

# A Coupling Support Vector Machines with the Feature Learning of Deep Convolutional Neural Networks for Classifying Microarray Gene Expression Data



Phuoc-Hai Huynh, Van-Hoa Nguyen and Thanh-Nghi Do

**Abstract** Support vector machines (SVM) and deep convolutional neural networks (DCNNs) are state-of-the-art classification techniques in many real-world applications. Our investigation aims at proposing a hybrid model combining DCNNs and SVM (called DCNN-SVM) to effectively predict very-high-dimensional gene expression data. The DCNN-SVM trains the DCNNs model to automatically extract features from microarray gene expression data and followed which the DCNN-SVM learns a non-linear SVM model to classify gene expression data. Numerical test results on 15 microarray datasets from Array Expression and Medical Database (Kent Ridge) show that our proposed DCNN-SVM is more accurate than the classical DCNNs algorithm, SVM, random forests.

**Keywords** Microarray gene expression · Convolutional neural networks  
Support vector machines

## 1 Introduction

Nowadays, the development of high-throughput technologies such as DNA microarray has led to incremental growth in the public databases such as the ArrayExpress [1] and NCBI Gene Expression Omnibus [2]. Microarray is technology which enables researchers to investigate and address issues which is once thought to be non traceable by facilitating the simultaneous measurement of the expression levels of thousands of genes in a single experiment [3]. A characteristic of microarray gene

---

P.-H. Huynh (✉) · V.-H. Nguyen (✉)  
An Giang University, An Giang, Vietnam  
e-mail: hphai@agu.edu.vn

V.-H. Nguyen  
e-mail: nvhoa@agu.edu.vn

T.-N. Do (✉)  
Can Tho University, Can Tho, Vietnam  
e-mail: dtngchi@cit.ctu.edu.vn

expression data is that the number of variables (genes)  $m$  far exceeds the number of samples  $n$ , commonly known as curse of dimensionality problem. The vast amount of gene expression data leads to statistical and analytical challenges and conventional statistical methods give improper result due to high dimension of microarray data with a limited number of patterns [4]. It is not feasible when build machine learning model due to the extremely large features set with millions of features and high computing cost.

With the wealth of gene expression data from microarrays being produced, more and more new prediction, classification, and clustering techniques are being used for the analysis of the data. Many methods have been used for microarray gene expression data classification, and typical methods are support vector machines (SVM) [5–8], k-nearest neighbor classifier [9], C4.5 decision tree [10–12] and ensemble methods, such as random forests [13], random forests of oblique decision trees [14], bagging and boosting [15, 16].

In recent years, convolutional neural networks (CNNs) have achieved remarkable results in computer vision [17], text classification [18]. In addition, CNNs is also used for omics, biomedical imaging and biomedical signal processing [19]. Most data in bioinformatics are raw data such as gene sequences, proteins, microarray, medical image. Conventional machine learning algorithms have limitations in processing the raw form of data, so hybrid models often are used to combine the advantage of features extraction from the raw data of CNNs and performance classification of SVM or random forests (RF). The hybrid model neural network and SVM was initially proposed in [20]. In [21], model is later proposes in for handwritten digit recognition. More relevant previous work include [22], where a hybrid model approach is presented: the CNNs has trained using the back-propagation algorithm and the SVM is trained using a non-linear regression approach. It is noticeable that error classification rate gained by the hybrid model has achieved better results. In [23], the hybrid model uses for recognition for mobile swarm robotic systems. In addition, CNNs and RF are also combined to build hybrid model for electron microscopy images segmentation [24].

In this paper, we propose a hybrid model combining DCNNs and SVM (called DCNN-SVM) to effectively classify very-high-dimensional gene expression data. The main idea of our approach is to train a specialized DCNNs to extract robust hierarchical features from microarray gene expression data (MGE data) and provide them to SVM classifier using radial basis function kernel (RBF). Our approach differs from these previous ones as we build a single model instead of using disjoint classifiers trained separately. In relevant previous work, the CNNs is trained using the back-propagation algorithm and the SVM is trained using a non-linear regression approach, linear kernel function and random forest. The data in the relevant previous work was image such as: handwritten digit, medical image and video.

We have used 15 datasets of ArrayExpress [1] and Biomedical repository [25] to evaluate our model and also to compare to traditional classification methods such as DCNNs, support vector machines [26] and random forests [27]. The results showed that DCNN-SVM extract robust hierarchical features and improves classification

accuracy. Our method shows an excellent performance in general with support vector machines classifier using radial basis function kernel.

The paper is organized as follows. Section 2 presents our approach, a hybrid model combining DCNNs and SVM. Section 3 shows the experimental results. We then conclude in Sect. 4.

## 2 Methods

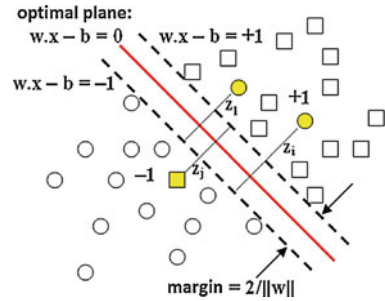
### 2.1 Deep Convolutional Neural Networks

DCNNs are designed to process multiple data types, especially two-dimensional images, and are directly inspired by the visual cortex of the brain. In the visual cortex, there is a hierarchy of two basic cell types: simple cells and complex cells [28]. Simple cells react to primitive patterns in sub-regions of visual stimuli, and complex cells synthesize the information from simple cells to identify more intricate forms. Since the visual cortex is such a powerful and natural visual processing system, DCNNs are applied to imitate three key ideas: local connectivity, invariance to location, and invariance to local transition [29]. There are three main types of layers used to build DCNNs architectures: convolutional layer, pooling layer, and fully connected layer. Normally, a full DCNNs architecture is obtained by stacking several of these layers. In a DCNNs, the key computation is the convolution of a feature detector with an input signal. Convolutional layer computes the output of neurons connected to local regions in the input, each one computing a dot product between their weights and the region they are connected to in the input volume. The set of weights which is convolved with the input is called filter or kernel. Every filter is small spatially (width and height), but extends through the full depth of the input volume. For inputs such as images typical filters are small areas and each neuron is connected only to this area in the previous layer. The weights are shared across neurons, leading the filters to learn frequent patterns that occur in any part of the image. The distance between the applications of filters is called stride. Whether stride hyper parameter is smaller than the filter size the convolution is applied in overlapping windows.

### 2.2 Support Vector Machines

Support vector machines (SVMs) proposed by Vapnik [26] are systematic and properly motivated by statistical learning theory. SVMs are the most well known as class of learning algorithms using the idea of kernel substitution. SVM and kernel-based methods have shown practical relevance for classification, regression [30]. The SVM algorithm is to find the best separating plane furthest from the different classes. In order to achieve this purpose, a SVM algorithm tries to simultaneously maximize

**Fig. 1** Linear separation of the datapoints into two classes



the margin (the distance between the supporting planes for each class) and minimize the error (any point falling on the wrong side of its supporting plane is considered to be an error). For binary classification problem (see Fig. 1), samples of one class are located on one side of the hyper-plane while samples of the other class are located on the other side of the hyper-plane.

For multiclass, one-versus-all [26], one-versus-one [31] are the most popular methods due to their simplicity. Let us consider  $k$  classes ( $k > 2$ ). The one-versus-all strategy builds  $k$  different classifiers where the  $i$ th classifier separates the  $i$ th class from the rest. The one-versus-one strategy constructs  $k(k - 1)/2$  classifiers, using all the binary pairwise combinations of the  $k$  classes. The class is then predicted with a majority vote.

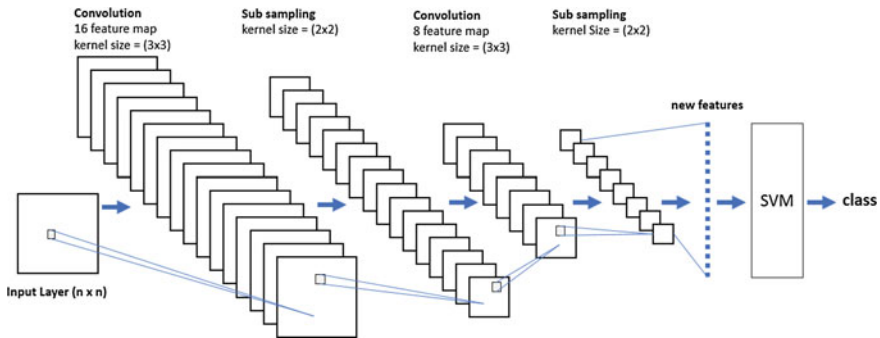
SVM can use some other classification functions, for example a polynomial function of degree  $d$ , a radial basis function (RBF) or a sigmoid function. More details about SVM and other kernel-based learning methods can be found in [32].

### 2.3 Support Vector Machines Using the Feature Extraction from Deep Convolutional Neural Networks

DCNNs are efficient at learning invariant features from data, but do not always produce optimal classification results. Conversely, a non-linear SVM cannot learn complex invariances, but produce good decision surfaces by maximizing margins using soft-margin approaches [33].

Our investigation is to propose a hybrid model architecture: A coupling SVM with the feature learning of DCNNs (denoted by DCNN-SVM) for classifying microarray gene expression data. The training task of DCNN-SVM consists of two main steps. First, the algorithm learns DCNNs to deeply extract functional features from high dimensional gene expression profiles. Next, it trains non-linear SVM models to perform the classification of the data representation extracted by the previous one.

The network architecture is shown in Fig. 2. Firstly, the first layer uses gene expression data. Secondly, the second and fourth layers of the network are convolution layers alternator with sub-sampling layers, which take the pooled maps as input.



**Fig. 2** The DCNN-SVM architecture

Consequently, they are able to extract features that are more and more invariant to local transformations of the input layer. The sixth layer is fully connected layer. The final layer is substituted by SVM with the RBF kernel for classification. The outputs from the hidden units are taken by the SVM as a feature vector for the training process. After that, the training stage continues till realizing good trained. Finally, classification on the test set is performed by the SVM classifier with such automatically extracted features.

### 3 Evaluation

We implement DCNN-SVM, SVM and random forests in python, using library SVM, LibSVM [34], tensorflow [35] and scikit library [36]. All tests were run under Linux Mint on a single 2.4 GHz Core I3 PC with 8 GB RAM.

#### 3.1 Experiments Setup

In our experiments, we use datasets provided by ArrayExpress database [1] and the Medical Database (Kent Ridge) [25]. ArrayExpress archive of Functional Genomics Data stores data from high-throughput functional genomics experiments. We downloaded MGE datasets from the ArrayExpress. The criteria for selecting the datasets were that the experiments had been conducted in humans and in the field of cancer. Datasets published or updated after 2012 and provided processed data. To reduce the source of variability of classification model performances because of the array used in the experiments, we retained studies conducted with Affymetrix array. The datasets and their characteristics are summarized in Table 1.

The test protocols are presented in the column 5 of Table 1. Some datasets are already divided in training set (Trn) and testing set (Tst). For these datasets, we used

**Table 1** Description of microarray gene expression datasets

ID	Name	Individuals	Attributes	Classes	Protocols	References
1	Lung cancer	181	12533	2	Trn-Tst	[37]
2	Prostate cancer	136	12600	2	Trn-Tst	[38] ID:68907
3	Astra Zeneca	627	54756	22	k-fold	[1] ID:57083
4	L. Leukaemia	575	22500	3	k-fold	[1] ID:33315
5	CCELE	917	19044	21	kfold	[1] ID:36133
6	GTex	837	22801	13	k-fold	[1] ID:45787
7	Breast cancer	327	54627	6	k-fold	[1] ID:20685
8	Breast cancer	97	24481	2	Trn-Tst	[39]
9	Leukemia	40	54675	2	loo	[1] ID:14858
10	Cancer cell. Project	950	54627	50	k-fold	[1] ID:MT-AB37
11	T-ALL & T-LL	29	15435	3	loo	[1] ID:1577
12	Sarcoma	105	22283	10	loo	[1] ID:6481
13	Lung cancer	203	12600	5	loo	[40]
14	Breast cancer	286	22283	3	loo	[1] ID:2034
15	Miscellaneous	50	10100	5	loo	[41]

the training data to build the our model. Then, we classified the testing set using the resulted model. With a datasets having less than 300 data points, the test protocol is leave-one-out cross-validation (loo). For the others, we used 10-fold cross-validation protocols remains the most widely to evaluate the performance [42]. Our evaluation used on the classification accuracy.

The DCNN-SVM architecture is shown in Table 2. It consist of 2 convolutional layers with 32 and 16 feature maps of  $(3 \times 3)$  kernel, and each convolutional layer has a  $(2 \times 2)$  average pooling layer followed. The features are taken from the last fully connected layer. SVM takes these outputs from the fully connected for classification. The one-versus-all method is utilized for the multi-class SVM that is possibly to be viewed as a trainable feature extractor. We have also tried other configurations of CNN, whereas this one gives the best performance. Input data are transformed the following way: we use microarray expression feature to represent each sample patient, which transform into a feature matrix. For deep convolutional neural networks configurations, we use ADAM method [43] for optimization, cross-entropy for loss function. The batch size is set to 16 and 50 epochs are used. We also tried to tune activation function with ReLU, Tanh and Sigmoid. The Tanh activation works better than other activation functions for microarray gene expression data.

We propose to use RBF kernel type in SVM models because it is general and efficient [44]. We also tried to tune parameters  $\gamma$  of RBF kernel and the cost  $C$  (a trade-off between the margin size and the errors) to obtain a good accuracy. These parameters are presented in Table 2.

**Table 2** Hyper-parameters of SVM used in DCNN-SVM

ID	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
$\gamma$	0.01	0.01	$3.85e-05$	0.01	0.001	0.01	0.01	0.01	0.01	$3.85e-05$	0.01	0.01	0.01	0.01	0.001
$C$	$10^5$	$10^5$	$10^5$	$10^5$	$10^6$	$10^5$	$10^5$	$10^5$	$10^5$	$10^5$	$10^6$	$10^5$	$10^5$	$10^5$	$10^6$

In order to evaluate the effectiveness of our approach, we used two different experiments to classify microarray samples. First, we compare DCNN-SVM with SVM, random forests (RF) and traditional DCNNs. In this experiments, RF algorithms build 200 decision trees and we use linear kernel type in SVM models ( $C = 10^5$ ,  $\gamma = 0.01$ ). Second, we compare different kernel functions in the SVM classifier: a linear kernel (DCNN-SVM linear) and a radial basis function (DCNN-SVM) with best parameter in Table 2. In addition, we also compared DCNN-SVM with DCNNs using random forest (DCNN-RF) classifier.

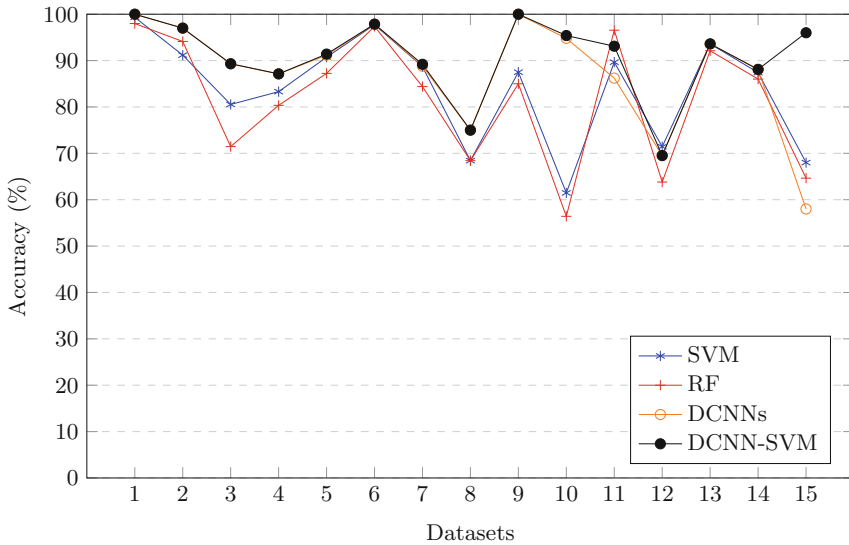
### 3.2 Experiments Results

Numerical test results on 15 microarray datasets are shown in Table 3. Results on 15 datasets showed that DCNN-SVM is more accuracy than the classical DCNNs algorithm, SVM, random forests. DCNN-SVM has the best accuracy of 11 out of 15 datasets. SVM and RF have the best only 1 out of 15 datasets. Table 3 and Fig. 3 showed that DCNN-SVM uses the RBF kernel to achieve the best accuracy result of 11 over 15 datasets. The DCNN-SVM uses linear kernel to achieve the best accuracy of 6 out of 15 datasets and DCNN-RF uses RF classifier has the best accuracy of 5

**Table 3** Classification results in terms of accuracy (%)

ID	SVM	RF	DCNNs	DCNN-RF	DCNN-SVM	
					Linear kernel	RBF kernel
1	99.33	97.99	100.0	100.0	100.0	100.0
2	91.18	94.12	97.00	97.00	97.00	97.00
3	80.54	71.45	89.30	88.51	88.96	89.32
4	83.30	80.35	87.14	86.62	87.13	87.13
5	90.73	87.24	91.18	91.40	91.17	91.40
6	97.73	97.37	97.86	97.74	97.86	97.86
7	88.69	84.40	88.80	88.30	88.80	89.20
8	68.42	68.42	75.00	75.00	75.00	75.00
9	87.50	85.00	77.50	85.00	100.0	100.0
10	61.41	56.42	94.8	95.00	95.4	95.4
11	89.66	96.55	86.21	96.55	93.10	93.10
12	71.43	63.81	65.71	62.86	69.52	69.52
13	93.60	92.12	93.60	94.09	93.60	93.60
14	87.41	86.01	87.06	87.76	88.11	88.11
15	68.00	64.65	56.00	62.00	58.00	96.00





**Fig. 3** Comparison of accuracy (%)

out of 15 datasets. DCNNs has the best accuracy of 4 out of 15 datasets. This superiority of DCNN-SVM (RBF) on CNNs, DCNN-SVM (RF) and DCNN-SVM (linear) showed in table results: 5 wins of DCNN-SVM (RBF) on DCNN-SVM (linear), 10 wins of DCNN-SVM (RBF) on DCNN-SVM (RF) and DCNNs on 15 datasets.

### 4 Conclusion and Future Works

We have presented a hybrid model combining DCNNs and SVM to classify very-high-dimension microarray gene expression data. The features are learned through a convolution process and then sent as input to a SVM classifier using RBF kernel to the objective of interest. After modifications through specified hyper parameters, the model performs quite comparatively well on the task tested on 15 different datasets from ArrayExpression and Medical Database. The numerical test results show that our proposal is more accurate than the classical DCNNs algorithm, support vector machines, random forests for classifying.

In the near future, we intend to provide more empirical test on large datasets of microarray gene expression and comparisons with other algorithms. Our proposal can be effectively parallelized. A parallel implementation that exploits the multicore processors can greatly speed up the learning and predicting tasks.

## References

1. Brazma, A., et al.: ArrayExpress a public repository for microarray gene expression data at the EBI. *Nucleic Acids Res.* **31**(1), 68–71 (2003)
2. Edgar, R., Domrachev, M., Lash, A.E.: Gene expression omnibus: NCBI gene expression and hybridization array data repository. *Nucleic Acids Res.* **30**(1), 207–210 (2002)
3. Schena, M., et al.: Quantitative monitoring of gene expression patterns with a complementary DNA microarray. *Science (New York then Washington)* 467–470 (1995)
4. Pinkel, D., et al.: High resolution analysis of DNA copy number variation using comparative genomic hybridization to microarrays. *Nat. Genet.* **20**(2) (1998)
5. Brown, M.P.S., et al.: Support vector machine classification of microarray gene expression data. University of California, Santa Cruz, Technical Report UCSC-CRL-99-09 (1999)
6. Furey, T.S., Cristianini, N., Duffy, N., Bednarski, D.W., Schummer, M., Haussler, D.: Support vector machine classification and validation of cancer tissue samples using microarray expression data. *Bioinformatics* **16**(10), 906–914 (2000)
7. Guyon, I., Weston, J., Barnhill, S., Vapnik, V.: Gene selection for cancer classification using support vector machines. *Mach. Learn.* **46**(1), 389–422 (2002)
8. Hasri, N.N.M., et al.: Improved support vector machine using multiple SVM-RFE for cancer classification. *Int. J. Adv. Sci. Eng. Inf. Technol.* **7**(4–2), 1589–1594 (2017)
9. Yeang, C.H., Ramaswamy, S., Tamayo, P., Mukherjee, S., Rifkin, R.M., Angelo, M., Reich, M., Lander, E., Mesirov, J., Golub, T.: Molecular classification of multiple tumor types. *Bioinformatics* **17**(suppl-1), S316–S322 (2001)
10. Li, J., Liu, H.: Ensembles of cascading trees. In: 2003 Third IEEE International Conference on Data Mining, ICDM 2003, pp. 585–588. IEEE (2003)
11. Li, J., Liu, H., Ng, S.K., Wong, L.: Discovery of significant rules for classifying cancer diagnosis data. *Bioinformatics* **19**(suppl-2), ii93–ii102 (2003)
12. Tsai, M.H., et al.: A decision tree based classifier to analyze human ovarian cancer cDNA microarray datasets. *J. Med. Syst.* **40**(1), 21 (2016)
13. Diaz-Uriarte, R., De Andres, S.A.: Gene selection and classification of microarray data using random forest. *BMC Bioinf.* **7**(1), 3 (2006)
14. Do, T.N., Lenca, P., Lallich, S., Pham, N.K.: Classifying very-high-dimensional data with random forests of oblique decision trees. In: *Advances in Knowledge Discovery and Management*, pp. 39–55. Springer (2010)
15. Tan, A.C., Gilbert, D.: Ensemble machine learning on gene expression data for cancer classification. *Bioinformatics* (2003)
16. Dettling, M.: Bagboosting for tumor classification with gene expression data. *Bioinformatics* **20**(18), 3583–3593 (2004)
17. Krizhevsky, A., et al.: ImageNet classification with deep convolutional neural networks. In: *Advances in Neural Information Processing Systems*, pp. 1097–1105 (2012)
18. Lai, S., Xu, L., Liu, K., Zhao, J.: Recurrent convolutional neural networks for text classification. *AAAI* **33**, 2267–2273 (2015)
19. Min, S., Lee, B., Yoon, S.: Deep learning in bioinformatics. *Brief. Bioinf.* (2016). <https://doi.org/10.1093/bib/bbw068>
20. Suykens, J.A., Vandewalle, J.: Training multilayer perceptron classifiers based on a modified support vector method. *IEEE Trans. Neural Netw.* **10**(4), 907–911 (1999)
21. Bellili, A., Gilloux, M., Gallinari, P.: An hybrid MLP-SVM handwritten digit recognizer. In: *Proceedings of the Sixth International Conference on Document Analysis and Recognition 2001*, pp. 28–32. IEEE (2001)
22. Niu, X.X., Suen, C.Y.: A novel hybrid CNN-SVM classifier for recognizing handwritten digits. *Pattern Recognit.* **45**(4), 1318–1325 (2012)
23. Nagi, J., et al.: Convolutional neural support vector machines: hybrid visual pattern classifiers for multi-robot systems. In: 2012 11th International Conference on Machine Learning and Applications (ICMLA), vol. 1, pp. 27–32. IEEE (2012)

24. Cao, G., Wang, S., Wei, B., Yin, Y., Yang, G.: A hybrid CNN-RF method for electron microscopy images segmentation. *Tissue Eng. J. Biomim. Biomater. Tissue Eng.* **18**, 2 (2013)
25. Jinyan, L., Huiqing, L.: Kent ridge bio-medical data set repository (2002)
26. Vapnik, V.: *Statistical Learning Theory*, vol. 1. Wiley, New York (1998)
27. Breiman, L.: Random forests. *Mach. Learn.* **45**(1), 5–32 (2001)
28. Hubel, D., Wiesel, T.: Shape and arrangement of columns in cat's striate cortex. *J. Physiol.* **165**(3), 559–568 (1963)
29. LeCun, Y., Bengio, Y., Hinton, G.: Deep learning. *Nature* **521**(7553), 436–444 (2015)
30. Burges, C.J.: A tutorial on support vector machines for pattern recognition. *Data Min. Knowl. Discov.* **2**(2), 121–167 (1998)
31. Krebel, U.H.G.: Pairwise classification and support vector machines. In: *Advances in Kernel Methods*, pp. 255–268. MIT press (1999)
32. Cristianini, N., Shawe Taylor, J.: *An introduction to support vector machines and other kernel-based learning methods*. Cambridge university press (2000)
33. Huang, F., LeCun, Y.: Large-scale learning with SVM and convolutional nets for generic object recognition. In: *2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition* (2006)
34. Chang, C.C., Lin, C.J.: LIBSVM: a library for support vector machines. *ACM Trans. Intell. Syst. Technol.* **2**, 27:1–27:27 (2011). Software available at <http://www.csie.ntu.edu.tw/~cjlin/libsvm>
35. Abadi, M., et al.: TensorFlow: large-scale machine learning on heterogeneous systems (2015). Software available from <http://www.tensorflow.org>
36. Pedregosa, F., et al.: Scikit-learn: machine learning in python. *J. Mach. Learn. Res.* **12**, 2825–2830 (2011)
37. Gordon, G.J., et al.: Translation of microarray data into clinically relevant cancer diagnostic tests using gene expression ratios in lung cancer and mesothelioma. *Cancer Res.* **62**(17), 4963–4967 (2002)
38. Singh, D., et al.: Gene expression correlates of clinical prostate cancer behavior. *Cancer Cell* **1**(2), 203–209 (2002)
39. Veer, V., et al.: Gene expression profiling predicts clinical outcome of breast cancer. *Nature* **415**(6871), 530–536 (2002)
40. Bhattacharjee, A., et al.: Classification of human lung carcinomas by mRNA expression profiling reveals distinct adenocarcinoma subclasses. *Proc. Natl. Acad. Sci.* **98**(24), 13790–13795 (2001)
41. Subramanian, A., et al.: Gene set enrichment analysis: a knowledge-based approach for interpreting genome-wide expression profiles. *Proc. Natl. Acad. Sci. U. S. A.* **102**(43), 15545–15550 (2005)
42. Wong, T.T.: Performance evaluation of classification algorithms by k-fold and leave-one-out cross validation. *Pattern Recognit.* **48**(9), 2839–2846 (2015)
43. Diederik, P., Kingma, J.B.: Adam: a method for stochastic optimization. In: *Proceedings of the 3rd International Conference on Learning Representations (ICLR)* (2014)
44. Hsu, C.W., et al.: *A practical guide to support vector classification* (2003)