OPEN ACCESS
ISSN 2460-9056
socj.telkomuniversity.ac.id/indojc

Ind. Journal on Computing
Vol. 5, Issue. 1, March 2020. pp. 33-44
doi:10.21108/indojc.2020.5.1.372

Eye State Prediction Based on EEG Signal Data Using Neural Network and Evolutionary Algorithm Optimization

Untari N. Wisesty [#1], Hifzi Priabdi [#2], Rita Rismala [#3], Mahmud D. Sulistiyo [#4]

 # School of Computing, Telkom University
Bandung, Indonesia
[1] untarinw@telkomuniversity.ac.id
[2] hifzinistique@gmail.com
[3] ritaris@telkomuniversity.ac.id
[4] mahmuddwis@telkomuniversity.ac.id

Abstract

Eye state prediction is one study using EEG signals to predict the state of human eye based on its previous states. In the development, many researchers have also built eye states detection schemes, but the system built is only limited to classifying a single record of input data obtained from the Emotive EPOC headset channel into the eye state. Therefore, this paper proposes eye state prediction system where the system can immediately predict the human eye state based on the previous sequences of the EEG signals. The proposed system consists of two parts; The first part is the prediction of EEG signal value using the Differential Evolution, while the second part is the eye state detection based on the predicted signal value using the Neural Network optimized by Evolution Strategies. We conducted experiments and obtained the highest accuracy for the eye state prediction task of 73.2%. The result is obtained by the best combination of parameters from the three methods used in this study.

Keywords: eye state prediction, EEG signals, Differential Evolution, Neural Network, Evolution Strategies

Abstrak

Prediksi *eye state* merupakan salah satu studi yang menggunakan sinyal EEG untuk memprediksi keadaan mata manusia berdasarkan kondisi beberapa saat sebelumnya. Dalam perkembangannya, banyak peneliti yang telah membangun skema deteksi *eye state*, tetapi sistem yang dibangun hanya terbatas pada pengklasifikasian berdasarkan satu baris data input yang didapatkan dari saluran headset Emotive EPOC ke dalam *eye state*. Oleh karena itu, paper ini mengusulkan sistem prediksi *eye state*, di mana sistem dapat memprediksi keadaan mata yang akan terjadi berdasarkan rangkaian sinyal EEG. Sistem yang kami usulkan terdiri dari dua bagian; Bagian yang pertama adalah prediksi nilai sinyal EEG menggunakan Differential Evolution, dan bagian kedua adalah deteksi keadaan mata berdasarkan nilai sinyal yang telah diperoleh menggunakan Neural Network yang dioptimasi oleh Evolution Strategies. Eksperimen yang dilakukan menghasilkan akurasi tertinggi untuk prediksi *eye state* sebesar 73.2%. Hasil tersebut diperoleh dengan kombinasi parameter terbaik dari tiga metode yang digunakan pada penelitian ini.

Kata Kunci: prediksi *eye state*, sinyal EEG, Differential Evolution, Neural Network, Evolution Strategies

Received on Jan 2020. Accepted on March 2020

I. INTRODUCTION

IN the modern era, the technology is growing very rapidly, especially in the applications of medical purposes. One of the emerging technologies in the medical field is the electroencephalogram (EEG). In practice, doctors are usually assisted by the EEG in diagnosing a patient's disease based on signals emitted by the brain. EEG is a device that captures *electrical-like* signals formed by the human brain's activity in the form of an analog wave. The analog signals are converted into digital signals to be processed as input features for a computer-based application. The signals are possibly resulted and influenced by various activities carried out by the human body, such as the movements of the hands and feet, the closing and opening of the eyelids, and so on [1]. The results of EEG signals perception can be used for a variety of applications in very broad fields, such as driving tools for people with disability, analyzing person's emotion, controlling computer games, and even for military purposes [2]. However, the noisy EEG signal may result in misperception and hence affect the further analysis. Therefore, accurate classification of EEG signals is required and considered important to help any related application functioning properly.

In the development, many studies as mentioned in [3, 4, 5, 6] have proposed eye state detection schemes. However, they were limited to classifying a single record of input data obtained by the Emotive EPOC headset channel into the eye state. In our previous works, Wisesty [7] used the Levenberg Marquardt algorithm to optimize the Neural Network weights in the eye state detection task; It reached the highest accuracy rate of 98.912%, but it unfortunately requires a very long time in the training process. Prakoso, et al. [8] used the Extreme Learning Machine algorithm to optimize Neural Network weights and it could reduce the training time to yield the accuracy rate of 93.34%; However, it needs a lot of neurons, meaning a complex model, to achieve that performance.

Based on the problems above, in this research, we develop eye state prediction system where the system can predict the state of the human eye based on the previous sequences of the EEG signals. The proposed system consists of two parts, namely the prediction of the EEG signal value and the eye state detection based on the value of the predicted signal. This research uses the Differential Evolution for the EEG signal prediction task prior to the eye detection and the Multi-Layer Perceptron optimized by the Evolution Strategies for the eye state detection task. The Neural Network model classifies the eye state quickly after the training process. The standard ANN uses Backpropagation algorithm in the training process. Backpropagation is sometimes difficult to make a good classification model when it has a low convergence rate as demonstrated in the previous eye state detection study using standard Multi-Layer Perceptron, achieving an accuracy of 56.45% [5]. Therefore, the Evolution Strategies algorithm is employed in this research to help optimize the ANN training process. Evolution Strategies is an optimization algorithm that moves towards convergent conditions through its evolutionary process. Differential Evolution and Evolution Strategies very appropriate for searching task in the domain of real numbers.

The rest of this paper explores literature review, proposed method, and experimental results and discussion in Section II, III, and IV, respectively. It is then closed with conclusion in Section V.

II. LITERATURE REVIEW

Eye state detection is one study using EEG signals using Emotive EPOC tools. Oliver Rösler and David Suendermann make a dataset for eye state classification based on EEG signals [3]. The dataset was obtained by recording the brain signals of a volunteer using an Emotive EPOC headset in a closed room. The volunteer was free to open and close his eyes. When recording, there was a camera in front of the volunteer to get the target class from the recorded data [3, 6]. They applied the standard classifier available on the Weka toolkit; The classifiers were Artificial Neural Network, Naïve Bayes, K-star, and Decision Tree. From the 42 methods used, K-star obtained the lowest error rate of 2.7%. However, there was a fatal weakness of K-star, that was very slow to calculate the detection output, as well as another instance-based learner. In this data, it took up to 38 minutes to get the output label using K-star for entire existing data [3].

One of the existing studies revealed how the EEG changes based on the movement of the human eyelids. It discovered that the largest difference of the state resulted when the eyes are closed and opened is the power of both states. The power when they are closed is larger than when they are opened [4]; The results of that research becomes basis of the study in eye state classification task. Ting Wang, et al. [9] tried to classify the same data using a classifier namely Incremental Neural Network training with Increasing Input Dimension (ITID). The lowest error rate obtained using the classifiers was 27.46%. Sabanci, et al. [5] compared two classification methods to detect the eye state, namely k-Nearest Neighbors (k-NN) and Multilayer Perceptron (MLP); The highest accuracies for k-NN and MLP were 84.05% and 56.45%, respectively. Devipriya, et al. [2] used the Information Gain feature selection to evaluate features that affected the eye state classification results; It found that the F7, P7, O1, O2, F8 and AF4 channels were the most influential attributes. By using the J48 classification method, the accuracy was 80%. Sahu, et al. [10] used various classification methods from the instant-based until the decision tree-based classifier to detect the eye state and measure its performance; The study inferred that the instant-based classifier, namely IBL, obtained the highest accuracy for detecting the eye state using EEG signals data. From these papers, the studies carried out are still limited to either detecting or classifying the eye state at any given time without predicting the value of the EEG signal when volunteers think an action to take, to either open or close their eyes.

III. PROPOSED METHOD

In general, we propose three main processes for the eye state prediction system, namely data preprocessing, EEG value prediction, and eye state detection. In the process of predicting EEG signal values, the Differential Evolution algorithm is used. Whereas, to detect the eye state from the predicted EEG signal value, the Neural Network method is used and its weights are optimized by the Evolutionary Strategies algorithm. The steps of the system are shown in Fig. 1.

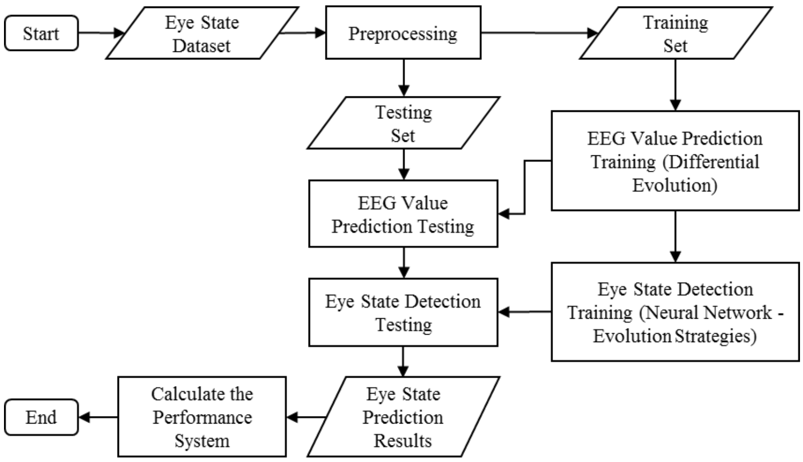

Fig. 1. Diagram of the Eye States Prediction System.

A. Eye State Dataset

The data used in this research is the eye state dataset constructed by Oliver Rösler and David Suendermann, consisting of the eye states when it opens and closes. The dataset collection is supported by a volunteer wearing Emotive EPOC headset in a quiet room. The volunteer is free to open and close his eyes and recorded by a camera to capture whether the volunteer is opening or closing his eyes [3, 6].

TABLE I
Sample of the Eye State Dataset

Input								Output
AF3	F7	F3	FC5	T7	P7	…….	AF4	Eye state
4317.44	3949.23	4191.28	4110.77	4350.77	4615.90	…….	4771.28	0
4317.95	3961.54	4196.92	4133.33	4370.26	4614.87	…….	4784.10	0
4318.97	3962.56	4194.87	4133.33	4371.28	4611.28	…….	4781.54	0
4317.95	3964.62	4193.33	4124.62	4369.74	4613.85	…….	4764.10	0
4452.82	4032.31	4295.38	4130.26	4330.26	4592.31	…….	4549.23	0
4445.13	4017.95	4292.82	4121.54	4325.13	4591.79	…….	4552.82	1
4490.26	4106.67	4350.77	4142.05	4408.21	4612.31	…….	4363.08	1
4485.13	4112.31	4343.59	4142.56	4405.13	4610.77	…….	4360.00	1
4484.62	4124.10	4339.49	4143.08	4394.87	4615.38	…….	4369.23	1
4488.21	4130.77	4344.62	4144.62	4396.41	4618.46	…….	4374.36	1

The experiment was carried out for 2 minutes; The eye state is categorized as close, or labeled as '1', when the eyes are completely closed. The eye state dataset based on the EEG signals is available at the Repository of California University, Irvine (UCI) with URL: https://archive.ics.uci.edu/ml/datasets/EEG+Eye+State. The dataset has 14,977 records and 15 attributes where 14 attributes data are obtained from the values of electroencephalogram sensor attached on the volunteer head (AF3, AF4, F7, F3, F4, F8, FC5, FC6, T7, T8, P7, P8, O1, and O2), and one attribute left is the state of the volunteer eye at the same time based on the video recording using a camera. For the eye states, the value of '1' is when the eye completely closed, and the value '0' is otherwise. Table I shows some examples of the eye state dataset.

B. Data Preprocessing

The data preprocessing conducted include the data cleaning and the data normalization. The data cleaning is necessary because in the eye state dataset, there are attribute values that exceed the normal value limit, so that the data needs to be removed from the dataset. Furthermore, the data will be normalized into the range of 0 to 1 to facilitate the process of classification using the Neural Network. (1) is used to normalize the data [7].

$$X' = \frac{X - X_{min}}{X_{max} - X_{min}} \quad (1)$$

Note that X' is the new value of the feature in the normalization domain, X is the value before normalization, X_{min} is the smallest value of all data in one attribute to be normalized, and X_{max} is the largest value of all data in one attribute to be normalized. Normalized data samples are found in Table II.

TABLE II
Sample of the Eye State Dataset After Normalization Process

Input								Output
AF3	F7	F3	FC5	T7	P7	…….	AF4	Eye state
0.30	0.37	0.27	0.32	0.12	0.20	…….	0.38	0
0.31	0.28	0.35	0.26	0.23	0.28	…….	0.44	1

C. EEG Values Prediction

The EEG values prediction system aims at predicting the value of the EEG signal in N series based on some previous data series. The output of the prediction system becomes the input for the eye state detection (classification) system. The prediction system utilizes the Differential Evolution (DE) algorithm and is performed for each value from attribute 1 to attribute 14. In predicting EEG signal values, the linear regression concept is used, where the optimized parameters are $a_1, \dots, a_n$ and b, as shown in (2). Thus, the individual representation (vector) contains real numbers either positive or negative along with the number of series used.

$$y = a_1 x_1 + a_2 x_2 + \cdots + a_n x_n + b \tag{2}$$

Note that y is the predicted value, $x_1, \dots, x_n$ are the input series, and n is number of input series.

The DE algorithm steps carried out in this research are as follows [11]: The initial population is randomly generated using a uniform distribution. Next, DE generates a new vector (individu) parameter involving three vectors as parents. The parent selection is done with the same probability for each vector regardless of their fitness values. If the new generated vector has fitness value better than any other vector in the population, then this new vector will replace the old vector. The differential mutation scheme used is the DE2 scheme, where the scheme also uses the best vector to generate a new vector. Differential mutation is done by the (3).

$$\underline{v} = \underline{x}_{i,G} + \alpha\left(\underline{x}_{best,G} - \underline{x}_{i,G}\right) + F\left(\underline{x}_{r2,G} - \underline{x}_{r3,G}\right) \tag{3}$$

Note that G is generation, α and F are the control variables to improve the searching direction.

To increase the diversity of vectors in the population, vector v is combined with an arbitrary vector x in the population. This recombination process produces the vector u.

$$u_j = \begin{cases} v_j, \text{ for } j = \langle n \rangle_D, \langle n+1 \rangle_D, \dots, \langle n+L-1 \rangle_D \\ \left(x_{i,G}\right)_j, \text{ otherwise} \end{cases} \tag{4}$$

The symbol $\langle . \rangle_D$ denotes the modulo function with modulus D, D is the vector dimension, n is the starting point for recombination, and L is a random integer with interval $[0, D-1]$.

D. Eye State Detection

The proposed eye state detection system is based on the classification of EEG signals that have 14 input channels using Neural Network method. In Neural Network method, there are two important components to concern, which are the Neural Network architecture and the learning algorithm. First, this research uses Multi-Layer Perception architecture as shown in Fig. 2 with 14 input neurons that represent the input data (as shown at Table I), 1 hidden layer with N hidden neurons, and 2 output neurons, i.e. neuron 0 and 1. Each output neuron represents each class, if the value of neuron 0 is bigger than the neuron 1, then the output is 0, and vice versa.

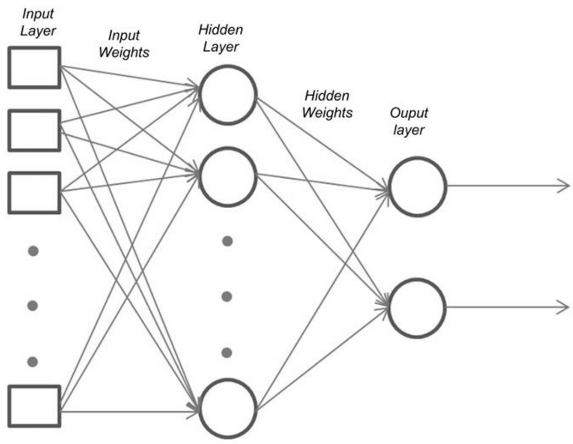

Fig. 2. General Architecture of the Neural Network for the Eye State Detection Task.

Meanwhile, for the learning algorithm that aims at finding the optimal weights between neurons, we use the Evolution Strategies (ES) algorithm. ES will generate chromosomes as many as the population size randomly. Each chromosome is then decoded into the weights for the Neural Network classifier. The classifier uses the weights to proceed forward propagation during the training process and the error is obtained from the difference between the classifier's output and the target data; From that, fitness value of the corresponding chromosome is calculated based on the Mean Squared Error (MSE). Afterwards, the process of parents selection is performed before the evolution operators that include parents selection, crossover, and survivor selection. These processes produce new chromosomes called children. The number of children are approximately 7 times of the initial population size. The survivor selection is done based on the fitness values of all population and children produced; The best chromosome is obtained through this process. Then, we check the stopping condition of the evolution loop based on the number of generations. If the stopping condition is not reached, the process continues to the next loop and perform sequence of processes including the evolution operators; Otherwise, we stop the loop and obtain the model weighted by the best chromosome. Sequence of these processes are illustrated in Fig. 3 as follows.

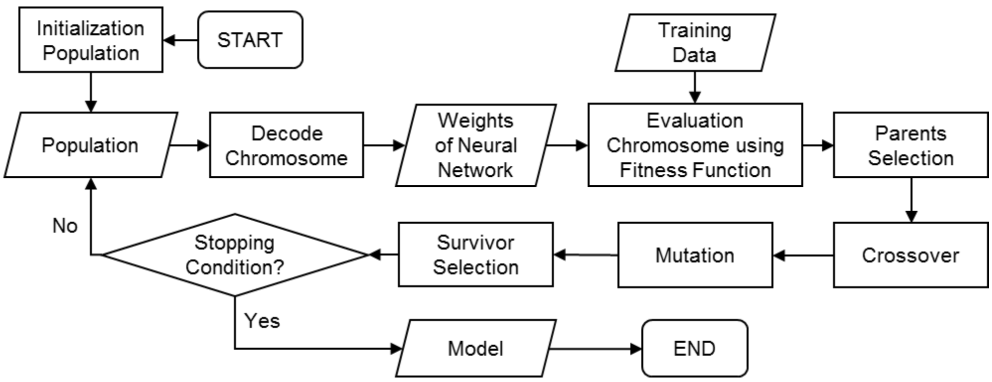

Fig. 3. Evolution Strategies Diagram for Weights Optimization of Neural Network.

In this research, the chromosome representations to be evaluated are **1-mutation step chromosome** and **n-mutation step chromosome**. With 1-mutation step chromosome, there is only one value for the mutation step size to perform the mutation for all chromosome genes. Meanwhile, with n-mutation step chromosome, there are n values of the mutation step sizes, where each is used to mutate the corresponding gene. Fig. 4 and 5 below depict 1-mutation step chromosome and n-mutation step chromosome representations, respectively.

w_{111}	$w_{1..}$	w_{1ij}	w_{211}	$w_{2..}$	w_{2jk}	σ

Fig. 4. Representation of 1-Mutation Step Chromosome.

w_{111}	$w_{1..}$	w_{1ij}	w_{211}	$w_{2..}$	w_{2jk}	σ_1	$\sigma_{..}$	σ_n

Fig. 5. Representation of n-Mutation Step Chromosome.

Note that $w_{1..}$ are weight between the input layer (i neurons) and the hidden layer (j neurons), $w_{2..}$ are weights between the hidden layer (j neurons), and the output layer (k neurons), and σ is the mutation step size.

The decode chromosome is a process of transforming the chromosome into the weights for the Neural Network classifier; Once a chromosome is decoded, the weights are distributed to the feed-forward propagation. Fig. 6 shows the weights resulted by the decode chromosome process distributed for the Neural Network's feed-forward propagation; There are 14 input neurons (x_1, x_2, …, x_n), j hidden neurons (H_1, H_2, …, H_j), and 2 output neurons (o_1 and o_2).

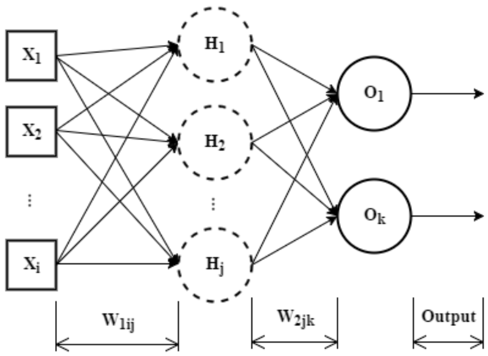

Fig. 6. The Weights Resulted by the Decode Chromosome are Distribute to the Neural Network Classifier.

The fitness function in the Artificial Neural Network (ANN) training to optimize the weights is based on the MSE. The feed-forward propagation only performs one-way calculation from the input layer to the output layer; This process is ended by the error calculation and hence the MSE is also calculated. The weights generated by ES will connect between neurons in the ANN classifier. For each neuron, an activation function is applied to determine output of the neuron; Here, we use the sigmoid-based activation function.

The process of parents selection is performed before the crossover as it requires two chromosomes from the population to produce a new chromosome. The parents selection process uses uniform distribution so that all chromosomes in the population have the same probability of being selected as the parent chromosome [11].

After the parents chromosomes are selected, the crossover probability is checked; If a random value generated is smaller than the crossover probability, then the crossover is performed; Otherwise, it is not performed. The crossover in ES is local intermediary crossover as formulated in (5). z is the child chromosome from the parent x and y. i is the position of the gene being recombined. The result of the crossover in ES is only one child.

$$z_i = \frac{x_i + y_i}{2} \tag{5}$$

In the mutation process, it also has a mutation probability to determine whether the mutation is performed or not. Before entering the mutation process, a random value is generated; If the value is smaller than the mutation probability, then the mutation is performed; Otherwise, it is not performed. The mutation mechanism depends on the chromosome representation. (6) and (7) are used for the mutation with 1-mutation step representation, whereas (8) and (9) are for the mutation with n-mutation step representation.

$$\sigma' = \sigma.\exp\left(\tau.N(0,1)\right) \tag{6}$$
$$x_i' = x_i + \sigma'.N(0,1) \tag{7}$$
$$\sigma_i' = \sigma.\exp(\eta.N(0,1) + \tau.N_i(0,1)) \tag{8}$$
$$x_i' = x_i + \sigma'_i.N(0,1) \tag{9}$$

In the 1-mutation step, σ' is resulted by the mutated σ, x_i' is resulted by the i-th mutated x, and τ is set close to $\frac{1}{\sqrt{n}}$. As for the n-mutation step, η is the learning rate for all genes, τ is the learning rate for each gene position, $N(0,1)$ is a random numbers from 0 to 1, η is set close to $\frac{1}{\sqrt{2n}}$, n is the number of genes apart from mutation step size, and τ is set close to $\frac{1}{\sqrt[4]{2n}}$ [11].

The number of all children chromosomes is usually more than the initial population size; Therefore, the selective pressure available in the ES is high [11]. The survivor selection method uses (μ, λ)-selection; it selects the survived population for the next generation from the children chromosomes that are produced.

IV. Experimental Results And Discussion

We build the eye state prediction system that includes predicting the value of the EEG signal based on the previous series of signals and classifying the eye state based on the predicted EEG signal. The EEG prediction task uses the Differential Evolution method with the DE2 mutase scheme. In this scheme, there are several parameters used; The population size is 10 times the dimensions of chromosomes; The range of F values is from 0.0001 to 1 and adaptive with the changes at the F value of 0.02; The number of data series is from 1 to 5; And the crossover probability is 0.8. The EEG prediction task predicts the value of each attribute (including the 14 input attributes) so that it results in 14 prediction models. Table III shows the best prediction model for each attribute including its performance as our experimental results. In the rightmost column, X1 to X5 indicates the data series used. In these experiments, we distributed the training, validation, and testing sets with a proportion of 70%, 15%, and 15%, respectively, over all records in the dataset.

TABLE III
Best EEG Prediction Models for the 14 Attributes

Attribute	MAPE (%)	Best Chromosome Result
1	0.50821	$0.024157 - 0.016097X_1 + 0.051199X_2 + 0.31278X_3 - 0.37613X_4 + 0.99988X_5$
2	1.32010	$0.0031669 + 0.069173X_1 - 0.17237X_2 + 0.6073X_3 - 0.5285X_4 + 0.99781X_5$
3	1.75120	$0.0051483 + 0.96095X_1$
4	0.55183	$0.030266 + 0.96417X_1$
5	0.53194	$0.049254 + 0.20828X_1 - 0.25563X_2 + 0.64521X_3 - 0.58458X_4 + 0.92937X_5$
6	0.54163	$0.01828 + 0.10958X_1 - 0.07789X_2 + 0.59598X_3 - 0.63366X_4 + 0.98437X_5$
7	0.90792	$0.013942 + 0.19697X_1 - 0.27727X_2 + 0.642X_3 - 0.58034X_4 + 1X_5$
8	1.76900	$0.0096622 + 0.17641X_1 - 0.25905X_2 + 0.5232X_3 - 0.52512X_4 + 1X_5$
9	4.84480	$0.01913 + 0.20096X_1 - 0.31653X_2 + 0.73621X_3 - 0.72229X_4 + 0.99846X_5$
10	1.57030	$0.009125 + 0.0041556X_1 + 0.078462X_2 + 0.26698X_3 - 0.41889X_4 + 0.99566X_5$
11	2.07670	$0.0058074 + 0.18274X_1 - 0.26246X_2 + 0.58051X_3 - 0.5346X_4 + 1X_5$
12	1.57410	$0.0048019 + 0.19008X_1 - 0.20618X_2 + 0.49016X_3 - 0.51317X_4 + 1X_5$
13	0.49468	$0.018327 + 0.066282X_1 - 0.0014222X_2 + 0.43068X_3 - 0.49727X_4 + 0.98045X_5$
14	0.56304	$0.01696 + 0.20047X_1 - 0.20916X_2 + 0.60611X_3 - 0.61764X_4 + 1X_5$

Furthermore, the prediction results of the EEG values will be used for the eye state detection task. The eye state detection system uses ANN classifier where the weight values are trained using the ES algorithm. The ES algorithm has several parameters that can affect the system performance. In the experiments, observations were conducted on the ES parameters as follows.

a. Analyzing the influence of chromosome representation used to the system performance; The chromosome representations for the ES algorithm include 1-mutation step and n-mutation step chromosomes.
b. Analyzing the effect of population size or the number of chromosomes to the system performance; The population sizes to observe include 5, 50, and 100.
c. Analyzing the effect of crossover and mutation probabilities to the learning process carried out by the ES; The crossover and mutation probabilities to observe are 0.3 and 0.7.
d. Analyzing the effect of the number of hidden neurons in the ANN classifier to the system performance; The number of hidden neurons to observe are 1, 6, 12, and 18.

Mutation step and n-mutation step chromosome representations in ES are influential to the mutation process. In the 1-mutation step representation, all genes on the chromosome have the same learning rate; Whereas for the n-mutation step representation, each gene has its own learning rate. However, the number of learning rates reflects the number of genes or the chromosome length. The n-mutation step chromosome representation has more genes than the 1-mutation step chromosome representation, and hence makes the evolution-based searching more complex. Therefore, in the experiments, finding a solution using the 1-mutation step representation is better than using the n-mutation step representation; These are shown in Table IV.

TABLE IV
TESTING ACCURACIES WITH DIFFERENT CHROMOSOME REPRESENTATIONS IN THE ES ALGORITHM

Models Chromosome	Testing Accuracy
1-mutation step	**73.2%**
n-mutation step	66.8%

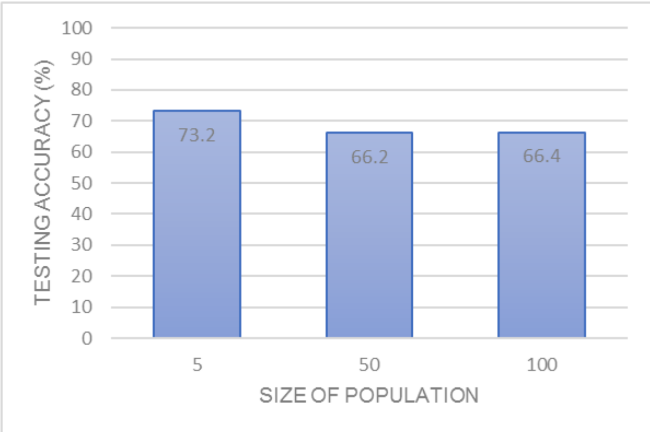

Fig. 7. Testing Accuracies with Various Population Sizes in the ES Algorithm.

Fig. 7 shows the testing accuracies using 3 different values for the population size. According to those values, when the population size increases, the testing accuracy slightly gets decreased. The increase in population size is influential towards the selection of parent chromosomes from the existing chromosomes in the population. From the selected parent chromosomes, we perform crossover and mutation processes given the crossover and mutation probabilities. The selection of these parent chromosomes uses normal distribution, so that all existing chromosomes in the population have the equal opportunity to be chosen as the parent chromosomes. Therefore, it is reasonable that the more chromosomes existed in the population, the lower the probability of choosing chromosomes with high fitness values as the parents. The parents selection affects which children chromosomes to survive for the next generation since we use (μ, λ)-selection as the method for the survivor selection. The survivor selection with (μ, λ)-selection method will only choose the survivors from the children chromosomes to survive for the next generation. Therefore, the large population will cause a small opportunity for the chromosomes that have high fitness values to survive for the next generation compared to the small population.

TABLE V
TESTING ACCURACIES WITH VARIOUS CROSSOVER AND MUTATION PROBABILITIES IN THE ES ALGORITHM

Probability	Testing Accuracy	
	Crossover	Mutase
0.3	**73.2%**	69.2%
0.7	72.2%	**73.2%**

For the probability of crossover, if the crossover process is done infrequently due to the small probability value, the weights generated by the ES for the system can produce better accuracy. It can be seen in Table V, the crossover probability of 0.3 yields higher accuracy than that of 0.7. It is, however, contradictory with the probability of mutation; The mutation probability of 0.7 yields higher accuracy than that of 0.3. The crossover and mutation probabilities determine whether the crossover or mutation process will be performed. The crossover method takes the average gene values of the parents for corresponding gene on the child chromosome;

While the mutation method possibly changes all genes on the parent chromosome based on the mutation step size. It can be inferred that performing the mutation in ES is considered more significant to find the best solution than performing the crossover.

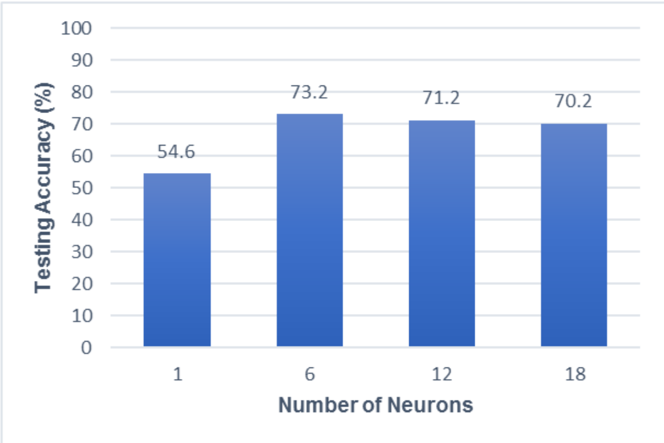

Fig. 8. Testing Accuracies with Various Numbers of Hidden Neurons.

The number of neurons in the ANN architecture affects the number of weights that store information for the ANN classification. However, the increasing number of neurons in the hidden layer is not necessarily make the ANN classifier providing the optimal results. This was proved by Wisesty [7] that conducted experiments on the ANN architecture using one hidden layer and tested the number of neurons from 1 neuron to 50 neurons. In the experiments, the optimal number of neurons, which was 35, used the sigmoid activation function. However, in this paper, the optimum number of neurons is 6, as shown in Fig. 8. When the number of neurons is greater than 6, the testing accuracy slightly decreases although the decrement is not too large compared to the use of 1 neuron. Remember that the more hidden neurons used, the longer it takes to find the optimal results.

The results of the optimum ES parameters combination indicate that ES has better convergence rate than the standard Backpropagation algorithm. It is shown on the error rate of 26.8% yielded by the ANN trained using ES which is smaller than the ANN trained by Oliver Rösler in [3]. This is due to the ES that finds the optimum weights gradually through an evolution process. However, for the solution space of 10,000, it is insufficient for the ES to find the optimum weights since the ES does not always perform the evolution of the weights. It can be seen in the less significant difference between the accuracy of ANN classifier trained using ES (73.2%) with the accuracy of ANN trained by Oliver Rösler (less than 70%) in [3]. These results are below the experiments conducted by Untari [7] and Prakoso [8] which achieve accuracies more than 90%. However, the studies conducted by Oliver Rösler [3], Untari [7], and Prakoso [8] are still limited to building an eye state detection system, where the system only detect the current eye state, without predicting the future eye state.

V. Conclusion

Based on several experiments we conducted, it can be concluded that the Differential Evolution and the Evolutionary Strategies can predict eye state quite well. The highest accuracy obtained from the eye state prediction system in this study is 73.2%. The result is yielded by the best combination of parameters from the three methods used in the experiments. The parameters used for the Differential Evolution include the population size of 10 multiplied by the dimensions of chromosomes, the F values ranging from 0.0001 to 1 that are adaptive with changes in the F value of 0.02, the number of series used from 1 to 5, and the crossover probability of 0.8. Meanwhile, the parameters used for the Evolution Strategies are the hidden neuron of 6, the population size of 5, the crossover probability of 0.3, the mutation probability of 0.7, and the chromosome representation with 1 mutation step size. In the parameters tested against ES, the chromosome representation will affect the mutation process in finding the solution. The number of chromosomes in the population influences the selection of parent chromosomes and the survived chromosomes for the next generation; If the population is small, then the possible parent chromosomes with high fitness have greater chance to be selected and survive for the next generation. Based on the crossover and mutation probabilities, it can be concluded that the mutation process in the ES is more important than the crossover process.

ACKNOWLEDGMENT

The authors would like to thank to the School of Computing, Telkom University, for providing facilities related to the data processing devices and any other technical support to this research.

REFERENCES

[1] N. Jain, S. Bhargava, S. Shivani, and D. Goyal, “Eye State Prediction Using EEG by Supervised Learning”, International Journal of Science, Engineering, and Technology”, 2015.
[2] A. Devipriya, N. Nagarajan, and D. Brindha, “Expert System based Machine Learning Techniques for Eye State Prediction”, International Journal of Pure and Applied Mathematics, Vol.110, No.12, pp.1173—1186, 2018.
[3] O. Rösler and D. Suendermann, “A First Step Towards Eye State Prediction using EEG”, in Proceeding International Conference on Applied Informatics for Health and Life Sciences, 2013.
[4] L. Li, L. Xiao, and L. Chen, “Differences of EEG between Eyes-Open and Eyes-Closed States Based on Autoregressive Method”, Journal of Electronic Science and Technology, Vol.7, No.2, pp.175—179, 2009.
[5] K. Sabanci and M. Koklu, “The Classification of Eye State by Using KNN and MLP Classification Model According to the EEG Signal”, International Journal of Intelligent System and Applications in Engineering, Vol.3, No.4, pp.127—130, 2015.
[6] O. Roesler, L. Bader, J. Forster, Y. Hayashi, S. Heßler, and D. Suendermann-Oeft, “Comparison of EEG Devices for Eye State Classification”, in Proceeding International Conference on Applied Informatics for Health and Life Sciences, 2014.
[7] U.N. Wisesty, “Levenberg-Marquardt Neural Network for Eye States Detection Based on Electroencephalography Data”, International Journal on Information and Communication Technology, Vol.2, No.1, pp.23—36, 2016.
[8] E.C. Prakoso, U.N. Wisesty, and Jondri. “Klasifikasi Keadaan Mata Berdasarkan sinyal EEG menggunakan Extreme Learning Machines”, Indonesian Journal on Computing, Vol.1, No.2, pp.97—116, 2016.
[9] T. Wang, S.U. Guan, K.L. Man, and T.O. Ting, “Time Series Classification for EEG Eye State Identification based on Incremental Attribute Learning”, in Proceeding International Symposium on Computer, Consumer and Control, pp.158—161, 2014.
[10] M. Sahu, N.K. Nagwani, S. Verma, and S. Shirke, “Performance Evaluation of Difference Classifier for Eye State Prediction using EEG Signal”, International Journal of Knowledge Engineering, Vol.1, No.2, pp.141—145, 2015.
[11] Suyanto, “Evolutionary Computation. Komputasi Berbasis ‘Evolusi’ dan ‘Genetika’”, Bandung: Informatika, 2008.