Journal of information and communication convergence engineering
http://jicce.org

J. Inf. Commun. Converg. Eng. 17(1): 14-20, Mar. 2019 Regular paper

Enhancing Gene Expression Classification of Support Vector Machines with Generative Adversarial Networks

Phuoc-Hai Huynh[1*], Van Hoa Nguyen[1], and Thanh-Nghi Do[2,3]

[1]Information Technology Faculty, An Giang University, An Giang 90000, Vietnam
[2]College of Information Technology, Can Tho University, Can Tho 92100, Vietnam
[3]UMI UMMISCO 209 (IRD/UPMC)

Abstract

Currently, microarray gene expression data take advantage of the sufficient classification of cancers, which addresses the problems relating to cancer causes and treatment regimens. However, the sample size of gene expression data is often restricted, because the price of microarray technology on studies in humans is high. We propose enhancing the gene expression classification of support vector machines with generative adversarial networks (GAN-SVMs). A GAN that generates new data from original training datasets was implemented. The GAN was used in conjunction with nonlinear SVMs that efficiently classify gene expression data. Numerical test results on 20 low-sample-size and very high-dimensional microarray gene expression datasets from the Kent Ridge Biomedical and Array Expression repositories indicate that the model is more accurate than state-of-the-art classifying models.

Index Terms: Classification, Support vector machines, Generative adversarial networks, Enhancing data, Gene expression data

I. INTRODUCTION

Cancer is one of the most dangerous diseases around the world today. According to data from the World Health Organization, the total number of cancer patients rose to 18.1 million new cases and 9.6 million cancer deaths in 2018 [1]. Gene expression data take advantage of the sufficient classification of cancers and become effective tools in gene discovery, disease diagnosis, and treatment support. Therefore, this technology has been applied to build a comprehensive database of gene expression differences. However, one of the challenges in the gene expression classification is how to cope with low-sample-size datasets [2], especially when using classification models that need labeled data and a large sample size. Increasing sample sizes generates a new gene signature from original datasets that improves the accuracy of classification models [3].

Many machine-learning approaches are applied to classify tumors and diseases based on gene expression data. The support vector machine (SVM) [4] has been proposed to perform gene expression data classification in previous research [5, 6]. The artificial neural network model has been used to predict cancers based on gene expression profiling [7]. A previous study [8] proposed to use the *k*-nearest-neighbor (*k*NN) algorithm for classifying with colon and leukemia datasets. Gene expression data have been classified by decision tree (DT) C4.5 proposed in prior work [9]. In addition, the random-forest (RF) algorithm [10] was applied to classify microarray gene expression data [11]. An RF of oblique DTs effectively classified high-dimensional gene expression data in previous work [12]. In addition, other ensemble methods, such as bagging [13] and AdaBoost [14], have been used

Received 06 November 2018, **Revised** 26 February 2019, **Accepted** 26 February 2019
***Corresponding Author** Phuoc-Hai Huynh (E-mail: hphai@agu.edu.vn, Tel: +84-91-8939068)
An Giang University, No. 14 Ung Van Khiem Road, An Giang 90000, Viet Nam.

Open Access https://doi.org/10.6109/jicce.2019.17.1.14 print ISSN: 2234-8255 online ISSN: 2234-8883

[15, 16]. Recently, many researchers have used deep-learning approaches for gene expression classification. The deep convolutional neural network (DCNN) model has been used to extract features and to classify gene expression [17]. One approach uses a sample enlargement method that combines a stacked autoencoder and convolutional neural network [18]. However, a large sample size is necessary to train an effective classification model, but the sample size of expression data is limited. In practice, collecting a large number of data in classification gene expression is infeasible. For this reason, data enhancement methods have been used to improve the accuracy of classification models in many studies.

The generative adversarial network (GAN) [19] is a deep neural network that learns from some known training data to generate synthetic data similar to the training ones. This model has not only been successfully applied to image data but also to text, video, and medical data [20]. However, the application of GANs in the field of classifying gene expression data is rare. A GAN can learn features from unlabeled microarray data [18]. In addition, GANs can also be used to analyze RNA-Sequencing (RNA-Seq) gene expression data [21]. Therefore, the aim here is to use the GAN model to generate gene expression data. To the authors' knowledge, this approach has not been investigated for gene expression data.

In this work, an accuracy approach is proposed for the precise classification of gene expression data (SVM with generative adversarial network, or GAN-SVM). The GAN-SVM trains GANs to generate new training data, following which the nonlinear SVM learns to classify gene expression data efficiently. Results for 20 low-sample-size and very high-dimensional microarray gene expression datasets from the Kent Ridge Biomedical [22] and ArrayExpress repositories [23] illustrate that the proposed GAN-SVM is more accurate than the state-of-the-art classifying models, including linear SVMs (LSVMs), *k* nearest neighbors (*k*NN), DTs, and RFs.

This work consists of four sections. Section II gives a brief overview of GANs, SVMs, and the proposed GAN-SVM. Section III shows the experimental results, and the conclusions are presented in the final section.

II. METHODS

A. GANs

A GAN [19] is a deep-neural-network architecture consisting of two neural networks: a generator network (denoted by G) and a discriminator network (denoted by D). The aim is to train the G, which generates new samples that are indistinguishable from the data distribution. The D is optimized to distinguish samples from the real data distribution P_{data} from those of the generated data distribution p_g. The G takes vector noise $z \sim p_z$ as input networks and generates samples $G(z)$ with distribution p_g. The generated data samples generated by model G are then sent to the D to determine their similarity with original training data. GAN optimization finds a Nash equilibrium [19] between the G and D. Training a GAN can be formulated as the following mini-max objective function:

$$\min_G \max_D E(x)\sim P_{data}(x)[\log D(x)] + E_{z\sim P_z}[\log(1 - D(G(z)))]. \quad (1)$$

After the success of GANs, they have been widely used in many studies to generate data [20]. The GAN has been used to generate image [24, 25], text [26], and musical data. In addition, Ofir et al. generated language data using a GAN [27]. In other works, researchers built a GAN model to generate resolution natural images [28]. Recently, there have been several applications of the GAN in bioinformatics, such as [29, 30]. GANs have been used to solve the problem of limited data by enhancing synthetic data.

To the authors' knowledge, applications of GANs to increase the classification accuracy of gene expression data are scarce. In the classifying model, a successful prediction system requires a good amount of quality data. Therefore, the aim here is to use the GAN and SVM algorithm for gene expression data classification. The low-sample-size problem of gene expression data classification is solved by generating new data to enlarge gene expression datasets.

B. SVMs

The original SVM algorithm was invented by V. Vapnik [4]. This approach is systematic and properly motivated by the statistical learning theory. SVM is a supervised learning model and has been widely applied to classification problems and regression [31].

The object of the SVM algorithm is to find the optimal hyperplane (the best separating plane furthest from both class +1 and class -1). The separating hyperplane is a plane such that datapoints on one side will be labeled y_i = +1, while datapoints of other class are labeled as y_i = -1. To achieve this purpose, the SVM tries to maximize the distance between two boundary hyperplanes to reduce the probability of misclassification. The optimal hyperplane found by SVM is maximally distant from the two classes of labeled points located on each side (Fig. 1). The most popular approaches for multiclass classifiers commonly used in SVM are the one-versus-all [32] and one-versus-one [33] approaches.

In addition to performing linear classification, the SVM has been very successful in building highly nonlinear classifiers by means of kernel-based learning methods [34]. Kernel-based learning methods aim to transform the input space into higher dimensions, such as a radial basis function (RBF), sigmoid function, and polynomial function.

In practice, the SVM model gives good accuracy in classi-

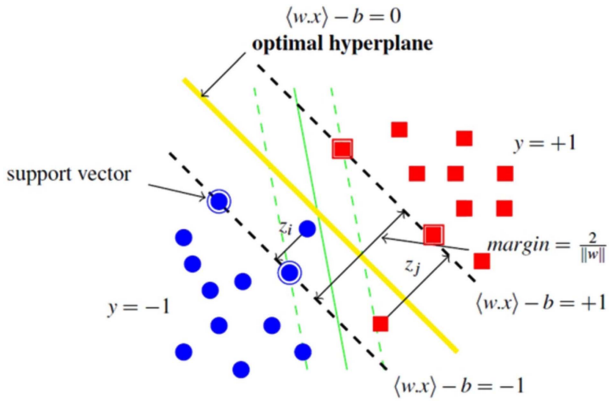

Fig. 1. SVM for binary classification.

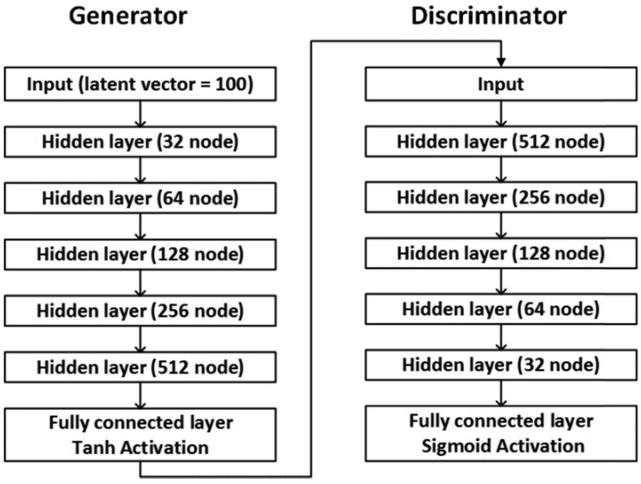

Fig. 2. Architecture of a generative adversarial network.

fying low-sample-size and very high-dimensional data domains. Previous studies [5, 6, 35] have reported classifying gene expression where the SVM is directly trained on the original high-dimensional input spaces. The SVM algorithm has been employed and compared with other classifiers, like *k*NN and DTs [35]. The test results show that the SVM outperforms the traditional algorithms. In view of what has been mentioned so far, one can suppose that the SVM effectively classifies gene expression data. In the proposed approach, a non-linear SVM with an RBF kernel is used for classifying gene expression after these datasets are enlarged by the GAN.

C. GANs and SVMs for the Gene Expression Data Classification

Although the SVM is well-known as an efficient model for classifying very high-dimensional gene expression datasets, the low-sample-size training datasets degrade the classification performance of any model. To overcome this situation, it is proposed to train a GAN model from original datasets to generate new samples for enlarging the training datasets, following which the nonlinear SVM learns to classify gene expression data.

The GAN architecture in this approach has two deep-neural-network models: a generator G model and discriminator D model (Fig. 2).

The generator G takes a noise vector from 100 random numbers to draw from a uniform distribution as an input player. The output of G is a vector gene expression. The network architecture consists of five hidden layers with the following layer sizes: 32, 64, 128, 256, and 512. The Tanh activation function is used at the output layer.

The discriminator network D has a typical neural-network architecture that takes the input data of a vector gene expression. D consists of five hidden layers with sizes 512, 256, 128, 64, and 32. The sigmoid activation function is used at the output layer.

We use batch normalization for generator and discriminator networks. It works by normalizing the input features of a layer to have zero mean and unit variance [36]. In addition, the model uses leaky rectified linear unit (ReLU) activations in the discriminator networks. Leaky ReLU makes it possible to pass a small gradient signal for negative values. Therefore, it makes the gradients from the discriminator flows stronger in the generator. Instead of passing a gradient of zero in the back-prop pass, it passes a small negative gradient. The Adam optimizer has been used for all networks (learning rate of $\eta = 0.0002$ and decay rates of $\beta = 0.5$).

III. EVALUATION

The GAN-SVM was implemented in Python using Tensor Flow [38] and Scikit library [39]. Three algorithms — *k*NN, DT C4.5, and RF — in the Scikit library and the highly efficient standard LSVM [40] were used as baselines. All experiments were done on a NVIDIA GeForce 1050 graphics card with 2 GB of GPU memory. The Student's test was used to assess the classification results of the learning algorithms.

A. Experimental Setup

Experiments were conducted with 20 low-sample-size and high-dimensional microarray gene expression datasets from the Kent Ridge Biomedical [22] and ArrayExpress [23] repositories. The characteristics of datasets are presented in Table 1.

The evaluation protocol was tenfold cross-validation. The RF algorithm learned 200 DTs for classifying all datasets. *k*NN tried to use *k* among {1, 3; 5; 7}. The $C = 10^5$ (a tradeoff between the margin size and the errors) was used for 20 data-

Table 1. Description characterizes of 20 datasets

ID	Name	#Samples	#Dim	#Classes	Ref
1	CNS	60	7129	2	[22]
2	COLON	62	2000	2	[22]
3	DLBCL	47	4026	2	[22]
4	DLBCL_SHIPP	58	7129	2	[22]
5	E-GEOD-10072	107	22283	2	[23]
6	E-GEOD-13911	69	54675	2	[23]
7	E-GEOD-20711	90	54675	5	[23]
8	E-GEOD-25136	79	22283	2	[23]
9	E-GEOD-29354	53	22215	3	[23]
10	E-GEOD-31189	92	54675	2	[23]
11	E-GEOD-36771	107	54675	2	[23]
12	E-GEOD-36895	76	54675	14	[23]
13	E-GEOD-3726	52	22283	2	[23]
14	E-GEOD-37364	94	54675	4	[23]
15	E-GEOD-51024	96	54675	2	[23]
16	E-GEOD-62452	130	33297	2	[23]
17	E-GEOD-63270	104	18989	9	[23]
18	E-GEOD-63885	101	54675	4	[23]
19	E-GEOD-65106	59	33297	3	[23]
20	E-GEOD-66533	58	54675	3	[23]

Table 2. Hyper-parameters of GAN-SVM

ID	Samples generated	C	γ	ID	Samples generated	C	γ
1	50	1E+04	1E-03	11	200	1E+02	2E-05
2	100	1E+02	5E-04	12	50	1E+02	2E-05
3	50	1E+02	2E-04	13	50	1E+02	2E-05
4	200	1E+02	1E-04	14	200	1E+02	4E-05
5	50	1E+02	1E-04	15	100	1E+02	2E-05
6	100	1E+04	2E-05	16	50	1E+02	2E-05
7	100	1E+02	2E-05	17	100	1E+02	3E-05
8	100	1E+02	4E-05	18	100	1E+02	5E-05
9	300	1E+04	1E-04	19	200	1E+04	2E-05
10	50	1E+04	1E-03	20	300	1E+02	3E-05

sets for the LSVM. The total classification accuracy measure was used to classify the results of the learning algorithms.

The GAN-SVM parameters included the number of samples generated by the GAN and the number of epochs. An attempt was made to tune the epoch parameter from 50 to 100 to find the best experiment results. Furthermore, it was also attempted to tune the cost constant of LSVM for GAN-SVM to obtain good accuracy. Then, the LSVM used C = 10^5 for the set label for the generated data. Finally, an attempt was made to tune parameters C and γ of the RBF kernel to obtain good accuracy for the nonlinear SVM. Table 2 shows the best parameters after tuning.

Table 3. Classification results on 20 datasets

ID	*k*NN	C45	RF	LSVM	GAN *k*NN	GAN C45	GAN RF	GAN LSVM	GAN SVM
1	56.48	64.71	60.05	68.48	56.48	65.29	66.71	66.81	70.14
2	75.95	78.81	82.62	80.71	75.95	80.71	72.86	84.29	85.71
3	70.00	77.50	97.50	86.67	65.83	88.83	91.33	88.33	88.33
4	49.48	60.81	57.10	56.24	47.81	65.86	42.57	56.14	59.00
5	50.58	46.84	61.46	55.4	50.58	54.19	57.54	57.22	57.22
6	85.65	90.18	97.14	97.14	87.08	86.85	94.23	97.14	98.57
7	53.38	64.89	74.23	67.34	53.38	63.15	73.84	67.34	72.39
8	58.21	55.54	72.14	65.71	58.21	53.04	63.39	65.71	66.96
9	61.33	63.50	72.17	77.17	61.33	71.00	72.17	77.17	77.5
10	56.22	46.78	56.22	68.22	56.22	54.33	57.22	68.11	68.33
11	81.14	85.78	89.00	89.00	81.14	89.82	84.16	89.00	89.91
12	71.21	69.78	72.84	73.15	71.21	66.77	67.00	73.15	74.27
13	94.00	86.00	94.00	92.00	92.00	88.00	92.00	92.00	92.00
14	73.93	76.72	75.38	77.31	73.93	71.80	76.20	78.31	79.31
15	89.67	85.54	96.89	94.78	89.67	94.98	95.89	94.78	94.78
16	65.48	63.81	80.88	78.63	65.48	67.77	80.16	77.91	79.33
17	41.64	50.07	59.71	55.12	41.64	55.33	60.62	55.95	56.97
18	58.61	51.00	57.67	60.61	58.61	52.72	63.67	60.61	61.61
19	58.57	68.67	65.10	66.10	58.57	52.57	60.86	67.29	66.76
20	85.52	76.86	86.95	95.14	85.52	72.69	88.05	93.48	97.14

B. Classification Results

The results obtained from the classification accuracy are presented in Table 3. The line graphs in Fig. 3 also represent classification results. The accuracy of the GAN-SVM method is shown in column 9 of Table 3. The results of other algorithms are presented from columns 2 to 8, respectively. Table 4 summarizes the results of these statistical tests with the paired Student ratio test.

The focus was on the classification performance comparison of the GAN-SVM with four other methods (*k*NN, DT, RF, and LSVM). In addition, four classifiers were also compared using enhanced GAN data: *k*NN (GAN-*k*NN), DT (GAN-DT), RF (GAN-RF), and LSVM (GAN-LSVM).

Table 4 shows that GAN-SVM outperforms *k*NN, DT, RF, and LSVM. Tables 3 and 4 also show that GAN-SVM significantly improves the accuracy means of 9.96, 8.62, 1.36, and 1.51 percentage points compared with *k*NN, DT, RF, and LSVM, respectively. All p-values in Table 4 are less than 0.05, except RF. In detail, GAN-SVM has 18 wins, 2 ties, and 0 defeats (p-value = 7.76E-05) against LSVM in column 4 of Table 3. In addition, GAN-SVM has 12 wins, 0 ties, and 8 defeats (p-value = 2.71E-01) compared with RF in column 3. In the comparison with *k*NN, the GAN-SVM has 19 wins, 0 ties, and 1 defeat (p-value = 2.31E-04). GAN-SVM gives good performance compared with DT, with 18 wins, 0 ties,

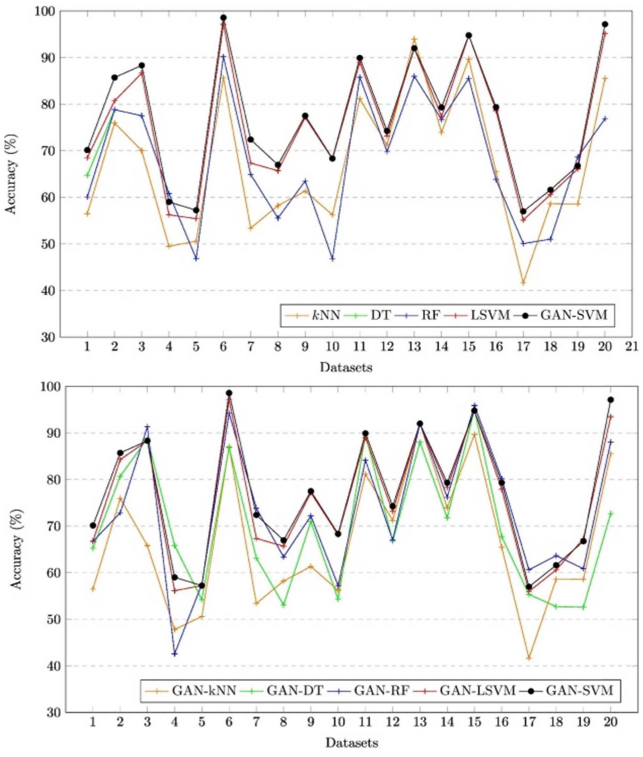

Fig. 3. Accuracy of these models on 20 datasets.

Table 4. Accuracy comparison between these models

Accuracy	Mean	Win	Tie	Defeat	p-value
*k*NN	66.85				
C4.5	68.19				
RF	75.45				
LSVM	75.25				
GAN-*k*NN	66.53				
GAN-C4.5	69.79				
GAN-RF	73.02				
GAN-LSVM	75.54				
GAN-SVM	**76.81**				
GAN-SVM & *k*NN		19	0	1	2.31E-04
GAN-SVM & C4.5		18	0	2	5.03E-06
GAN-SVM & RF		12	0	8	2.71E-01
GAN-SVM & LSVM		18	2	0	7.76E-05
GAN-SVM & GAN-*k*NN		15	4	1	7.94E-04
GAN-SVM & GAN-C4.5		19	1	0	1.22E-07
GAN-SVM & GAN-RF		17	0	3	2.31E-04
GAN-SVM & GAN-LSVM		12	1	7	6.62E-03

and 2 defeats, p-value = 5.03E-06.

The results in Tables 3 and 4 also show that GAN-SVM improves the accuracy mean of 10.28, 7.03, 3.79, and 1.27 percentage points obtained by GAN-*k*NN, GAN-DT, GAN-RF, and GAN-LSVM, respectively. These improvements are significant because the p-values are less than 0.05. In detail, GAN-SVM has 19 wins, 1 ties, and 0 defeats (p-value = 1.22E-07) compared with GAN-*k*NN. GAN-SVM has 17 wins, 0 ties, and 3 defeats (p-value = 2.31E-04) versus GAN-DT. In the comparison with GAN-RF, GAN-SVM outperforms 16 out of 20 datasets (12 wins, 1 tie, and 7 defeats, p-value = 6.62E-03).

GAN-SVM is slightly superior to GAN-LSVM, with 15 wins, 4 ties, and 1 defeat, p-value = 7.94E-04.

IV. CONCLUSION AND FUTURE WORKS

A new GAN-SVM method was proposed to classify gene expression data efficiently. The approach uses the GAN to generate new samples from original datasets, and then SVM is used as the classifying model. The test results of this investigation on 20 low-sample-size and very high-dimensional microarray gene expression datasets from the Kent Ridge Biomedical and ArrayExpress repositories show that the GAN-SVM model is more accurate than the state-of-the-art classifications, including *k*NN, SVMs, DTs of C4.5, and RFs. Further experimental investigations are recommended to estimate the best number of enhancement samples to provide a classification model for large datasets of microarray gene expression.

REFERENCES

[1] F. Bray, J. Ferlay, I. Soerjomataram, R. L. Siegel, L. A. Torre, and A. Jemal, "Global cancer statistics 2018: GLOBOCAN estimates of incidence and mortality worldwide for 36 cancers in 185 countries," *CA. Cancer J. Clin*, 2018. DOI: 10.3322/caac.21492.

[2] P. W. Novianti, V. L. Jong, K. C. B. Roes, and M. J. C. Eijkemans, "Factors affecting the accuracy of a class prediction model in gene expression data," *BMC Bioinformatics*, vol. 16, no. 1, 2015. DOI: 10.1186/s12859-015-0610-4.

[3] S. Y. Kim, "Effects of sample size on robustness and prediction accuracy of a prognostic gene signature," *BMC Bioinformatics*, vol. 10, no. 1, 2009. DOI: 10.1186/1471-2105-10-147.

[4] V. Vapnik, The nature of statistical learning theory, *Springer science & business media*, 1995.

[5] T. S. Furey, N. Cristianini, N. Duffy, D. W. Bednarski, M. Schummer, and D. Haussler, "Support vector machine classification and validation of cancer tissue samples using microarray expression data," *Bioinformatics*, vol. 16, no. 10, pp. 906-914, 2000. DOI: 10.1093/bioinformatics/16.10.906.

[6] I. Guyon, J. Weston, S. Barnhill, and V. Vapnik, "Gene selection for cancer classification using support vector machines," *Mach. Learn*, vol. 46, no. 1-3, pp. 389-422, 2002. DOI: 10.1023/A:1012487302797.

[7] J. Khan, J. S. Wei, M. Ringnér, L. H. Saal, M. Ladanyi, F. Westermann, F. Berthold, M. Schwab, C. R. Antonescu, C. Peterson and P. S. Meltzer, "Classification and diagnostic prediction of cancers using gene expression profiling and artificial neural networks," *Nat. Med*, vol. 7, no. 6, p. 673, 2001. DOI: 10.1038/89044.

[8] L. Li, C. R. Weinberg, T. A. Darden, and L. G. Pedersen. "Gene selection for sample classification based on gene expression data: study of sensitivity to choice of parameters of the GA/KNN method," *Bioinformatics*, vol. 17, no. 12, pp. 1131-1142, 2001. DOI: 10.1093/bioinformatics/17.12.1131.

[9] O. P. Netto, S. R. Nozawa, R. A. R. Mitrowski, A. A. Maced and J. A. Baranauskas, "Applying decision trees to gene expression data from DNA microarrays: A leukemia case study," *Anais*, 2010.

[10] L. Breiman, "Random forests," *Mach. Learn*, vol. 45, no. 1, pp. 5-32, 2001. DOI: 10.1023/A:1010933404324.

[11] R. Díaz-Uriarte and S. A. De Andres, "Gene selection and classification of microarray data using random forest," *BMC Bioinformatics*, vol. 7, no. 1, p. 3, 2006. DOI: 10.1186/1471-2105-7-3.

[12] T. N. Do, P. Lenca, S. Lallich, and N. K. Pham, "Classifying very-high-dimensional data with random forests of oblique decision trees," in *Advances in Knowledge Discovery and Management*, Springer, 2010, pp. 39-55. DOI: 10.1007/978-3-642-00580-0_3.

[13] L. Breiman, "Bagging predictors," *Mach. Learn*, vol. 24, no. 2, pp. 123-140, 1996. DOI: 10.1023/A:1018054314350.

[14] Y. Freund and R.E. Schapire, "A decision-theoretic generalization of on-line learning and an application to boosting," *J. Comput. Syst. Sci*, vol. 55, no. 1, pp. 119-139, 1995. DOI: 10.1006/jcss.1997.1504.

[15] M. Dettling, "BagBoosting for tumor classification with gene expression data," *Bioinformatics*, vol. 20, no. 18, pp. 3583-3593, 2004. DOI: 10.1093/bioinformatics/bth447.

[16] A. C. Tan and D. Gilbert, "Ensemble machine learning on gene expression data for cancer classification," *Appl. Bioinformatics*, vol. 2, no. 3 Suppl, pp. S75-83, 2003.

[17] P. H. Huynh, V. H. Nguyen, and T. N. Do, "A coupling support vector machines with the feature learning of deep convolutional neural networks for classifying microarray gene expression data," in *Modern Approaches for Intelligent Information and Database Systems*, Springer, 2018, pp. 233-243. DOI: 10.1007/978-3-319-76081-0.

[18] R. R. Bhat, V. Viswanath, and X. Li, "DeepCancer: Detecting cancer via deep generative learning through gene expressions," in *Proceedings of 2017 IEEE 15th Intl Conf on Dependable, Autonomic and Secure Computing, 15th Intl Conf on Pervasive Intelligence and Computing, 3rd Intl Conf on Big Data Intelligence and Computing and Cyber Science and Technology Congress*(*DASC/PiCom/DataCom/CyberSciTech*), 2017. DOI: 10.1109/DASC-PICom-DataCom-CyberSciTec. 2017.152.

[19] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. W. Farley, S. Ozair, A. Courville, Y. Bengio, "Generative adversarial nets," in *Advances in Neural Information Processing Systems*, pp. 2672-2680, 2014.

[20] A. Creswell, T. White, V. Dumoulin, K. Arulkumaran, B. Sengupta, and A. A. Bharath, "Generative adversarial networks: An overview," *IEEE Signal Process. Mag*, vol. 35, no. 1, pp. 53-65, 2018. DOI: 10.1109/MSP.2017.2765202.

[21] A. Ghahramani, F. M. Watt, and N. M. Luscombe, "Generative adversarial networks simulate gene expression and predict perturbations in single cells." Cold Spring Harbor Laboratory, 08-Feb-2018, [Online] Available: http://dx.doi.org/10.1101/262501.

[22] L. Jinyan and L. Huiqing, Kent Ridge Biomedical datasets repository. Technical report, 2002.

[23] A. Brazma et al., "ArrayExpress a public repository for microarray gene expression data at the EBI," *Nucleic Acids Res*, vol. 31, no. 1, pp. 68-71, 2003. DOI: 10.1093/nar/gkg091.

[24] A. Dosovitskiy, J. T. Springenberg, M. Tatarchenko, and T. Brox, "Learning to generate chairs, tables and cars with convolutional networks," *IEEE Trans. Pattern Anal. Mach. Intell*, vol. 39, no. 4, pp. 692-705, 2017. DOI: 10.1109/TPAMI.2016.2567384.

[25] C. Ledig, L. Theis, F. Huszar, J. Caballero, A. Cunningham, A. Acosta, A. Aitken, A. Tejani, J. Totz, Z. Wang, and W. Shi, "Photo-realistic single image super-resolution using a generative adversarial network," in *Proceedings of 2017 IEEE Conference on Computer Vision and Pattern Recognition* (*CVPR*), vol. 2, p. 4, 2017. DOI: 10.1109/CVPR.2017.19.

[26] E. Choi, S. Biswal, B. Malin, J. Duke, W. F. Stewart, and J. Sun, "Generating multi-label discrete patient records using generative adversarial networks," 2017, [Online] Available: https://arxiv.org/abs/1703.06490.

[27] O. Press, A. Bar, B. Bogin, J. Berant, and L. Wolf, "Language generation with recurrent generative adversarial networks without pre-training," 2017, [Online] Available: https://arxiv.org/abs/1706.01399, 2017.

[28] E. L. Denton, S. Chintala, R. Fergus, and others, "Deep generative image models using a laplacian pyramid of adversarial networks," in *Advances in Neural Information Processing Systems*, pp. 1486-1494, 2015.

[29] P. Costa, A. Galdran, M. I. Meyer, M. Abràmoff, A. M. Mendonça, and A. Campilho, "End-to-end adversarial retinal image synthesis," *IEEE Trans. Med. Imaging*, vol. 8, 2017. DOI: 10.1109/TMI.2017.2759102.

[30] P. Moeskops, M. Veta, M. W. Lafarge, K. A. Eppenhof, and J. P. Pluim, "Adversarial training and dilated convolutions for brain MRI segmentation," in *Deep Learning in Medical Image Analysis and Multimodal Learning for Clinical Decision Support,* Springer, pp. 56-64, 2017 DOI:10.1007/978-3-319-67558-9_7.

[31] C. J. Burges, "A tutorial on support vector machines for pattern recognition," *Data Min. Knowl. Discover*, vol. 2, no. 2, pp. 121-167, 1998. DOI: 10.1023/A:1009715923555

[32] V. Vapnik, "An overview of statistical learning theory," *IEEE Transactions on Neural Networks*, vol. 10, no. 5, pp. 988-999, 1998.

[33] U. H. G. Kreßel, "Pairwise Classification and Support Vector Machines," *Advances in Kernel Methods: Support Vector Learning*, 1999, pp. 255-268.

[34] N. Cristianini and J. Shawe-Taylor, "An introduction to support vector machines and other kernel-based learning methods." Cambridge University Press, 2000 [Online]. Available: http://dx.doi.org/10.1017/CBO9780511801389.

[35] M. Pirooznia, J. Y. Yang, M. Q. Yang, and Y. Deng, "A comparative study of different machine learning methods on microarray gene expression data," *BMC Genomics*, vol. 9, no. 1, p. S13, 2008. DOI: 10.1186/1471-2164-9-S1-S13.

[36] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," in *Proceedings of International Conference on Machine Learning*, pp. 448-456, 2015.

[37] C. W. Hsu, C. C. Chang, and C. J. Lin, "A practical guide to support vector classification," 2003, [Online] Available: https://www.csie.ntu.edu.tw/~cjlin/papers/guide/guide.pdf.

[38] M. Abadi, A. Agarwal, P. Barham, E. Brevdo, and C. Tensor, "Flow: Large-scale machine learning on heterogeneous systems," 2015, [Online] Available:.http://download.tensorflow.org/paper/whitepaper2015.pdf.

[39] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, and R. Weiss, "Scikit-learn: Machine learning in python," *J. Mach. Learn. Res*., vol. 12, pp. 2825-2830, 2011.

[40] C. C. Chang and C. J. Lin, "LIBSVM: a library for support vector machines," *ACM Trans. Intell. Syst. Technol. TIST*, vol. 2, no. 3, p. 1-27, 2011. DOI: 10.1145/1961189.1961199.

Phuoc-Hai Huynh

was born in Angiang in 1985. He received his bachelor's degree in Information Technology in 2007 from Angiang university, Vietnam. In 2014, he received his master's degree from Cantho university, Vietnam. Since 2008, he has been working at the Faculty of Information Technology, Angiang University. He is currently working as a Ph.D. candidate at the College of Information Technology, Cantho University. His research interests include data mining and bioinformatics.

Van Hoa Nguyen

was born in Dongthap in 1974. He received a Ph.D. degree in Computer Science from the University of Rennes 1 in 2009. He is currently deputy head of the Faculty of Information Technology and lecturer at the Information Technology, Angiang University, Vietnam. His research interests include bioinformatics, parallel computing, and data mining.

Thanh-Nghi Do

was born in Cantho in 1974. He received his Ph.D./M.S. degree in Computer Science from the University of Nantes in 2004 and 2002, respectively. He is currently head of the computer networks department, and senior lecturer at the College of Information Technology, Cantho University, Vietnam. He is also an associate researcher at UMI UMMISCO 209 (IRD/ UPMC), Sorbonne university, Pierre and Marie Curie University, France. His research interests include data mining with support vector machines, kernel-based methods, decision tree algorithms, ensemble-based learning, and information visualization. He has served on the program committees of international conferences and is a reviewer for the journals in his fields.