Received: September 7, 2020. Revised: October 6, 2020.

International Journal of
Intelligent Engineering & Systems

http://www.inass.org/

PSO-ELM with Time-varying Inertia Weight for Classification of SMILES Codes

Dian Eka Ratnawati[1,2]* **Marjono Marjono[2]** **Widodo Widodo[2]** **Syaiful Anam[2]**

[1]*Faculty of Computer Science, Brawijaya University, Malang, Indonesia*
[2]*Faculty of Mathematics and Natural Science, Brawijaya University, Malang, Indonesia*
* Corresponding author's Email: dian_ilkom@ub.ac.id

Abstract: Classification of active compounds is necessary to trivialize their properties and activities and can be carried out by identifying functional groups and their structures based on the simplified molecular input line entry system (SMILES) code. This study classified active compounds based on the SMILES structure using a combination of the machine learning techniques, extreme learning machine (ELM), and particle swarm optimization (PSO). ELM facilitates learning very quickly and at with high predictive accuracy. The PSO covers the limitations of the ELM, i.e., randomly determining the weight and bias, trial, and error in determining the number of hidden neurons. PSO optimizes the weight, bias, and number of hidden neurons. This study's results indicate that the classification of active compounds' function based on the SMILES code can be done by the ELM and PSO-ELM with high accuracy. The PSO-ELM can improve the accuracy performance of the ELM classification by optimizing the weight, bias, and number of hidden neurons automatically. Moreover, the proposed PSO-ELM is better than the PSO-ELM for average accuracy, computation time, and standard deviation. The proposed algorithm has the highest average accuracy compared to the ELM by increasing the accuracy to 2.54%, 6.43%, and 3.85% for 2, 3, and 4 classes, respectively. A comparison with other machine algorithms shows that the proposed is superior.

Keywords: SMILES, ELM, PSO, Feature selection, PSO.

1. Introduction

The SMILES code is a string; it represents a one-dimensional molecular structure discovered by D. Weininger [1]. The SMILE codes are widely used as the basis to classify compounds with better accuracy rather than using two-dimensional molecular structures [2]. Research by H. Öztürk, E. Ozkirimli, and A. Özgür has also shown that the SMILES code is better and more efficient to predict drug targets based on their structural similarities [3].

Recently, researchers have considered the SMILES code to classify its function because of its simplicity and fast processing compared to other molecular structures. Several previous studies used the SMILES code for the classification of 2 classes, i.e., metabolic and cancer, which is only categorized with 11 features. These researches were carried out using fuzzy KNN [4], C4.5 [5], and backpropagation[6] for the classification of cancer and non-cancer. Besides, D. F. Indarwati, D. E. Ratnawati, and S. Anam classified compounds into 3 classes (metabolic, infectious, and anti-inflammatory) with 15 features using the support vector machine (SVM) method to produce an average accuracy of 83.33% [7].

In this study, the SMILES code will be classified into 2, 3, and 4 classes using the ELM, with 29 features. The classification is based on its molecular function for the treatment of neurological diseases, inflammation, cancer, and viral infections. The ELM can learn very quickly and has excellent performance, which is better than gradient descent-based learning algorithms [8, 9]. This performance advantage was shown by G.-B. Huang, Q. Zhu, and C. Siew [9], and A. Toprak [10]. G.-B. Huang, Q. Zhu, and C. Siew proved that the ELM has a better performance compared to the SVM, SAOCIF, Cascade-

Correlation, C4.5, and RBF. A. Toprak's report shows that ELM has the best accuracy and speed compared to Naïve Bayes, SVM, and artificial neural networks [10]. Another advantage of ELM is the algorithm only requires one parameter, that is, the number of hidden neurons. However, to get the right number of hidden neurons, it needs a lot of testing and takes time. Determining the correct amount of hidden neurons can be improved by the optimization algorithm, PSO [11, 12].

Some of the advantages of PSO compared to other optimization algorithms are [13-15] first, it is simple and easy to implement. Second, PSO has control parameters to balance global exploration and local search space. Third, it has a memory for saving information in the best condition. Fourth, PSO excels in convergence, speed, and accuracy. Finally, it only updates the speed parameters. Because of the advantages of PSO, this study used PSO to optimize the number of hidden neurons as well as optimize the weight and bias. If the randomly generated weights and bias are 0, this means that some hidden layer nodes are invalid [13], thus requiring a higher number of hidden neurons to produce the desired accuracy [16].

The PSO-ELM has been used for the optimization of weight and bias [13], the number of hidden neurons to identify groundwater contamination [16], and to optimize the input, bias and the number of hidden neurons applied for cancer classification [12]. The novelty of this research is to apply PSO-ELM for SMILES code classification and use inertia weights (w) and c_1 and c_2 adaptive properties.

This rest of the paper is organized as follows: Section 2 discusses the features of the SMILES code, ELM, PSO, and the inertia weights. Section 3 discusses the proposed methodology. Section 4 deals with results, discussion, and finally, conclusions.

2. Material and methods

2.1 Simplified molecular input line system (SMILES)

The SMILES is a modern chemical notation system designed by D. Weininger [1]. It is suitable for high-speed machine processing and is easy to use for predictive modeling. The molecular structure is determined uniquely, accurately, and can represent the characteristics of the molecule [1].

Apart from using the SMILES code, molecular structures can be represented using two dimensions (2D) and three dimensions (3D). An example of representation of the molecular structure for methyl eugenol, which functions as an anti-cancer agent, can be seen in Fig. 1.

COC1=C(C=C(C=C1)CC=C)OC

(a)

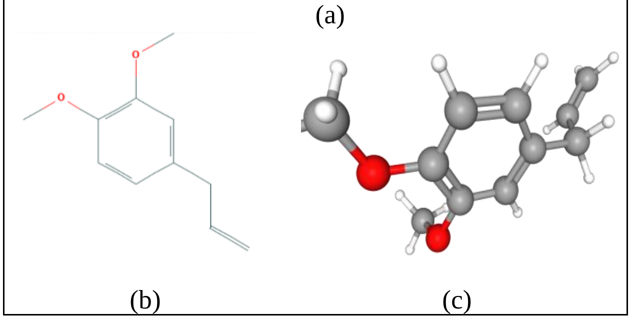

Figure. 1 SMILES code, 2-dimensional (2D) and 3-dimensional (3D) structure of a compound: (a) SMILES code, (b) molecule structure - 2D, and (c) molecule structure - 3D

2.2 Feature representation

Features are specific characters taken from the SMILES code. This study used 29 features, which is the highest compared to 11, 15, 22, and 27 features[17]. The value of each feature is a real number with a range of 0–1, and the details of the features are listed in Table 1.

In the 28th feature, the total valence is not divided by the length of the SMILES code because the value is higher than the SMILES code; therefore, normalization is carried out to get the value range between 0 and 1. Normalization of the total valence (x_{tv}) is presented in Eq. (1),

$$x_{tv}{}^{*} = \frac{tv- min}{max-min} \qquad (1)$$

where tv is the total valence, min is the minimum value of the total valence, and max is the maximum value of the total valence.

2.3 Extreme learning machine (ELM)

The ELM is an artificial neural network algorithm with a single layer feedforward network (SLFN) architecture [8-9]. It consists of one input layer, one hidden layer, and one output layer. The ELM uses randomly generated weights and bias, whereas a Moore–Penrose generalized inverse operation analyses the output of the hidden layer. The performance of this algorithm may be close to or exceed SVM and backpropagation; even the computation speed could be faster tens to thousands of times [18]. The ELM can avoid local optimal problems, incorrect learning rate, and overfitting that usually occurs in gradient-descent-based methods. It can be applied to small, medium, large, and complex problems. Another advantage is that the algorithm is easy to implement with a small error and excellent

Table 1. Features used and their descriptions

No	Feature	Description
1-11	Atom	The number of occurrences of atoms B, C, N, O, P, S, F, Cl, Br, I and OH in the SMILES code
12	=	Number of double bond occurrences
13	#	Number of triple bond occurrences
14	@	Number of chirality occurrences
15	()	Number of branching appearances
16	[ ]	Number of occurrences of atom groups
17	+	The number of occurrences of the cation / positive ion
18	-	Number of occurrences of anions / negative ions
19	.	Number of occurrences of the ionic bond
20	:	Amount of aromatic bond
21	C=C	The number of occurrences C = C
22	N+	The number of occurrences is N +
23	[O-]	Number of occurrences [O-]
24	C=O	The number of occurrences C = O
25	COC	Number of COC occurrences
26	charge	The difference between cations and anions
27	total cyclic	The number of cyclic in the SMILES code
28	total valence	The number of valence atoms that make up a compound without H bonds and 3 free electrons
29	Length of SMILES code	Used as a divider for features 1 - 27

performance [9].

ELM algorithm:

Given the training data $(\boldsymbol{x_i}, \boldsymbol{t_i})$ where $\boldsymbol{x_i} = [x_{i1}, x_{i2}, \ldots, x_{in}]$ for $\boldsymbol{x_i} \in \mathbf{R^n}$ and $\widetilde{N}$ is the number of the hidden neuron. The ELM algorithm steps are [9]:

1. Generate input weights $\boldsymbol{w}_{i,}$ and bias $\boldsymbol{b}_{i,}$ $i = 1,2, \ldots \tilde{\mathrm{N}}$ at random, where $\boldsymbol{w_i} = [w_{i1}, w_{i2}, \ldots, w_{in}]$ is the weight
2. Calculate the hidden layer output matrix **H** as in Eq. (2), where $g(z)$ is an activation function

$$\boldsymbol{H} = g(\boldsymbol{x}.\boldsymbol{w}^T + b) \tag{2}$$

3. Calculate the output weight $\hat{\beta}$, as in Eq. (3)

$$\hat{\beta} = \boldsymbol{H}^{\dagger}\boldsymbol{T} \tag{3}$$

$$\boldsymbol{\beta} = \begin{bmatrix} \beta_1^T \\ \vdots \\ \beta_{\tilde{N}}^T \end{bmatrix}_{\tilde{N}xm} \text{ and } \mathbf{T} = \begin{bmatrix} t_1^T \\ \vdots \\ t_N^T \end{bmatrix}_{Nxm}$$

with value $\mathbf{H}^{\dagger}$ as in the Eq. (4)

$$\mathbf{H}^{\dagger} = (\mathbf{H^T H})^{-\mathbf{1}}\mathbf{H^T} \text{ and } \mathbf{H^T H} \text{ non-singular}$$
$$\mathbf{H}^{\dagger} = \mathbf{H^T}(\mathbf{HH^T})^{-\mathbf{1}} \text{ and } \mathbf{HH^T}\text{non-singular} \tag{4}$$

m is the feature count, Ñ is the number of hidden neurons, $\mathbf{H}^{\dagger}$ is the Moore–Penrose generalized inverse of the matrix **H**, **T** is the target matrix, **H** is the hidden layer output matrix, and $\mathbf{H}^{\mathrm{T}}$ is the hidden layer transpose output matrix.

The target is defined in Eq. (5) as [11-12]

$$\boldsymbol{T_i^k} = \begin{cases} 1 \; if \; c_i = k \\ -1 \; if \; c_i \neq k \end{cases} \qquad k = 1,2, \ldots C \tag{5}$$

where c_i is the label of the class, and C is the number of classes.

The predictive outcome of learning is defined in Eq. (6) as[11-12]

$$y_k = \boldsymbol{H}.\, \beta_k, \mathrm{k} = 1,2, \ldots \mathrm{C}. \tag{6}$$

The output is calculated using Eq. (7) [12]

$$\widehat{C_\iota} = \underset{k=1,2,..C}{arg\, max}\, y_i^k \tag{7}$$

2.4 Activation function

This research will use the activation function of the hyperbolic tangent function (TanH), according to formulas (8) and (9). The TanH has a range [-1,1] that is calculated in Eq. (8)[19-20] as,

$$f(z) = tanh(z) = \frac{2}{1+e^{-2z}} - 1 \tag{8}$$

where on ELM

$$\mathrm{z} = \mathbf{x}.\mathbf{w^T} + \mathrm{b} \tag{9}$$

2.5 Particle swarm optimization (PSO)

Y. Shi and R. Eberhart introduced the PSO algorithm in 1995 [21], which is a random search algorithm that is adopted from the feeding behavior of birds either individually or in groups. In line with other evolutionary algorithms, the PSO generates solutions for some problems at random; each of these solutions is called a particle. Each particle has parameters, namely the current position, velocity, and the current best position found by the particle. The D dimensional vector represents the parameter. The position and velocity of the particles are expressed in terms of [15]:

$$\boldsymbol{x_i} = (x_{i1}, x_{i2}, \dots, x_{iD})$$
$$\boldsymbol{v_i} = (v_{i1}, v_{i2}, \dots, v_{iD})$$

At each iteration, the position and velocity of the particles are updated using Eq. (10) and Eq. (11):

$$\boldsymbol{x_i}(t+1) = \boldsymbol{x_i}(t) + \boldsymbol{v_i}(t+1) \quad (10)$$

$$\boldsymbol{v_i}(t+1) = w\boldsymbol{v_i}(t) + R_1 c_1(\boldsymbol{P_i} - \boldsymbol{x_i}(t)) + R_2 c_2\left(\boldsymbol{P_g} - \boldsymbol{x_i}(t)\right) \quad (11)$$

where w is the inertia weight, R_1 and R_2 are random numbers that differ between 0 and 1, c_1 and c_2 are the acceleration coefficients known as the cognitive and social scaling parameter, $\boldsymbol{P_i}$ is the best previous position of the particle itself, and $\boldsymbol{P_g}$ shows the previous best position of all the particles in the swarm.

2.5.1 Inertia weight

Inertia weight (w) is one of the most important parameters of the PSO algorithm. Adaptive inertia weight is better than constant inertia weight [22]**.** The high value of w makes exploration easier by increasing diversity while a low value of w will only support local exploitation. The balance between global and local searches during the training process is essential for the success of an optimization algorithm. This study will compare five adaptive inertia weights to determine the characteristics of these inertia weights in conducting global and local searches. The inertia weights compared are as follows:

1. Linearly decreasing inertia weight (LDIW)

This model was introduced by H. Zhu, Y. Wang, K. Wang, and Y. Chen in 2011 and is most commonly used (Eq. (12)). It can ensure the algorithm has strong global search capabilities in the early period and strong local search capabilities in the late period [23]:

$$w = w_{max} - \frac{t \times (w_{max} - w_{min})}{t_{max}} \quad (12)$$

Where w_{min} and w_{max} are respectively the minimum and maximum of the weight value of inertia. Usually, for a value w_{max} = 0.9 and value w_{min}= 0.4, t is the current iteration while t_{max} is the maximum iteration.

2. Nonlinear decreasing inertia weight (NDIW)

C. Li and X. Liu introduced this model in 2016 (Eq. (13)). It can avoid being trapped in the local minimum and slow down the convergence compared to the LDIW model [24]:

$$w = w_{max} - \frac{\sqrt{t+1} \times (w_{max} - w_{min})}{\sqrt{t_{max}+1}} \quad (13)$$

3. Linearly varying inertia weight (LVIW)

LVIW was discovered by Y. Shi and R. C. Eberhart (Eq. (14))[25]. In this model, particles tend to continuously increase for local search[26]:

$$w = (w_{max} - w_{min}) \times \frac{(t_{max} - t)}{t_{max}} + w_{min} \quad (14)$$

4. Piecewise-Varied Inertia Weight Piecewise (PIW*)*

Q. Quande, L. Li, and L. Rongjun introduced the PIW in 2010 (Eq. (15)) [27]. This study uses an $\alpha = 0.2$. This is because the average accuracy of the α value is better than the $\alpha = 0.6$, although 0.2 converges slower than 0.6 [27].

$$w = \tan\left(\frac{\pi}{4} \times \left(1 - \frac{t}{t_{max}}\right)^{\alpha}\right) \times (w_{max} - w_{min}) + w_{min} \quad (15)$$

5. Simulated annealing inertia weight (SAIW)

SAIW was introduced by W. Al-Hassan, M.B. Fayekl, S.I. Shaheen in 2006 [28], and W. Han, P. Yang, H. Ren, and J. Sun in 2010 [29] described it (Eq. (16)) as,

$$w = (w_{max} - w_{min}) \times \lambda^{t-1} + w_{min} \quad (16)$$

In his research, the value $\lambda = 0.95$; therefore, Eq. (16) is written as,

Received: September 7, 2020. Revised: October 6, 2020.

$$w = (w_{max} - w_{min}) \times 0.95^{t-1} + w_{min} \quad (17)$$

In this paper, the $\lambda = 0.95$ refers to previous researches which were conducted by W. Al-Hassan, M.B. Fayekl, S.I. Shaheen in 2006 and W. Han, P. Yang, H. Ren, and J. Sun in 2010.

3. Proposed PSO-ELM

The proposed PSO-ELM algorithm was used to optimize the ELM accuracy to classify active compounds based on the SMILES code. Unlike the ELM, which determines the number of hidden neurons by trial and error, the PSO-ELM will automatically search for the best number of hidden neurons to produce good accuracy.

3.1 Proposed inertia weights

The proposed inertia weight is a modification of SAIW, named as Modified SAIW. The change made by set value $\lambda = 0.85$ and λ^{t-1} transform into λ^{t}; therefore, Eq. (16)-(17) was transformed as

$$w = (w_{max} - w_{min}) \times 0.85^{t} + w_{min} \quad (18)$$

The graphs of the reduction of inertia weight ranging from 0.9 to 0.3 of SAIW and Modified SAIW can be seen in Fig. 2. The proposed algorithm (Modified SAIW) decreases faster than the SAIW while the next movement is slower.

The curve shows that the algorithm can find local solutions, which is better than SAIW. At the same time, the particle velocity drops faster so that when the particles are close to the global optimal solution, it can be achieved. This results in the particles not losing their optimal solution [24].

3.2 Parameter analysis

This analysis is used to consider the use of the $\lambda = 0.85$. Consideration of the use of the values is the result of curve analysis in Fig. 3 and performance (i.e accuracy) in Fig. 4. Curve analysis was performed by comparing the λ values in the range (0.6-0.95) and using $t_{max} = 50$. In Fig. 3, the λ =0.85 curve is in the middle position among other λ values. This shows that this value is the most ideal in exploring the search space. Ideal means that the balance between global and local search is the best for SMILES code classification.

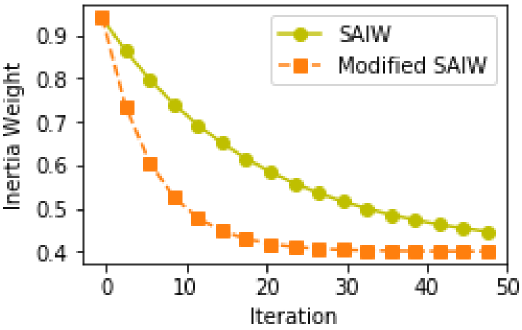

Figure. 2 Graph of reduction in inertia weight

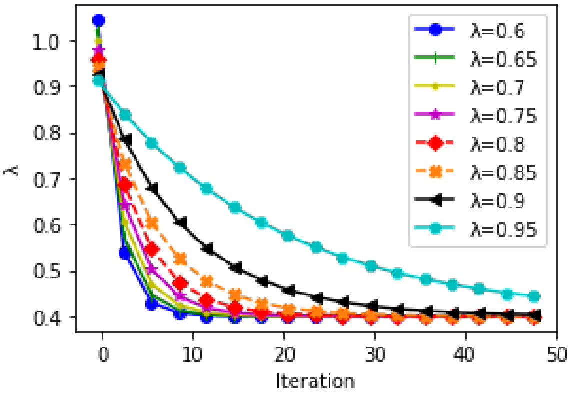

Figure. 3 Variations of λ values

Accuracy tests on various λ values are carried out to know the λ that produces the highest accuracy. The average accuracy of λ can be seen in Fig. 4. Fig. 4 shows that the accuracy of λ = 0.85 is the highest.

3.3 Representation of particles

In this research, integer and real values were used to represent the contents of the particles. Particle representation consists of the number of hidden neurons, weight, and bias. The length of the particle depends on the number of hidden neurons entered, as shown in Eq. (19)

$$L = a + \aleph \times f_i + \text{bk}\tilde{\text{N}} \quad (19)$$

Where a =1 is a storage area for the number of hidden neurons, $\aleph$ is the maximum number of hidden neurons, f_i is the number of features in this study = 29, and $bk\tilde{N}$ is the amount of bias. The representation of the particles can be seen in Table 2.

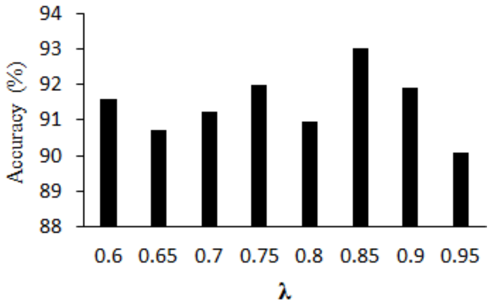

Figure. 4 Comparison of accuracy of variations of λ values

Table 2. Representation of particles

Number of *hidden neurons*	***weight*: feature x *hidden neuron***			***bias***		
Ñ	**w$_{11}$**	**…**	**w$_{f\tilde{N}}$**	**b$_{11}$**	**…**	**b$_{k\tilde{N}}$**

3.4 Definition of fitness

Each particle has a fitness function that can be used to evaluate the performance of the particle. In this study, the fitness function ($f(i)$) (Eq. (20)) is defined as

$$fitness = accuracy = \frac{N_c}{N_t} \tag{20}$$

where N_c is the number of correct test data, and N_t is the total amount of all testing data[18].

3.5 Function classification of active compounds with the PSO-ELM

To be able to classify the PSO-ELM using the SMILES code, several steps are taken, as shown in Fig. 5, are as follows:

1. Perform data *preprocessing*

Data preprocessing consists of data cleaning and feature retrieval.

a. Cleaning of invalid data: The data were obtained from PubChem, drug bank, and Ttdrug; then, synchronization was carried out between the three dataset sources. The data had the same name of the active compound but a different SMILES code; then, the SMILES code from the PubChem database was used as a reference.

b. Feature retrieval. The features were extracted from the SMILES code into 29 features, as shown in Table 1.

2. Conduct training on training data using the PSO and ELM algorithms. The ELM algorithm was used to calculate the fitness value of each particle. The calculation of the fitness value referred to Eq. 20. This training used inertia weights (w) and the acceleration coefficient (c_1 and c_2) as adaptive. The adaptive inertia weights were used in Eq. (18); c_1 was used in Eq. (21) [26] as,

$$c_1 = \left(c_{1f} - c_{1i}\right) \times \frac{current\ iter}{\max iter} + c_{1i} \tag{21}$$

and c_2 was used in Eq. (22) as,

$$c_2 = \left(c_{2f} - c_{2i}\right) \times \frac{current\ iter}{\max iter} + c_{2i} \tag{22}$$

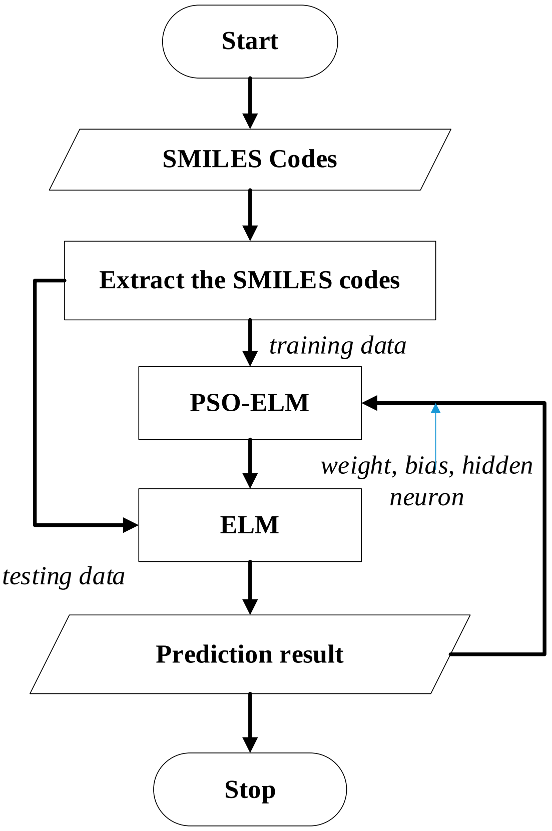

Figure. 5 PSO-ELM classification stages based on the SMILES code

where the values $c_{1i}, c_{1f}, c_{2i},$ and c_{2f} are constant. In this study, the value c_{1i}= c_{2f} = 2.5 and value c_{1f} = c_{2i}= 0.5. Therefore, the value c_1 ranges from 2.5 to 0.5 while the value c_2 from 0.5 to 2.5. In this training phase, the output was obtained in the form of the optimal number of hidden neurons, weight, and bias.

3. The weight, bias, and the number of hidden neurons obtained from step 2 were used to predict the testing data using the ELM algorithm. The performance of the ELM was measured by using Eq. (20). The stages of classification of the function of the active compound can be seen in Fig. 5.

4. Result and discussions

The data were taken from https: //www.pubchem.ncbi.nlm.nih.go and then divided into two groups: 90% as training data and 10% as testing data. The testing data was taken from each class with a proportional number randomly. A total of 1043 data were classified into four classes, i.e., neurological diseases, bacterial infections, inflammation, and cancer, in which each class consisted of 346, 281, 232, and 184, respectively.

According to the research of S. Saraswathi, S. Sundaram, and N. Sundararajan [12], the amount of

Table 3. The amount of data in each class

No	Class	Total number of molecules	Difference
1	Nerves	346	0
2	bacteria	281	18.78613
3	Inflammation	232	32.94798
4	Cancer	184	46.82081
	Total data	1043	

Table 4. Class descriptions

Number of Classes	Dataset
2	Nerves, bacterial infection
3	Nerves, bacterial infection, inflammation
4	Nerves, bacterial infection, inflammation, cancer

unbalanced data can still be tolerated if the difference between the classes is lesser than 40%. In this study, the difference between the neurological class and the bacteria class = 18.8%, the neurological class with inflammation = 33%, and the nerve class with cancer as much as 46.8%, as shown in Table 3.

In this study, testing will be carried out in 2 classes, 3 classes, and 4 classes, as shown in Table 4.

4.1 Parameter

The parameters used to carry out this experiment include the maximum number of iterations: 50, the maximum number of hidden neurons: 200, the Tanh activation function, and the population size (number of particles) used = 25. The number of particles is sufficient based on the research conducted by P. Umapathy, C. Venkataseshaiah, and M. S. Arumugam [30], A. Ratnaweera, S. K. Halgamuge, and H. C. Watson [26] and Y. Shi and R. C. Eberhart[25]. P. Umapathy, C. Venkataseshaiah, and M. S. Arumugam, and C. Li and X. Liu [24] using the number of particles as 20. The weight of inertia used is adaptive using Eqs. (12)-(18); hence, the value starts with w_{max} (0.9) and ends with w_{min} = 0.3. The acceleration coefficient is adaptive, which runs from 2.5 to 0.5 for c_1 and 0.5 to 2.5 for c_2. All these parameters will be used to evaluate the average accuracy, maximum accuracy, standard deviation of accuracy, and training time. Each scenario will be repeated 40 times.

4.2 Average accuracy

In this accuracy test, there are 2 scenarios carried out. First, comparing the performance of ELM and PSO-ELM with different inertia weights. The second compares the Proposed with other machine learning algorithms.

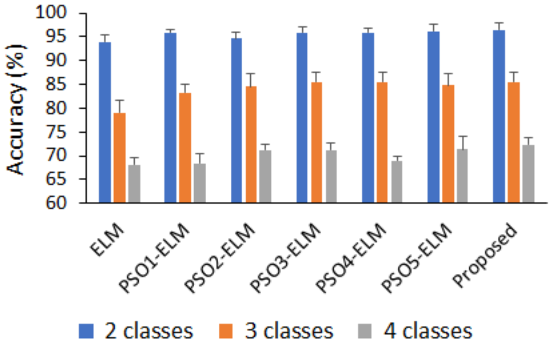

Figure. 6 Comparison of average accuracy of the ELM and PSO-ELM. The classification of compounds into twoclasses show the best accuracy, and the PSO-ELM propose algorithm shows the best accuracy in all classes

4.2.1 Average accuracy of PSO-ELM

The average accuracy was tested using the ELM, and PSO-ELM algorithms using the inertia weights in Eqs. (12) - (18) (Fig. 6). The algorithm labeling was adjusted according to the order of equation number in this article; for example, the LDIW inertia weight is called PSO1, the inertia weight for NDIW is called PSO2, the inertia weight for LVIW is called PSO3, the PIW inertia weight is called PSO4, the inertia weight for SAIW is called PSO5, and the Modified SAIW is called proposed.

The result of this experiment showed that the average accuracy test of the ELM is lower than the PSO-ELM in all classes. This is because the ELM does not have the ability to optimize appropriate weight, bias, and the number of hidden neurons. Further analysis suggested that the PSO using Modified SAIW (proposed) inertia has the highest average accuracy compared to the other PSO-ELM because of the strong propose inertia weight in exploring local optimal search solutions because the movement is very slow to get to a weight of 0.3 (Fig. 2). The proposed algorithm could increase the accuracy of the ELM until 2.54%, 6.43%, and 3.85% for the classes 2, 3, and 4, respectively.

The greater the number of classes used, the lower the accuracy. This is due to the increasing number of comparisons and the t level of difficulty in class determination. These results correspond to T. F. Lesmana, S. M. Isa, and N. Surantha [31], and O. K. Utomo, N. Surantha, S. M. Isa, and B. Soewito[32] showed that the accuracy of research on the testing data decreases in parallel with increase in the number of classes. The lowest accuracy in trials with 4 classes compared to 2 and 3 classes is also due to prediction errors for the cancer class. This is because the number

Received: September 7, 2020. Revised: October 6, 2020.

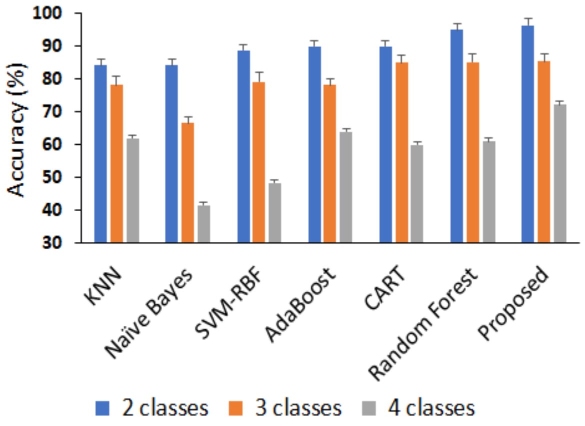

Figure. 7 Comparison of machine learning algorithms accuracy results

of data on the cancer class is the lowest, and the difference in the number of training data on the cancer class with other classes is more than 40%[33]. This lack of training data leads to data pattern recognition (learning conducted in the cancer classes is not accurate).

4.2.2 Comparison of classification accuracy of machine learning algorithms

The purpose of this test scenario is to know the proposed performance compared to other machine learning algorithms. In this scenario, the Proposed is compared with KNN, Naive Bayes, SVM-RBF, AdaBoost, Classification and Regression Tree (CART), and Random Forest. In this experiment using k = 5 and the Manhattan distance for the KNN method. While the Decision Tree and AdaBoost methods use a depth of 4 and n_estimator: 100. The experimental results are shown in Fig. 7.

The proposed has higher accuracy compared to KNN, Gaussian Naïve Bayes, SVM-RBF, AdaBoost, Decision Tree CART (Classification and Regression Tree), and Random Forest for all classes. The highest increase accuracy is in the number of class 4, i.e., Proposed 10.2% higher than KNN, 30.6% higher than Gaussian Naïve Bayes, 23.8% higher than SVM-RBF, 8.3% higher than AdaBoost, 12.3% more high compared to CART and 11.3% higher than random Forest.

4.3 Maximum accuracy

The maximum accuracy is obtained after testing with the same parameters as the previous test and repeated 40 times. The maximum accuracy results are shown in Fig. 8. Overall, the maximum accuracy of the PSO-ELM method is better than the ELM method for the classes 2 and 3. However, the PSO2-ELM and the PSO4-ELM have low maximum accuracy in class 4. The proposed PSO-ELM algorithm has higher accuracy compared with the ELM in all classes, i.e., 3.17% for class 2, 6.9% for class 3, and 2.9% for class 4. The maximum accuracy was achieved by the PSO5-ELM that was almost similar to the proposed algorithm for classes 2 and 3, even higher for class 4. This is because the inertia weight on the PSO5-ELM decreased slowly at the beginning of the iteration, which caused global search exploration.

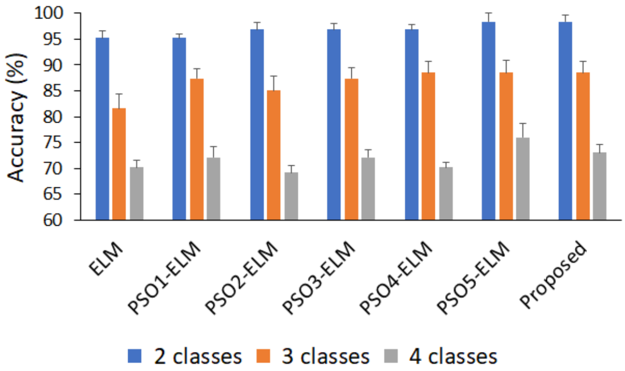

Figure. 8 Comparison of the maximum accuracy of the ELM and PSO-ELM

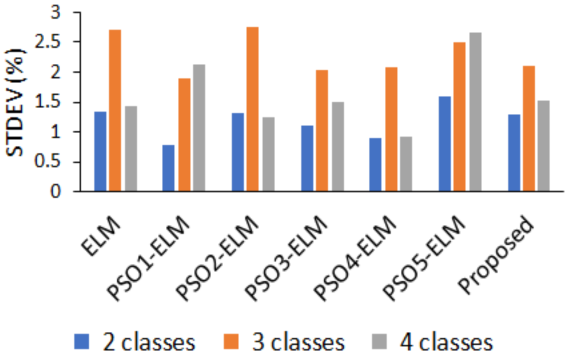

Figure. 9 Standard deviation which indicated the value of accuracy average of ELM and PSO-ELM

4.4 The standard deviation of average accuracy

In simple terms, the standard deviation (STDEV) describes the difference between the sample value and the average value or mean value. The greater the standard deviation value means more spread (varies) of the accuracy value from the average. Conversely, if it is small, the accuracy value will be more homogeneous (almost the same). The comparison of the standard deviation of the average accuracy among the ELM and PSO-ELM is quite low; this data suggests that the accuracy value of the method is close to the average accuracy value (Fig. 9).

The standard deviation in class 2 is the lowest compared to the other classes. Moreover, the accuracy of class 2 is also the best, in which the

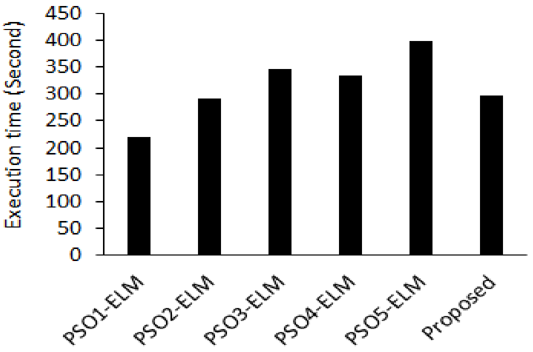

Figure. 10 Computing time computation of PSO-ELM

average accuracy value ranges from 93% (ELM) to 96.5% (proposed algorithm). The standard deviation of the proposed algorithm is smaller than PSO5, which indicates that the accuracy of the proposed algorithm is more convergent with average accuracy.

4.5 Training and testing time

The result from training and testing time indicated that the PSO1-ELM has the shortest execution time compared to the others because of the linear inertia weight of PSO1. The proposed algorithm, which is a modification of PSO5, has a shorter execution time than PSO5 (Fig. 10).

5. Conclusion

The study concluded that the ELM and PSO-ELM can be used to classify the function of active compounds based on the SMILES code with high accuracy. The PSO-ELM can improve the accuracy performance of the ELM classification by optimizing the weight, bias, and the number of hidden neurons automatically. The proposed algorithm has the highest average accuracy and better than the ELM by increasing the accuracy to 2.54%, 6.43%, and 3.85% for 2, 3, and 4 classes, respectively. Moreover, the proposed PSO-ELM was better than the PSO-ELM for average accuracy, computation time, and standard deviation. A comparison with KNN, Gaussian Naïve Bayes, SVM-RBF, AdaBoost, CART, and Random Forest for all classes shows that the proposed method is superior. Future work will focus on preprocessing, namely increasing the number of features that are extracted from the SMILES code. We hope that the increasing number of features, the recognition of the characteristics of the class be better.

Conflicts of Interest

The authors declare that there is no conflict of interest.

Author Contributions

The author's contributions are as follows: “Conceptualization, Dian Eka Ratnawati, Widodo and Syaiful Anam; methodology, Dian Eka Ratnawati; software, Dian Eka Ratnawati; validation, Marjono, and Widodo; formal analysis, Syaiful Anam; investigation, Dian Eka Ratnawati; literature reviews, Dian Eka Ratnawati and Widodo; resources, Widodo; data curation, Widodo; writing—original draft preparation, Dian Eka Ratnawati; writing—review and editing, Marjono, Widodo, and Syaiful Anam; revising, Dian Eka Ratnawati, Widodo and Syaiful Anam; corresponding, Dian Eka Ratnawati; Proofreading, Marjono, Widodo, and Syaiful Anam; visualization, Widodo; supervision, Marjono.

Acknowledgments

This research was supported by Brawijaya University, Malang, Indonesia.

References

[1] D. Weininger, “SMILES, a Chemical Language and Information System”, *J.Chem. Inf.Comput.Sci*, Vol. 28, pp. 31–36, 1988.

[2] A. A. Toropov, A. P. Toropova, S. E. Martyanov, E. Benfenati, G. Gini, D. Leszczynska, and J.Leszczynski, “Comparison of SMILES and Molecular Graphs as the Representation of the Molecular Structure for QSAR Analysis for Mutagenic Potential of Polyaromatic Amines”, *Chemom. Intell. Lab. Syst.*, Vol. 109, No. 1, pp. 94–100, 2011.

[3] H. Öztürk, E. Ozkirimli, and A. Özgür, “A Comparative Study of SMILES-Based Compound Similarity Functions for Drug-Target Interaction Prediction”, *BMC Bioinformatics*, Vol. 17, No. 1, pp. 1–11, 2016.

[4] R. R. W. Tigusti, D. E. Ratnawati, and S. Anam, “Implementasi Fuzzy K-Nearest Neighbor (FK-NN) untuk Mengklasifikasi Fungsi Senyawa Berdasarkan Simplified Molecular Input Line Entry System (SMILES)”, *J. Pengemb. Teknlogi Inf. dan Ilmu Komput.*, Vol. 2, No. 12, pp. 6331–6338, 2018.

[5] M. I. A. Rochman, D. E. Ratnawati, and S. Anam, “Penerapan Algoritme C4 . 5 untuk Klasifikasi Fungsi Senyawa Aktif Menggunakan Kode Simplified Molecular Input Line System (SMILES)”, *J. Pengemb. Teknlogi Inf. dan Ilmu Komput.*, Vol. 3, No. 1, pp. 761–769, 2019.

[6] D. E. Ratnawati, Marjono, and S. Anam, “Prediction of Active Compounds from SMILES Codes Using Backpropagation

Algorithm", In: *Proc. of AIP Conf. Proceedings,* Malang, Indonesia, Vol. 060009, pp. 1–6, 2018.

[7] D. F. Indarwati, D. E. Ratnawati, and S. Anam, "Klasifikasi Fungsi Senyawa Aktif berdasarkan Data Simplified Molecular Input Line Entry System (SMILES) menggunakan Metode Support Vector Machine (SVM)", *J. Pengemb. Teknol. Inf. dan Ilmu Komput.*, Vol. 3, No. 8, pp. 7844–7850, 2019.

[8] G.-B. Huang and C.-K. Siew, "Extreme Learning Machine with Randomly Assigned RBF Kernels", *Int. J. Inf. Technol.*, Vol. 11 No. 1, pp. 16–24, 2004.

[9] G.-B. Huang, Q. Zhu, and C. Siew, "Extreme Learning Machine : Theory and Applications", *Neurocomputing*, Vol. 70, pp. 489–501, 2006.

[10] A. Toprak, "Extreme Learning Machine (ELM)-Based Classification of Benign and Malignant Cells in Breast Cancer", *Med. Sci. Monit.*, pp. 6537–6543, 2018.

[11] R. Ahila, V. Sadasivam, and K. Manimala, "An Integrated PSO for Parameter Determination and Feature Selection of ELM and its Application in Classification of Power System Disturbances", *Appl. Soft Comput.*, Vol. 32, pp. 23–37, 2015.

[12] S. Saraswathi, S. Sundaram, and N. Sundararajan, "ICGA-PSO-ELM Approach for Accurate Multiclass Cancer Classification Resulting in Reduced Gene Sets in which Genes Encoding Secreted Proteins are Highly Represented", *IEEE/ACM Trans. Comput. Biol. Bioinforma.*, Vol. 8, No. 2, pp. 452–463, 2011.

[13] Y. Xue, C. Bai, D. Qiu, F. Kong, and Z. Li, "Predicting Rockburst with Database Using Particle Swarm Optimization and Extreme Learning Machine", *Tunn. Undergr. Sp. Technol.*, Vol. 98, No. 103287, pp. 1–12, 2020.

[14] M. Li, H. Chen, X. Shi, S. Liu, M. Zhang, and S. Lu, "A Multi-Information Fusion Triple Variables with Iteration Inertia Weight PSO Algorithm and its Application", *Appl. Soft Comput. J.*, Vol. 84, 2019.

[15] M. Taherkhani and R. Safabakhsh, "A Novel Stability-Based Adaptive Inertia Weight for Particle Swarm Optimization", *Appl. Soft Comput.*, Vol. 38, pp. 281–295, 2016.

[16] J. Li, W. Lu, H. Wang, Y. Fan, and Z. Chang, "Groundwater contamination source identification based on a hybrid particle swarm optimization-extreme learning machine", *J. Hydrol.*, Vol. 584, No. 124657, 2020.

[17] D. E. Ratnawati, Marjono, Widodo, and S. Anam, "Features Selection for Classification of SMILES Codes Based on Their Function", In: *2019 International Seminar on Research of Information Technology and Intelligent Systems (ISRITI),* Yogyakarta, Indonesia, pp. 103–108, 2019.

[18] F. Dong, J. Liu, L. He, X. Hu, and H. Liu, "Channel Estimation Based on Extreme Learning Machine for High Speed Environments", In: *Proc. of ELM-2015 Volume 1*, Vol. 1, No. I, pp. 159–167, 2015.

[19] J. Pomerat, A. Segev, and R. Datta, "On Neural Network Activation Functions and Optimizers in Relation to Polynomial Regression", In: *Proc. of 2019 IEEE International Conf. on Big Data (Big Data),* Los Angeles, CA, USA, pp. 6183–6185, 2019.

[20] D. E. Ratnawati, Marjono, Widodo, and S. Anam, "Comparison of activation function on extreme learning machine (ELM) performance for classifying the active compound Comparison", In: *Proc. of AIP Conf. Proceedings 2264,* Kuta Bali, Indonesia, Vol. 140001, pp. 1–8, 2020.

[21] Y. Shi and R. Eberhart, "A Modified Particle Swarm Optimizer", In: *Proc. of 1998 IEEE International Conf. on Evolutionary Computation Proceedings. IEEE World Congress on Computational Intelligence (Cat. No.98TH8360),* Anchorage, AK, USA, pp. 69–73, 1998.

[22] L. Wen and Z. Xi, "The Research of PSO Algorithms with Non-Linear Time-Decreasing Inertia Weight", In: *Proc. of the 7th World Congress on Intelligent Control and Automation June*, Chongqing, China, pp. 4002–4005, 2008.

[23] H. Zhu, Y. Wang, K. Wang, and Y. Chen, "Particle Swarm Optimization (PSO) for the constrained portfolio optimization problem", *Expert Syst. Appl.*, Vol. 38, No. 8, pp. 10161–10169, 2011.

[24] C. Li and X. Liu, "An Improved PSO-BP Neural Network and Its Application to Earthquake Prediction", In: *Proc. of 2016 Chinese Control and Decision Conf. (CCDC),* Yinchuan, China, pp. 3434–3438, 2016.

[25] Y. Shi and R. C. Eberhart, "Empirical Study of Particle Swarm Optimization", In: *Proc. of the 1999 Congress on Evolutionary Computation-CEC99 (Cat. No. 99TH8406),* Washington, DC, USA, pp. 1945–1950, 1999.

[26] A. Ratnaweera, S. K. Halgamuge, and H. C. Watson, "Self-Organizing Hierarchical Particle Swarm Optimizer with Time-Varying Acceleration Coefficients", *IEEE Trans. Evol. Comput.*, Vol. 8, No. 3, pp. 240–255, 2004.

[27] Q. Quande, L. Li, and L. Rongjun, “A Novel PSO with Piecewise-Varied Inertia Weight”, In: *Proc. of 2010 2nd IEEE International Conf. on Information and Financial Engineering*, Chongqing, China, pp. 503–506, 2010.

[28] W. Al-Hassan, M. B. Fayekl, S. I. Shaheen, “PSOSA : An Optimized Particle Swarm Technique for Solving the Urban Planning Problem”, In: *Proc. of 2006 International Conf. on Computer Engineering and Systems*, Cairo, Egypt, pp. 401–405, 2006.

[29] W. Han, P. Yang, H. Ren, and J. Sun, “Comparison Study of Several Kinds of Inertia Weights for PSO”, In: *Proc. of 2010 IEEE International Conf. on Progress in Informatics and Computing*, Shanghai, China, No. 09, pp. 280–284, 2010.

[30] P. Umapathy, C. Venkataseshaiah, and M. S. Arumugam, “Particle Swarm Optimization with Various Inertia Weight Variants for Optimal Power Flow Solution”, *Discret. Dyn. Nat. Soc.*, Vol. 2010, 2010.

[31] T. F. Lesmana, S. M. Isa, and N. Surantha, “Sleep Stage Identification Using the Combination of ELM and PSO Based on ECG Signal and HRV”, In: *Proc. of 2018 3rd International Conf. on Computer and Communication Systems*, Nagoya, Japan, pp. 258–262, 2018.

[32] O. K. Utomo, N. Surantha, S. M. Isa, and B. Soewito, “Automatic Sleep Stage Classification using Weighted ELM and PSO on Imbalanced Data from Single Lead ECG”, In: *Procedia Computer Science*, Yogyakarta, Indonesia, Vol. 157, pp. 321–328, 2019.

[33] A. Nagelli, R. Ragala, and B. S. Saleena, “Automotive Rank Based ELM Using Iterative Decomposition”, *International Journal of Intelligent Engineering and Systems*, Vol. 12, No. 5, pp. 287–298, 2019.