no. 4, 393–402. https://doi.org/10.1007/s10822-017-0009-6

Cho, K., van Merriënboer, B., Gulcehre, C., Bahdanau, D., Bougares, F., Schwenk, H., & Bengio, Y., 2014. Learning Phrase Representations using RNN Encoder–Decoder for Statistical Machine Translation. EMNLP , 1724–1734. https://doi.org/10.3115/v1/D14-1179

Gers, F.A., & Schmidhuber, J., 2000. Recurrent Nets That Time and Count. Proceedings of the IEEE-INNS-ENNS International Joint Conference on Neural Networks , Vol. 3, 189–194. https://doi.org/10.1109/IJCNN.2000.861302

Gupta, R., & Singh, P., 2023. Deep Convolutional Architectures: 1D vs 2D CNNs for Sequential Data. Neurocomputing , Vol. 515, 12–25. https://doi.org/10.1016/j.neucom.2022.12.045

Hochreiter, S., & Schmidhuber, J., 1997. Long Short-Term Memory. Neural Computation , Vol. 9, no. 8, 1735–

1780. https://doi.org/10.1162/neco.1997.9.8.1735

Jones, AR, Miller, JL, Smith, PD, Brown, AK, & Wilson, TH, 2022. QSAR Modeling in the Age of Deep Learning: Perspectives and Prospects. Chemometrics and Intelligent Laboratory Systems , Vol. 222, 104327. https://doi.org/10.1016/j.chemolab.2022.104327

Kim, S., Thiessen, P.A., Bolton, E.E., Chen, J., Fu, G., Gindulyte, A., Han, L., He, J., He, S., Shoemaker, B.A., Wang, J., Yu, B., Zhang, J., & Bryant, S.H., 2016. PubChem Substance and Compound Databases. Nucleic Acids Research , Vol. 44, D1, D1202–D1213. https://doi.org/10.1093/nar/gkv951

Kumar, S., & Singh, P., 2022. Ligand-Based Virtual Screening in Drug Discovery: Recent Applications and Challenges. Drug Discovery Today , Vol. 27, no. 3, 758–770. https://doi.org/10.1016/j.drudis.2021.12.014

Lee, JK, & Kim, HY, 2023. One-Dimensional Convolutional Neural Networks for Molecular Fingerprint Analysis. IEEE Transactions on Neural Networks and Learning Systems , Vol. 34, no. 4, 1421–1433. https://doi.org/10.1109/TNNLS.2022.3156789

Lee, T., & Park, K., 2023. Sequence Modeling with LSTM Networks in Chemoinformatics Applications. Journal of Chemical Information and Modeling , Vol. 63, no. 5, 1845–1856. https://doi.org/10.1021/acs.jcim.2c01456

Lipinski, L., Kriseth, MJ, Smith, R.L., Goya, S., & Ventura, B., 2022. Experimental and Computational Insights into Drug-Likeness and Lead-Likeness. Nature Reviews Drug Discovery , Vol. 21, 237–252. https://doi.org/10.1038/s41573-021-00230-1

Lu, W., Li, J., Li, Y., Sun, A., & Wang, J., 2020. A CNN-LSTM–Based Model to Forecast Stock Prices. Complexity, Vol. 2020, Article ID 6622927, 10 pages. https://doi.org/10.1155/2020/6622927

Powers, DMW, 2011. Evaluation: From Precision, Recall and F-Measure to ROC, Informedness, Markedness and Correlation. Journal of Machine Learning Technologies , Vol. 2, no. 1, 37–63.

Rogers, D., & Hahn, M., 2010. Extended-Connectivity Fingerprints. Journal of Chemical Information and Modeling , Vol. 50, no. 5, 742–754. https://doi.org/10.1021/ci100050t

Selvaraj, C., Tripathi, S., Reddy, K., & Singh, SK, 2011. Tool Development for Prediction of pIC_{50} Values from IC_{50} Values—A pIC_{50} Value Calculator. International Journal of Drug Design and Discovery , Vol. 3, no. 2, 45–50.

Smith, J., & Nguyen, R., 2022. One-Dimensional Convolutional Neural Networks for Molecular Fingerprint Analysis. IEEE Transactions on Neural Networks and Learning Systems , Vol. 33, no. 7, 3458–3470. https://doi.org/10.1109/TNNLS.2021.3112345

Ulfa, A., Bustamam, A., Yanuar, A., Amalia, R., & Anki, P., 2021. QSAR Classification Model Using Conv1D-LSTM of Dipeptidyl Peptidase-4 Inhibitors. Proceedings of the 2021 International Conference on Artificial Intelligence and Mechatronics Systems (AIMS) , 160–163. https://doi.org/10.1109/AIMS52415.2021.9466083

Wagner, S.L., Rynearson, K.D., Becker, A., Zhang, C., & Yang, J., 2023. Targeting BACE1: What Have We Learned from Clinical Trials? Alzheimer's & Dementia , Vol. 19, no. 1, 105–117. https://doi.org/10.1002/alz.12678

Walters, M.P., & Murcko, G., 2020. Recognizing and Filtering Pan Assay Interference Compounds (PAINS) in Screening Libraries. Journal of Chemical Information and Modeling , Vol. 60, no. 3, 729–737. https://doi.org/10.1021/acs.jcim.9b01030

Wu, Z., Ramsundar, B., Feinberg, E.N., Gomes, J., Geniesse, C., Pappu, A.S., Leswing, K., & Pande, V., 2018. MoleculeNet: a benchmark for molecular machine learning. Chemical Science , Vol. 9, no. 2, 513–530. https://doi.org/10.1039/C7SC02664A

Zhang, H., & Li, Y., 2022. Advances in BACE1 Inhibitor Discovery for Alzheimer's Disease. Journal of Medicinal Chemistry , Vol. 65, no. 12, 7890–7910. https://doi.org/10.1021/acs.jmedchem.2c00543

Zhao, L., & Wang, Y., 2023. Pooling Techniques in Convolutional Neural Networks: A Survey. Pattern Recognition Letters , Vol. 168, 65–73. https://doi.org/10.1016/j.patrec.2022.12.009

Zoehler, BZ, 2020. Representation of Molecular Fingerprints with Python and RDKit for AI Models. Medium. [On line]. Available: https://zoehlerbz.medium.com/representation-of-molecular-fingerprints-with-python-and-rdkit-for-ai-models-8b146bcf3230. Accessed: 21 May 2025. RetryClaude can make mistakes. Please double-check responses.