

STUDY COMPARISON BACKPROPAGATION, SUPPORT VECTOR MACHINE, AND EXTREME LEARNING MACHINE FOR BIOINFORMATICS DATA

Umi Mahdiyah¹, M. Isa Irawan¹, and Elly Matul Imah²

¹Faculty of Mathematics and Science, Institut Teknologi Sepuluh Nopember, Jl. Arief Rahman Hakim, Surabaya, 60111, Indonesia

² Mathematics Department, Universitas Negeri Surabaya, Jl. Ketintang, Surabaya, 60231, Indonesia

E-mail: umi13@mhs.matematika.its.ac.id

Abstract

A successful understanding on how to make computers learn would open up many new uses of computers and new levels of competence and customization. A detailed understanding on information-processing algorithms for machine learning might lead to a better understanding of human learning abilities and disabilities. There are many type of machine learning that we know, which includes Backpropagation (BP), Extreme Learning Machine (ELM), and Support Vector Machine (SVM). This research uses five data that have several characteristics. The result of this research is all the three investigated models offer comparable classification accuracies. This research has three type conclusions, the best performance in accuracy is BP, the best performance in stability is SVM and the best performance in CPU time is ELM for bioinformatics data.

Keywords: *Machine Learning, Backpropagation, Extreme Learning Machine, Support Vector Machine, Bioinformatics*

Abstrak

Keberhasilan pemahaman tentang bagaimana membuat komputer belajar akan membuka banyak manfaat baru dari komputer. Sebuah pemahaman yang rinci tentang algoritma pengolahan informasi untuk pembelajaran mesin dapat membuat pemahaman yang sebaik kemampuan belajar manusia. Banyak jenis pembelajaran mesin yang kita tahu, beberapa diantaranya adalah *Backpropagation* (BP), *Extreme Learning Machine* (ELM), dan *Support Vector Machine* (SVM). Penelitian ini menggunakan lima data yang memiliki beberapa karakteristik. Hasil penelitian ini, dari ketiga model yang diamati memberikan akurasi klasifikasi yang sebanding. Penelitian ini memiliki tiga kesimpulan, yang terbaik dalam akurasi adalah BP, yang terbaik dalam stabilitas adalah SVM dan CPU time terbaik adalah ELM untuk data bioinformatika.

Kata Kunci: *Machine Learning, Backpropagation, Extreme Learning Machine, Support Vector Machine, Bioinformatika*

1. Introduction

A successful understanding of how to make computers learn would open up many new uses of computers and new levels of competence and customization. And a detailed understanding of information-processing algorithms for machine learning might lead to a better understanding of human learning abilities (and disabilities) as well [1]. Many type of machine learning that we know, some of them are Backpropagation (BP), Extreme Learning Machine (ELM), and Support Vector Machine (SVM).

First Machine Learning is backpropagation. Backpropagation was initially formulated by Werbos in 1974, which was later modified by Rumelhart and McClelland [2]. Backpropagation is the

gradient descent type algorithm, which has connection parameter for each step or iteration. But, this algorithm can provide harmony result between "network capability" to recognize the patterns which used for training and "network capability" to respond correctly to the input patterns that similar (but not equal) to the training pattern. Then, Backpropagation algorithm has limit classification accuracy, because if the output value is different from the target value, an error will be calculated, and then taken from the output layer to the input layer (Backpropagation process).

Until now, many researchers have developed and implemented the BP algorithm [3-7]. Implementation BP for classification was implemented in most problem as bioinformatics, biomedical, chemistry, art, environment, etc. Several research

use this algorithm because BP has minimum error and effective for some problem, especially in this study for classification problem.

The next powerful machine learning is Support Vector Machines (SVM). Support vector machines (SVM) are a group of supervised learning methods that can be applied to classification or regression. This Machine learning is different from Machine Learning in ANN. In SVM when training process, SVM isn't training all data, but this algorithm just training support vector data. Based idea from this algorithm is optimization margin hyper plane. Although SVM need big memory and need long time to process data, but many researcher use SVM to solve some problem [8-11], because this machine learning have high performance.

Extreme Learning Machine is one of a new learning algorithm in neural networks, which has the Single-hidden Layer feed-forward Network (SLFN). ELM has a very fast learning capability and training small error [12]. First ELM was introduced by Huang in 2004 as Single-hidden Layer Feed-forward Network. ELM was made to overcome the weaknesses of the feed-forward neural networks problem that learning speed. Traditionally, feed-forward neural network using gradient-based learning algorithm for training, as well as all the parameters (input weight and hidden bias) are determined by iterative network, to solve that problem, Extreme Learning Machine using minimum norm least-squares (LS) solution of SLFNs.

Unlike the traditional function approximation theories which require to adjusted input weights and hidden layer biases, input weights and hidden layer biases can be randomly assigned if only the activation function is infinitely differentiable [13]. So, ELM can be faster than prior the neural network algorithm previously. ELM has been applied and developed in various fields [14-18]. Up until now this algorithm is developed, the main reason many research use this algorithm because this algorithm is simple and faster than several algorithm in ANN.

In this paper, we will compare three algorithm for classification several bioinformatics data that have some criteria. Dataset was taken from UCI Machine Learning. We use general BP, SVM using linear kernel and general ELM. We choose Backpropagation (BP), Extreme Learning Machine (ELM), and Support Vector Machine (SVM) because that all machine learning have good performance for classification [2-18]. The rest of the paper is organized as follows. In Section 2, we present the detail of BP, SVM, and ELM. Following that, Section 3 provides experimental results and discussions. Finally, we draw conclusions in Section 4.

2. Methods

Backpropagation

Backpropagation is one of many supervised machine learning. Backpropagation is the gradient descent type algorithm. This algorithm has two phases for processing data. First phase, input vector given to input layer, continued to hidden layer then finding output value in output layer. Second phase, if the output value is different from the target value, an error will be calculated, and then taken from the output layer to the input layer (Backpropagation process) [19].

In BP, transfer function must fulfill several conditions: continue differentiable, not descending function. In this paper we use sigmoid binary function as defined in equation(1) and its derivation as in equation(2).

$$f(x) = \frac{1}{1 + e^{-x}} \quad (1)$$

$$f'(x) = f(x)(1 - f(x)) \quad (2)$$

Backpropagation training follow these step [20]: a) Initialize neuron's weight with random number.

Forward propagation

b) Each input layer neuron receives the inputs and passes it to the connected hidden layer's neuron. c) Compute output value from hidden layer z_j , $j = 1, 2, \dots, n$, with the weight is v_{ji} that connected input layer and hidden layer using equation(3) and equation(4).

$$z_{net_j} = v_{j0} + \sum_{i=1}^p x_i v_{ji} \quad (3)$$

$$z_j = f(z_{net_j}) = \frac{1}{1 + e^{-z_{net_j}}} \quad (4)$$

If the hidden layer end, we continue to output layer. d) Compute all output values from output layer as equation(5) and equation(6).

$$y_{net_k} = w_{k0} + \sum_{i=1}^p x_i w_{ki} \quad (5)$$

$$y_k = f(y_{net_k}) = \frac{1}{1 + e^{-y_{net_k}}} \quad (6)$$

Backpropagation

e) Compute the output layer's error factor based on the error in each of the output layer's neuron using equation(7) and equation(8).

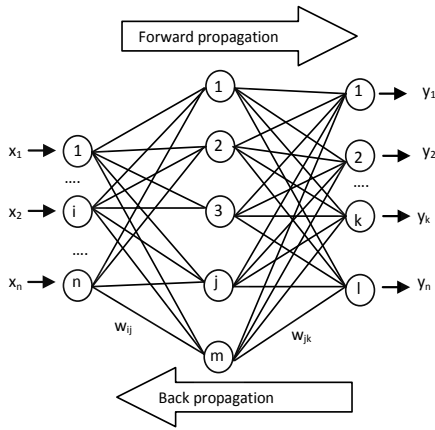


Figure 1. Backpropagation architecture.

$$\delta_{y_k} = (t_k - y_k) f'(y_{net_k}) \quad (7)$$

$$\Delta w_{kj} = \alpha \delta_{y_k} z_j \quad (8)$$

f) Compute the hidden layer's error factor based on error in each of the hidden layer's neuron using equation(9) to equation(11).

$$\delta_{z_{net_j}} = \sum_{k=1}^m \delta_{y_k} w_{kj} \quad (9)$$

$$\delta_{z_j} = \delta_{z_{net_j}} f'(\delta_{z_{net_j}}) \quad (10)$$

$$\Delta v_{ji} = \alpha \delta_{z_j} x_i \quad (11)$$

g) The last step count all change of weight of neurons using equation(12) and equation(13).

$$w_{kj}(new) = w_{kj}(old) + \Delta w_{kj} \quad (12)$$

$$v_{ji}(new) = v_{ji}(old) + \Delta v_{ji} \quad (13)$$

Figure 1 shows the architecture of Back-propagation Algorithm [19]. This architecture has two processes, i.e. forward propagation and back propagation. That is all was explained before.

Support Vector Machine

Support vector machines (SVM) are a group of supervised learning methods that can be applied to classification or regression. Originally, SVM is a training algorithm for linear classification. For non-linear case, SVM maps data sets of input space into a higher dimensional feature space, which is linear and the large-margin learning algorithm is then applied, the mapping can be done by kernel functions. Because, in the high dimensional feature space, that have maximal margin between the

 TABLE 1
 KERNEL FUNCTION

Kernel	Function
Linear	$K(x, y) = x \cdot y$
Polynomial	$K(x, y) = (x \cdot y + v)^d$
Gaussian RBF	$K(x, y) = \exp\left(\frac{-\ x - y\ ^2}{2 \cdot \sigma^2}\right)$
Sigmoid(tangent hyperbolic)	$K(x, y) = \tanh(\sigma(x \cdot y) + c)$
Multiquadratic Invers	$K(x, y) = \frac{1}{\sqrt{\ x - y\ ^2 + c^2}}$

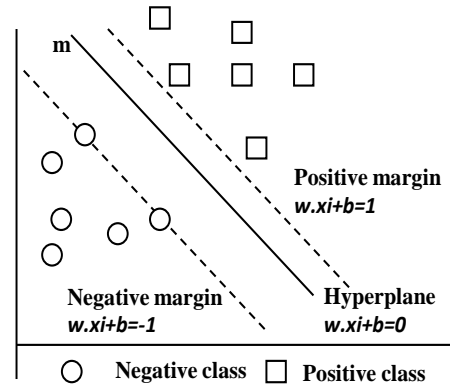


Figure 2. Margin hyperplane SVM.

classes can be obtained, that called linear hyper plane classifiers [21].

This is step of support vector machine algorithm: Given training data set D and q feature of each data, there are Φ , such that formulated in equation(14) and equation(15).

$$\Phi: D^q \rightarrow D^r \quad (14)$$

$$x \rightarrow \Phi(x) \quad (15)$$

Where r is new feature set that result from mapping D . while x is training data, which $x_1, x_2, \dots, x_n \in D^q$ is dataset that mapped to r dimension. The training data is defined using equation(16).

$$(\Phi(x_1), y_1, \dots, \Phi(x_n), y_n) \in D^q \quad (16)$$

Mapping process in SVM needs dot product operation, which denoted as $\Phi(x_i) \cdot \Phi(x_i)$. We can compute it without knowing the transform function of Φ . This computation technique is called kernel trick. Many type of kernel trick in SVM is shown in Table 1.

All explanation in Table 1 can be resumed in illustration of hyper plane SVM. Figure 2 shows an illustration of margin hyper plane in SVM: SVM has positive margin, hyper plane and negative margin.

Extreme Learning Machine

Extreme learning machine is a group of supervised learning methods too as BP and SVM. This algorithm is one of a new learning algorithm in neural networks, which has the Single-hidden Layer feed-forward Network (SLFN). ELM has a simple algorithm, very fast learning capability and training small error [12]. First ELM was introduced by Huang in 2004 as Single-hidden Layer Feed-forward Network (SLFNs). ELM made to overcome the weaknesses of the feed-forward neural networks problem that learning speed. Traditionally, feed-forward neural network using gradient-based learning algorithm for training, as well as all the parameters (input weight and hidden bias) are determined by iterative network, to solve that problem, Extreme Learning Machine using minimum norm least-squares (LS) solution of SLFNs.

A standard single layer feedforward neural network with n hidden neurons and activation function g(x) can be mathematically modeled as equation(17) and equation(18).

$$\sum_{i=1}^{\tilde{N}} \beta_i g_i(x_j) = \sum_{i=1}^{\tilde{N}} \beta_i g(w_i \cdot x_j + b_i) = o_j, \quad (17)$$

$$j \in [1, N] \quad (18)$$

Which $w_i = [w_{i1}, w_{i2}, \dots, w_{in}]^T$ is input weight. b_i is bias hidden layer, $\beta_i = [\beta_{i1}, \beta_{i2}, \dots, \beta_{in}]^T$ is out-put weight, and $w_i \cdot x_j$ is inner product from w_i , x_j and o_j output from this algorithm.

Standard SLFN with \tilde{N} hidden nodes and activation function g(x) can approximate these N samples with zero error means that if $\sum_{i=1}^{\tilde{N}} \|o_j - t_j\| = 0$, there exist β_i , w_i dan b_i such that shown by equation(19) [12].

$$\sum_{i=1}^{\tilde{N}} \beta_i g(w_i \cdot x_j + b_i) = t_j, j \in [1, N] \quad (19)$$

That equation(19) can be write as equation(20).

$$H\beta_i = T \quad (20)$$

Note that H is output matrix hidden layer. Thus,

$$H = \begin{bmatrix} g(w_1 \cdot x_1 + b_1) & \dots & g(w_{\tilde{N}} \cdot x_1 + b_{\tilde{N}}) \\ \vdots & \ddots & \vdots \\ g(w_1 \cdot x_N + b_1) & \dots & g(w_{\tilde{N}} \cdot x_N + b_{\tilde{N}}) \end{bmatrix}$$

ELM algorithm is derived from the minimum norm least squares solution SLFNs.

$$\beta = \begin{bmatrix} \beta_1^T \\ \vdots \\ \beta_{\tilde{N}}^T \end{bmatrix} \quad T = \begin{bmatrix} t_1^T \\ \vdots \\ t_N^T \end{bmatrix}$$

Although, ELM is "generalized" of SLFN but hidden layer (feature mapping) of the ELM does not need to be tuned. Main concepts of ELM as presented in the journal Huang (2006), as follow there:

Given training set

$$\aleph = \{(x_i, t_i) | x_i \in \mathbf{R}^n, t_i \in \mathbf{R}^m, i \in [1, N]\},$$

activation function $g(x)$, and the number of hidden nodes \tilde{N} . Step 1: Insert random weights and biases w_i and b_i , $i \in [1, N]$; Step 2: Calculate the hidden layer output matrix H ; Step 3: Calculate the output weights $\beta = H^\dagger T$ Which $T = [t_1, \dots, t_N]^T$.

H^\dagger is Moore–Penrose Generalized Inverse. A matrix $A \in M_{n \times m}(\mathbb{C})$. There exist a unique $A^\dagger \in M_{m \times n}(\mathbb{C})$, that called Moore–Penrose Generalized Inverse if a) $AA^\dagger A = A$; b) $A^\dagger AA^\dagger = A^\dagger$; c) $AA^\dagger \in H_n$; d) $A^\dagger A \in H_m$.

Then Activation function in the ELM must be infinitely differential (i.e sigmoid function, RBF, sine, cosine, exponential, etc.). Many hidden node depend on the number of training samples is N ($\tilde{N} \leq N$).

Several methods can be used to compute the Moore-Penrose Generalized Inverse of H, which are orthogonal projection, orthogonalization method, iterative method, and singular value decomposition (SVD).

This procedure of research in this study will be presented in the flow chart of research as follows in the Figure 3.

The first step in this research is preprocessing data, in this case used Z_{score} normalization. The formulation Z_{score} as the following equation (21).

$$Z_{score} = \frac{x_i - \bar{x}}{\sigma(x)} \quad (21)$$

Note that $x_i = \{x_1, x_2, \dots, x_n\} \in R$; \bar{x} = mean of data; $\sigma(x)$ = standard deviations. After pre-processing data (normalization) then conducted training data, the training data in this case using 80% data from existing datasets. After the training process is complete, further testing which uses 20% of the data. In this case, sigmoid activation function is

TABLE 2
TABLE OF CONFUSION MATRIX

		Actual	
		True	False
Prediction	True	TP (True Positive)	FP (False Positive)
	False	FN (False Negative)	TN (True Negative)

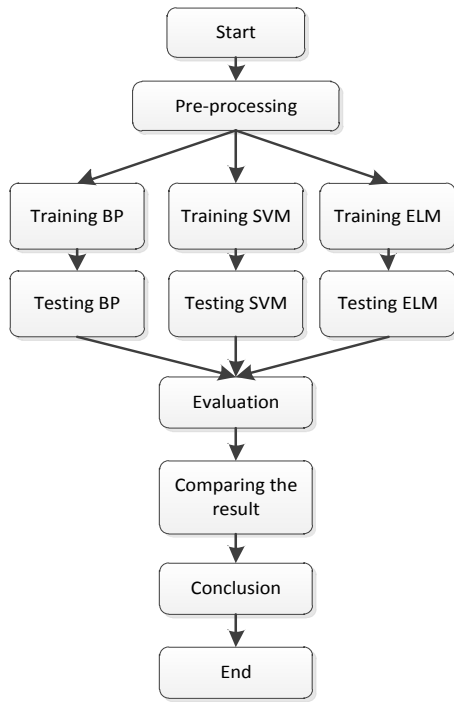


Figure 3. Scheme of methodology.

used as given by equation(22). Threshold in this research use 0.5.

$$f(x) = \frac{1}{1 + e^{-x}} \quad (22)$$

This research will compare Backpropagation algorithm, Support Vector Machine using linear kernel, and standard Extreme Learning Machine. They have been compared using six classification dataset, which is all binary classification.

The datasets in this case is taken from the UCI Machine Learning Repository. From the six selected dataset there are some attributes missing data, so that the pieces of data that have lost attributes was deleted.

Accuracy, precision, recall, CPU time, and misclassification data were used as evaluation measure to compare that three algorithm. Because the dataset uses not only balance data but also use imbalanced data. That formulation or accuracy, precision and recall as the following equation(23) to equation(25).

$$Accuracy = \frac{TP+TN}{total\ of\ sample} \quad (23)$$

$$Precision = \frac{TP}{TP+FP} \quad (24)$$

$$recall = \frac{TP}{TP+FN} \quad (25)$$

TABLE 3
DATA SET SUMMARY FOR CLASSIFICATION
PERFORMANCE COMPARISON CATEGORIES BASED ON
DATA VOLUME

Dataset	Data size	Input feature	+ data	- data	Data type
Small data					
Breast cancer	683	3	444	239	ID
Parkison	195	23	147	48	ID
Pima Indians Diabetes (PID)	768	8	268	500	ID
Promoter	106	58	53	53	BD
ig data					
QSAR biodegradation (QSAR)		1055	41	356	699 ID

ID = imbalanced data; BD= balanced data

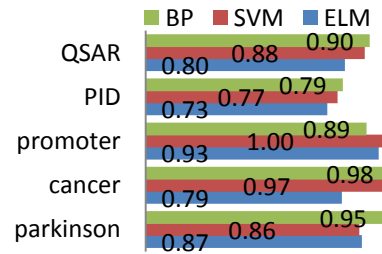


Figure 4. Diagram of accuracy.

The dataset has been split into two categories based on data volume [21] and balancing data [22]. All simulation have been carried out in MATLAB 2012b environment running in an AMD E-350 Processor 1,60GHz.

3. Results and Analysis

In this section we will discuss the results of this research, that is a comparison of the performance of BP, SVM, and ELM, including accuracy, precision, recall, CPU time and misclassification. First, we analyze the result of accuracy, the result of accuracy shown in the Figure 4.

From above data we can show the accuracy of ELM and SVM are not dependent on a small or big data but on balancing the data, whereas BP algorithm always achieves the highest accuracy. But in precision data SVM and ELM have precision higher than BP, that is accuracy for SVM and ELM is 1 and 0.93 then BP is 0.89. If we calculate accuracy of BP, SVM, and ELM for all data is 0.90 ± 0.072 , 0.90 ± 0.093 , 0.82 ± 0.079 in succession.

Second, we analyze the precision of each algorithm and each data. From Figure 5 we can show that precision is high if the data is balanced data, because the system or machine learns that

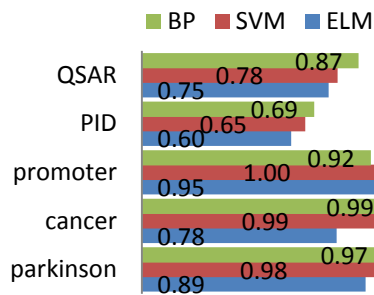


Figure 5. Diagram of precision.

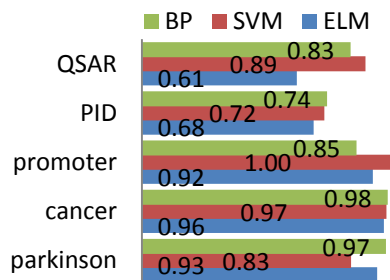


Figure 6. Diagram of recall.

negative and positive data is balance, which means a high precision in promoter data, precision of BP, SVM and ELM is 0.92, 1, and 0.9, then 0.95, 0.90, and 0.94, then 1, 0.97, and 0.86. If we calculate precision of BP, SVM, and ELM for all data is 0.89 ± 0.16 , 0.88 ± 0.16 , 0.79 ± 0.14 in succession.

Then, we analyze recall. That result of recall (see Figure 6). Recall, accuracy, and precision in BP and ELM not only depend on data volume or balancing volume, but also dependent on determine of threshold, in this result we use 0.5 for all data. If we calculate recall of BP, SVM, and ELM for all data is 0.87 ± 0.12 , 0.88 ± 0.11 and 0.82 ± 0.16 in succession.

We can show in three Figure above, accuracy, precision, and recall of BP in promoter dataset always above ELM and SVM. This occur phenomenon because SVM and ELM have best performance in balance dataset. Because promoter dataset have balanced characteristic, so ELM and SVM can surpass performance of BP.

The last, we show the diagram of CPU time and number misclassification data for each data and each algorithm. Result of CPU time and the number of misclassification data shown in the following diagram.

Figure 7 shows the CPU performance. The CPU time of Parkinson data using BP method is faster than CPU time in promoter data, however the Parkinson data size is bigger than promoter data and the data type of promoter data is balan-

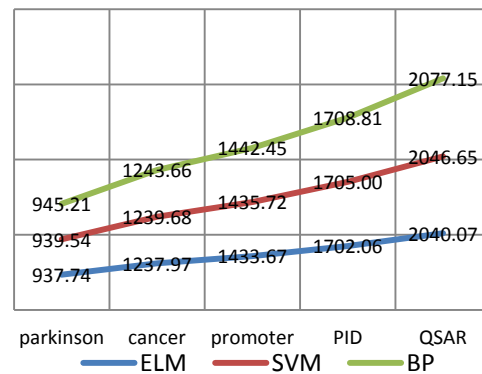


Figure 7. Diagram of CPU time.

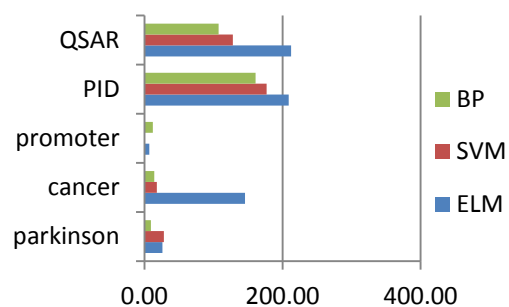


Figure 8. Diagram of the number of misclassification.

ced data. That could be happened because attribute of promoter data is more than Parkinson data.

The result of CPU time is in seconds. From the above results it can be seen that the CPU time of the ELM and SVM is almost always the same as the average difference of them is 5 second, while the CPU time of the BP is always high, especially for large data and imbalance.

Then, the most of misclassification data is in the large data and imbalance data, because the system more learns a lot of majority data than the minority data. Not only characteristic of data but also determine of threshold also have influence of misclassification data.

4. Conclusion

This paper discusses and compares three classification model and seven data that have two characteristic. All the four investigated models offer comparable classification accuracies. ELM has a good performance in balance data, so if we use ELM for imbalanced data must be conditioned so that the data used balance, using under-sampling or other. SVM has a very good performance but requires considerable memory in the training process. BP always has the best perform in accuracy, precision, and recall, the performance has a weak-

ness at the time of computation. So we have three type conclusion, the best perform of accuracy is BP, the best perform of stability is SVM and the best perform of CPU time is ELM.

References

- [1] T.M. Mitchell, *Machine Learning*, McGraw-Hill, 1997.
- [2] M.K. Alsmadi, K.B. Omar, S.A. Noah, & I. Almarashdah, "Performance Comparison of Multi-layer Perceptron (Back Propagation, Delta Rule and Perceptron) algorithms in Neural Networks", *IEEE International Advance Computing Conference*, pp. 296-299, 2009.
- [3] K. Rudd, G.D. Muro, & S. Ferrari, "A Constrained Backpropagation Approach for the Adaptive Solution of Partial Differential Equations", *Neural Networks and Learning Systems*, vol. 25, pp. 571-584, 2014.
- [4] A.H. Kridalaksana, E. Arriyanti, M.I.Ukkas, "Recognition of a human behavior pattern in paper rock scissor game using backpropagation artificial neural network method", *Information and Communication Technology (IC-ICT)*, pp. 238 – 243, 2014.
- [5] A.O. Ibrahim, S.M. Shamsuddin, N.B. Ahmad, and M.N. Salleh, "Hybrid NSGA-II of Three-Term Backpropagation network for multiclass classification problems", *Computer and Information Sciences (ICCOINS)*, pp. 1-6, 2014.
- [6] N. Suciati, W.A. Pratomo, D. Purwitasari, "Batik Motif Classification Using Color-Texture-Based Feature Extraction and Backpropagation Neural Network", *Advanced Applied Informatics*, pp. 517–521, 2014.
- [7] J.S. Bhalla, A. Aggarwal, "A Novel Method for Medical Disease Diagnosis Using Artificial Neural Networks Based On Backpropagation Algorithm", *The Next Generation Information Technology Summit*, pp. 55-61, 2013.
- [8] F.C. Er, G.H. Hatay, E. Okeer, M. Yildirim, "Classification of phosphorus magnetic resonance spectroscopic imaging of brain tumors using support vector machine and logistic regression at 3T", *Engineering in Medicine and Biology Society*, pp. 2392 - 2395 , 2014.
- [9] Y. Shuang, K.K. Tan, B. L. Sng, L. Shengjin, "Feature Extraction and Classification for Ultrasound Images of Lumbar Spine with Support Vector Machine", *Engineering in Medicine and Biology Society*, pp. 4659 – 4662, 2014.
- [10] J. Manikandan & V.K. Agrawal, "Digital Circuit Design Using Support Vector Machines", *Communication Systems and Network Technologies*, pp. 1114 – 1118, 2014.
- [11] G.Y. Wong, F.H.F. Leung, & S.H. Ling, "Predicting Protein-Ligand Binding Site Using Support Vector Machine with Protein Properties", *Transactions on Computational Biology and Bioinformatics*, vol. 10, No. 10, pp. 1517-1529, 2013.
- [12] G.B. Huang, Q.Y. Zhu, & C.K. Siew, "Extreme learning machine: Theory and applications", *Neuro-computing*, vol. 70, pp. 489–501, 2006.
- [13] G.B. Huang, Q.Y. Zhu, & C.K. Siew, "Extreme Learning Machine: A New Learning Scheme of Feed-Forward Neural Network", *Neuro-computing*, vol. 40, pp. 7803-8359, 2004.
- [14] X. Luo, F. Liu, S. Yang, X. Wang, & Z. Zhou, "Joint sparse regularization based Sparse Semi-Supervised Extreme Learning Machine (S3ELM) for classification", *Neuro-computing*, vol. 73, pp. 149-160, 2015.
- [15] H.G. Han, L.D. Wang, & J.F. Qiao, "Hierarchical extreme learning machine for feed-forward neural network", *Neuro-computing*, vol 128, pp. 128-135, 2014.
- [16] G.B. Huang, "An Insight into Extreme Learning Machines: Random Neurons, Random Features and Kernels", pp. 376-390, 2014.
- [17] J. Xin, Z. Wang, L. Qu, & G. Wang, "Elastic extreme learning machine for big data classification", *Neuro-computing*, vol. 149, pp. 464-471, 2015.
- [18] W. Maoa, S. Zhao, X. Mu, & H. Wang, "Multi-dimensional extreme learning machine", *Neuro-computing*, vol. 146, pp. 160-170, 2015.
- [19] E. Prasetyo, *Data Mining: Konsep dan Aplikasi Menggunakan MATLAB*, Andi, Yogyakarta, p. 88, 2012.
- [20] J.J. Siang, *Jaringan Saraf Tiruan dan Pemrogramannya Menggunakan MATLAB*. Andi Offset, Yogyakarta, p. 102-103, 2006.
- [21] Yi Wang, Yi Li, Momiao Xiong, Li Jin, "Random Bits Regression: a Strong General Predictor for Big Data", *Chinese Journal of Computer*, 2005.
- [22] G. Menardi & N. Torelli, "Training and Assessing Classification rules with Imbalanced Data", *Data Mining and Knowledge Discovery*, vol. 28, pp. 92-122, 2014.