Mathematical Biosciences and Engineering

MBE, 19(5): Page: 4881–4891.
DOI: 10.3934/mbe.2022228
Received: 21 December 2021
Revised: 03 February 2022
Accepted: 16 February 2022
Published 14 March 2022

http://www.aimspress.com/journal/MBE

Research article

GSEnet: feature extraction of gene expression data and its application to Leukemia classification

Kun Yu[1,2], Mingxu Huang[3], Shuaizheng Chen[3], Chaolu Feng[2,3,*] and Wei Li[2,3]

[1] College of Medicine and Biological Information Engineering, Northeastern University, Shenyang, Liaoning 110819, China
[2] Key Laboratory of Intelligent Computing in Medical Image, Ministry of Education, Shenyang, Liaoning 110819, China
[3] School of Computer Science and Engineering, Northeastern University, Shenyang, Liaoning 110819, China

* **Correspondence:** Email: fengchaolu@cse.neu.edu.cn;
Tel: +8613889147369; Fax: +8602483680729.

Abstract: Gene expression data is highly dimensional. As disease-related genes account for only a tiny fraction, a deep learning model, namely GSEnet, is proposed to extract instructive features from gene expression data. This model consists of three modules, namely the pre-conv module, the SE-Resnet module, and the SE-conv module. Effectiveness of the proposed model on the performance improvement of 9 representative classifiers is evaluated. Seven evaluation metrics are used for this assessment on the GSE99095 dataset. Robustness and advantages of the proposed model compared with representative feature selection methods are also discussed. Results show superiority of the proposed model on the improvement of the classification precision and accuracy.

Keywords: deep learning; gene expression; feature extraction; Resnet; SEnet

1. Introduction

With the rapid development of DNA microarray technology, it is possible to monitor gene activity from multiple aspects through the gene expression data. As gene expression reflects human health, it is potentially helpful for disease identification, prevention and treatment. However, it

remains a challenging task to find valuable information from gene expression data. One of the main reasons is that gene expression data consisting of thousands of dimensions, whereas only a small part is instructive [1]. Actually, feature selection and feature dimensionality reduction are two kinds of representative methods to select instructive features from gene expression data. For the former, a subset is selected from gene expression data, where Filter [2–9], Wrapper [10–15] and Embedded [16] are representative methods. Correspondingly, feature dimensionality reduction methods focus on mapping the features from high dimensional spaces to low-dimensional spaces, where feature values generally change during the mapping process. Representative feature dimensionality reduction methods are principal component analysis (PCA) [17,18], multiple dimensional scaling (MDS), locally linear embedding (LLE) [19,20], and so on.

Recently, deep learning methods have been widely used in many fields. For example, ResNet [21] and SENet [22] have achieved excellent performance in the field of image classification. However, there is still relatively little research in the field of selecting of instructive features from gene expression data. In this paper, we propose a deep learning model, namely GSEnet, to extract useful features from gene expression data. GSEnet is a hybrid of ResNet [21] and SENet [22]. Nine classifiers have been applied on the features extracted by the proposed model to evaluate its effectiveness.

The remainder of this paper is organized as follows. In Section 2, we briefly introduce the dataset used to evaluate the proposed model. In Section 3, we give details of the proposed model. In Section 4, we perform experimental evaluations of the proposed model. Finally, we discuss and conclude this paper in Sections 5 and 6.

2. DataSet

In this paper, we take a publicly available real single-cell RNA-seq as the evaluation dataset. The dataset comes from NCBI data repository (https://www.ncbi.nlm.nih.gov/geo/query/acc.cgi?acc=GSE99095), values of which are collected from bone marrow cells [23]. It consists of 979 samples among which 391 from healthy donors and 588 from patients with bone marrow failure and cytogenetic abnormalities. 17,258 expression genes have been monitored for each sample.

3. Methods

In this section, we describe details and framework of the proposed model (shown in Table 1 and Figure 1). The framework of the proposed model consists of three modules, namely the pre-conv module, the SE-Resnet module and the SE-conv module, respectively. Details of the modules are shown in the following subsections.

3.1. pre-conv module

The pre-conv module consists of a convolutional layer and a pooling layer, construction and function of which are the same as the first module in Resnet. Specifically, it employs a large convolutional kernel of size 7. Down-sampling has been directly performed at a stride of 2 to effectively reduce the feature dimension, as well as strengthen the local feature correlation. A maximum pooling layer is additionally used to reduce the feature dimension further. This operation

allows to effectively extract salient features encoded by the convolutional layer.

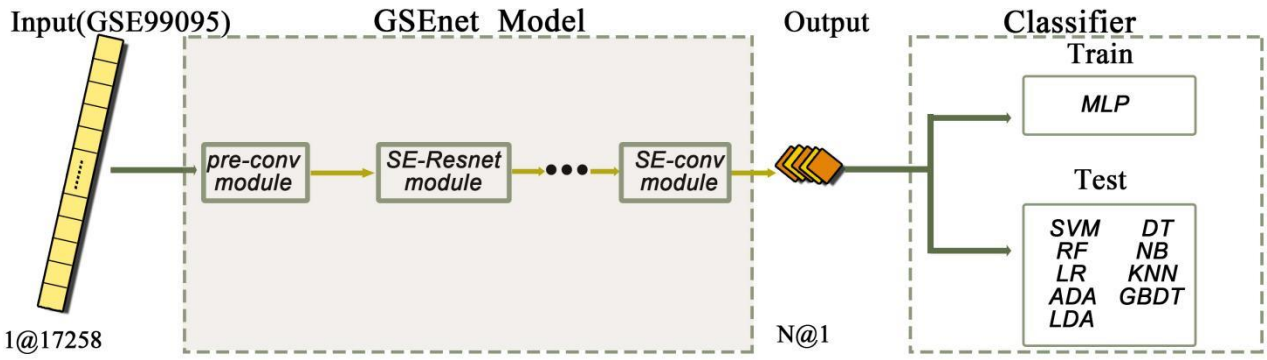

Figure 1. Framework of the proposed model.

Table 1. Setting details of GSEnet.

ID	Output Size	Block	Number	Block name
1	64 × 4314	conv [64, 7, 2, 3]; maxpool [3, 2]	1	pre-conv module
2	64 × 4314	conv [64, 1, 1, 0]; conv [64, 3, 1, 1]; conv [256, 1, 1, 0]; fc [16 , 256] (SE Block); conv [64 , 1, 1, 0]	2	SE-Resnet module
3	128 × 2157	conv [64, 1, 1, 0]; conv [64, 3, 1, 1]; conv [256, 1, 1, 0]; fc [16,256] (SE Block); conv [128, 1, 1, 0]; avgpool[2, 2]	1	SE-conv module
4	128 × 2157	conv [128, 1, 1, 0]; conv [128, 3, 1, 1]; conv [512, 1, 1, 0]; fc [32,512] (SE Block); conv [128, 1, 1, 0]	3	SE-Resnet module
5	256 × 1078	conv [128, 1, 1, 0]; conv [128, 3, 1, 1]; conv [512, 1, 1, 0]; fc [32,512] (SE Block); conv [256, 1, 1, 0]; avgpool[2, 2]	1	SE-conv module
6	256 × 1078	conv [256, 1, 1, 0]; conv [256, 3, 1, 1]; conv [1024, 1, 1, 0]; fc [64,1024] (SE Block); conv [256, 1, 1, 0]	5	SE-Resnet module
7	2048 × 1	conv [256, 1, 1, 0]; conv [256, 3, 1, 1]; conv [1024, 1, 1, 0]; fc [64,1024] (SE Block); conv [2048, 1, 1, 0]; global avgpool	1	SE-conv module

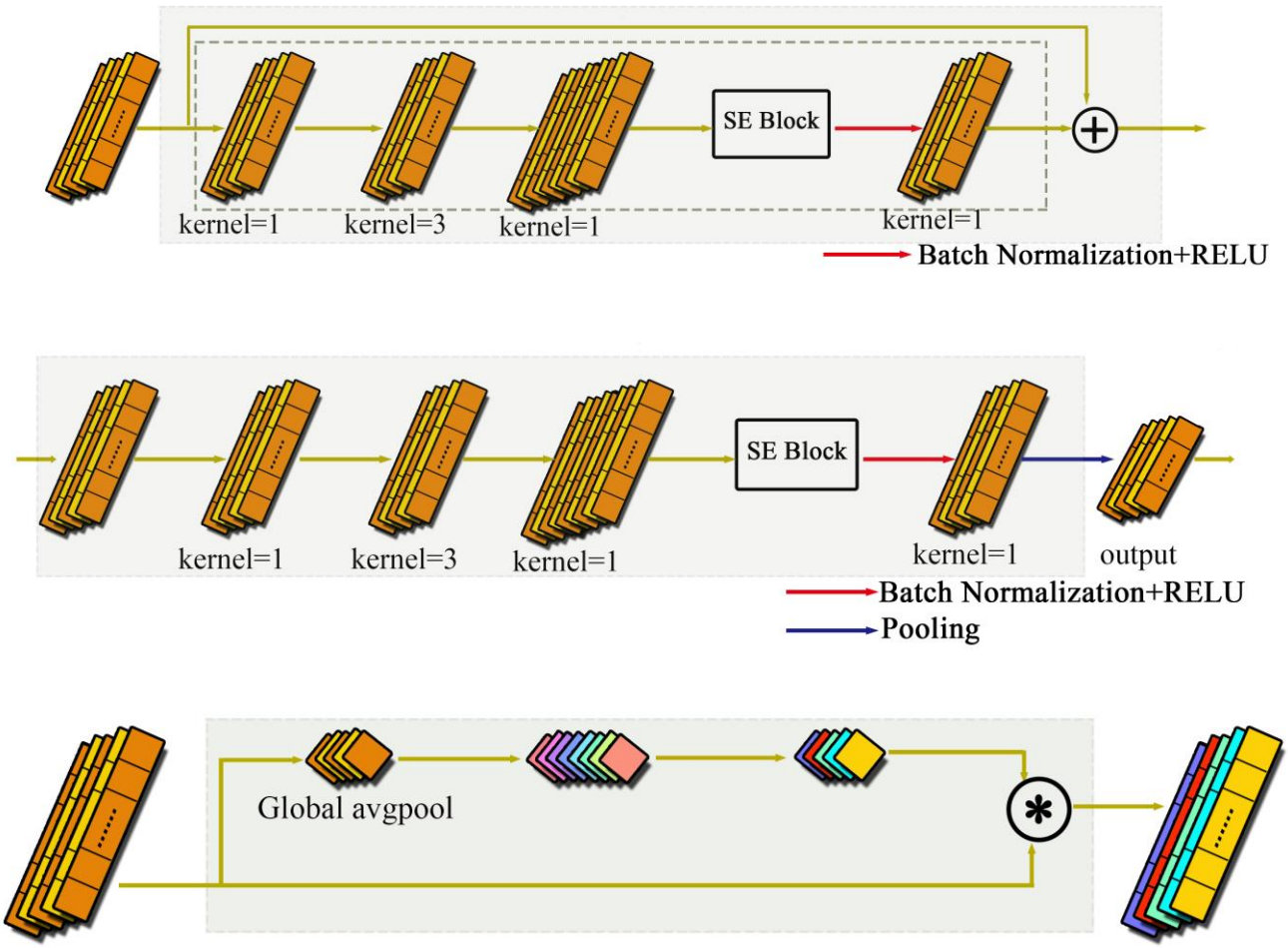

Figure 2: Details of the SE-Resnet module (upper), the SE-conv module (center), and the SE block (lower).

3.2. SE-Resnet module

The SE-Resnet module, as shown in Figure 2, consists of a ResNet block [21] and an SE block [22]. The ResNet block is helpful for the model to reuse features extracted by the pre-conv module, while the SE block is used to extract key features. The fusion and reasonable stacking of the two blocks helps the proposed model to extract higher-level semantic information. The output of the SE-Resnet module is defined as

$$\mathrm{F}\left(\mathbf{x}, w\right) = f\left(\delta\left(SE\left(f\left(f\left(f\left(\mathbf{x}, w_1\right), w_2\right), w_3\right), w_4\right)\right), w_5\right) + \mathbf{x} \tag{1}$$

where $\mathbf{x}$ is the input feature maps, f indicates the convolution operation, w_1, w_2, w_3 and w_5 are the parameters of each of the four convolution layers, δ refers to the ReLU function, and $SE(\bullet)$ corresponds to the SE block, parameters of which is denoted by w_4.

Let, $\hat{\mathbf{x}} = f\left(f\left(f\left(\mathbf{x}, w_1\right), w_2\right), w_3\right)$ the output of the SE block is defined as

$$\varepsilon\left(\hat{\mathbf{x}}, w_4\right) = SE\left(S\left(\hat{\mathbf{x}}\right), w_4\right)\hat{\mathbf{x}} \tag{2}$$

where

$$S\left(\hat{\mathbf{x}}\right) = \frac{1}{L}\sum_{i=1}^{L}\hat{\mathbf{x}}\left(i\right) \tag{3}$$

Note that L is the length of each feature map from $\hat{\mathbf{x}}$.

3.3. SE-conv module

The SE-conv module is similar to the SE-Resnet module except an additional pooling layer is appended at the end of the module as shown in Figure 2. Its function is mainly to increase the feature levels and reduce the feature dimension. Let w_6, w_7, w_8 and w_{10} be the parameters of each of the four convolution layers and $\mathbf{x}$ the input feature maps, the output of the SE-conv module is defined as

$$\mathrm{H}\left(\mathbf{x}, w\right) = P\left(f\left(\delta\left(SE\left(f\left(f\left(f\left(\mathbf{x}, w_6\right), w_7\right), w_8\right), w_9\right)\right), w_{10}\right)\right) \tag{4}$$

where w_9 denotes parameters of the SE block and $P\left(\bullet\right)$ is the global average pooling if the SE block is the last one or an average pooling with factor 2 for other SE blocks.

4. Experiments

4.1. Experimental details

Training details. During the training process, we connect a multilayer perceptron model at the end of the proposed instructive feature extraction model as shown in Figure 1. The number of nodes in the hidden layer is 256 and 64 respectively. Adam optimizer is used. The learning rate is set to 10^{-6}, and the loss function is cross entropy. The dataset is divided into training and validation sets by a 9-to-1 ratio. Early stop is adopted to avoid over fitting.

Test details. Ten-fold cross-validation is used to evaluate performance of the selected classifiers, details of which will be described in the next subsection.

Experimental Environment. The proposed model is implemented in PyCharm on a computer with Inter(R) Core(TM) i7-8700U CPU @ 3.20GHz, NVIDIA GeForce GTX 1050 Ti, and Windows 10 operating system. It costs 12 hours to train the proposed model and the prediction is less than 1 minute.

4.2. Experimental results

To verify effectiveness of features extracted by the proposed model, they are applied into the following classifiers, Support Vector Machine (SVM), Decision Tree (DT), Random forest (RF),

Naive Bayes (NB), Logistic Regression (LR), k-Nearest Neighbor (KNN), AdaBoost (ADA), Gradient Boosted Decision Tree (GBDT) and Linear Discriminant Analysis (LDA). Implementations of all these classifiers can be found in the scikit-learn library for python (https://scikit-learn.org/stable/). The evaluation metrics we adopted are true positive rate (TPR), false negative rate (FNR), false positive rate (FPR), ture negative rate (TNR), Precision(PRE), F1-score (F1), and accuracy (ACC), respectively. Specifically, the above mentioned evaluation metrics are defined as $\mathrm{TPR}=\mathrm{TP}/(\mathrm{TP}+\mathrm{FN})$, $\mathrm{FNR}=\mathrm{FN}/(\mathrm{TP}+\mathrm{FN})$, $\mathrm{FPR}=\mathrm{FP}/(\mathrm{FP}+\mathrm{TN})$, $\mathrm{TNR}=\mathrm{TN}/(\mathrm{TN}+\mathrm{FP})$, $\mathrm{PRE}=\mathrm{TP}/(\mathrm{TP}+\mathrm{FP})$, $\mathrm{F1}=2\times\mathrm{P}\times\mathrm{R}/(\mathrm{P}+\mathrm{R})$, and $\mathrm{ACC}=(\mathrm{TP}+\mathrm{TN})/(\mathrm{TP}+\mathrm{FP}+\mathrm{TN}+\mathrm{FN})$. TP is true positive representing the total number of samples correctly identified as positive, while FP is false positive representing the number of samples incorrectly identified as positive. Similarly, TN is true negative, namely the number of samples correctly identified as negative, whereas FN is a false negative, i.e. the number of samples incorrectly identified as negative. P and R are precision and recall. Experimental results are given in Figure 3 and Table 2. Note that values in Figure 3 are metric means of ten-fold cross-evaluation. Standard deviations are given in Table 2. It is obvious that performances of the classifiers are improved. In particular, KNN, ADA, NB, RF, DT and LDT, which do not perform well on original samples, achieve similar performances as SVM and GDBT. Thus, the proposed model is effective on performance improvement of the classifiers.

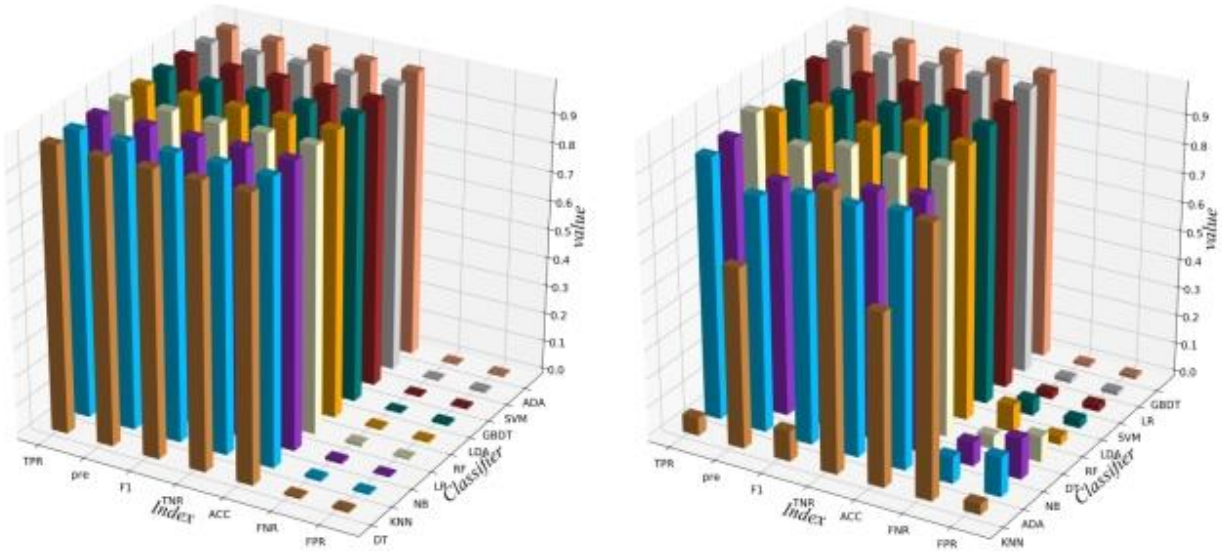

Figure 3. Effectiveness of the proposed model on performance improvement of different classifiers. On the left is without GSEnet, while the right is with GSEnet.

Table 2. Effectiveness of the proposed model on performances of different classifiers.

Classifier	method	TPR	TNR	ACC	PRE	F1	FNR	FPR
KNN	None	0.0622 ±0.0405	0.9616 ±0.0500	0.6027 ±0.0340	0.6267 ±0.3543	0.1065 ±0.0605	0.9378 ±0.0405	0.0384 ±0.0500
	GSEnet	0.9895 ±0.0132	0.9933 ±0.0083	0.9919 ±0.0058	0.9899 ±0.0125	0.9896 ±0.0071	0.0105 ±0.0132	0.0067 ±0.0083
ADA	None	0.9070 ±0.0450	0.8606 ±0.0414	0.8785 ±0.0264	0.8130 ±0.0463	0.8559 ±0.0281	0.0930 ±0.0450	0.1394 ±0.0414
	GSEnet	0.9954 ±0.0093	0.9933 ±0.0083	0.9940 ±0.0065	0.9899 ±0.0125	0.9926 ±0.0076	0.0046 ±0.0093	0.0067 ±0.0083
NB	None	0.9168 ±0.0429	0.8573 ±0.0405	0.8805 ±0.0245	0.8110 ±0.0447	0.8592 ±0.0266	0.0832 ±0.0429	0.1427 ±0.0405
	GSEnet	0.9895 ±0.0132	0.9951 ±0.0075	0.9929 ±0.0047	0.9919 ±0.0124	0.9906 ±0.0064	0.0105 ±0.0132	0.0049 ±0.0075
RF	None	0.9053 ±0.0370	0.9717 ±0.0205	0.9447 ±0.0189	0.9535 ±0.0376	0.9281 ±0.0275	0.0947 ±0.0370	0.0283 ±0.0205
	GSEnet	0.9922 ±0.0123	0.9933 ±0.0083	0.9930 ±0.0063	0.9899 ±0.0125	0.9910 ±0.0076	0.0078 ±0.0123	0.0067 ±0.0083
DT	None	0.9540 ±0.0284	0.9104 ±0.0387	0.9279 ±0.0226	0.8777 ±0.0476	0.9132 ±0.0263	0.0460 ±0.0284	0.0896 ±0.0387
	GSEnet	0.9922 ±0.0123	0.9914 ±0.0086	0.9919 ±0.0058	0.9876 ±0.0126	0.9898 ±0.0070	0.0078 ±0.0123	0.0086 ±0.0086
LDA	None	0.9452 ±0.0350	0.9707 ±0.0278	0.9593 ±0.0217	0.9532 ±0.0475	0.9481 ±0.0278	0.0548 ±0.0350	0.0293 ±0.0278
	GSEnet	0.9954 ±0.0093	0.9914 ±0.0086	0.9930 ±0.0063	0.9876 ±0.0126	0.9914 ±0.0073	0.0046 ±0.0093	0.0086 ±0.0086
SVM	None	0.9776 ±0.0227	0.9774 ±0.0217	0.9767 ±0.0153	0.9641 ±0.0377	0.9702 ±0.0203	0.0224 ±0.0227	0.0226 ±0.0217
	GSEnet	0.9954 ±0.0093	0.9933 ±0.0083	0.9940 ±0.0065	0.9899 ±0.0125	0.9926 ±0.0076	0.0046 ±0.0093	0.0067 ±0.0083
LR	None	0.9851 ±0.0162	0.9884 ±0.0129	0.9869 ±0.0088	0.9823 ±0.0192	0.9835 ±0.0109	0.0149 ±0.0162	0.0116 ±0.0129
	GSEnet	0.9895 ±0.0132	0.9951 ±0.0075	0.9929 ±0.0047	0.9919 ±0.0124	0.9906 ±0.0064	0.0105 ±0.0132	0.0049 ±0.0075
GBDT	None	0.9903 ±0.0157	0.9883 ±0.0163	0.9890 ±0.0118	0.9834 ±0.0235	0.9867 ±0.0145	0.0097 ±0.0157	0.0117 ±0.0163
	GSEnet	0.9948 ±0.0108	0.9936 ±0.0079	0.9939 ±0.0068	0.9887 ±0.0140	0.9917 ±0.0100	0.0052 ±0.0108	0.0064 ±0.0079

5. Discussion

5.1. Effect of the SE-Resnet modules

In this subsection, we discuss effects of SE-Resnet modules on classifier performance in terms of F1 score and results are given in Table 3. From Table 1, we can see that the number of SE-Resnet modules is ten. We delete existing modules or add new ones to the model with S and A as indicators. S2 and S4 indicate remove 2 and 4 modules from the SE-Resnet module shown in Table 1 with ID = 6, respectively, while S6 further remove 2 modules from the SE-Resnet module with tag ID = 4. Correspondingly, A2 and A4 indicate appending one or two additional SE-Resnet module for the modules given in ID = 4 and ID = 6. It is obvious that the original network setting given in Table 1, namely GSEnet, performs the best. However, effect of structure changes on the performance is little.

5.2. Comparison with feature selection methods

In this subsection, we compare the proposed model with representative feature selection methods, i.e., t-test (T), analysis of variance (Var), lasso feature selection (Lasso) and Logistic Regression feature selection (Log) in terms of F1 and show the result in Table 4. The features with a p-value less than 0.05 are included in the results of t-test, and 8174 feature values remain. For others, we select the top K salient feature values, where K = 256, K = 512, K = 1024, K = 2048 and K = 4096. For the proposed model, we modify the number of output channels of the last convolutional layer. It is obvious that the proposed model introduces the best performance on DT, RF, NB, KNN, ADA and GDBT classifiers.

Table 3. Effect of the SE-Resnet modules on performances of different classifiers in terms of F1.

	SVM	DT	RF	NB	LR	KNN	ADA	GDBT	LDA
GSENet S6	0.9820	0.9791	0.9791	0.9859	0.9833	0.9818	0.9807	0.9777	0.9831
GSEnet S4	0.9777	0.9764	0.9754	0.9750	0.9777	0.9735	0.9739	0.9716	0.9735
GSEnet S2	0.9811	0.9767	0.9741	0.9811	0.9798	0.9811	0.9775	0.9706	0.9714
GSEnet	0.9926	0.9898	0.9910	0.9906	0.9906	0.9896	0.9926	0.9917	0.9914
GSEnet A2	0.9774	0.9833	0.9857	0.9800	0.9865	0.9826	0.9802	0.9827	0.9842
GSEnet A4	0.9797	0.9807	0.9824	0.9823	0.9834	0.9730	0.9795	0.9869	0.9814

Table 4. Comparison of the proposed model with feature selection methods on performances of different classifiers in terms of F1.

	SVM	DT	RF	NB	LR	KNN	ADA	GDBT	LDA
T	0.9833	0.9132	0.9520	0.8828	0.9886	0.2154	0.8846	0.9840	0.9649
Var_256	0.9249	0.7563	0.9351	0.7126	0.8339	0.3924	0.7418	0.9411	0.8636
Var_512	0.9249	0.7563	0.9351	0.7126	0.8339	0.3924	0.7418	0.9389	0.8636
Var_1024	0.9249	0.7563	0.9351	0.7126	0.8339	0.3924	0.7418	0.9387	0.8636
Var_2048	0.9249	0.7563	0.9351	0.7126	0.8339	0.3924	0.7418	0.9373	0.8636
Var_4096	0.9249	0.7563	0.9351	0.7126	0.8339	0.3924	0.7418	0.9401	0.8636
Lasso_256	**0.9951**	0.9132	0.9894	0.9535	0.9963	0.8829	0.9681	0.9857	**0.9975**
Lasso_512	0.9939	0.9132	0.9857	0.9216	0.9951	0.6134	0.9421	0.9857	**0.9975**
Lasso_1024	0.9911	0.9132	0.9752	0.9118	0.9939	0.1550	0.9161	0.9819	0.9434
Lasso_2048	0.9899	0.9132	0.9735	0.8870	0.9939	0.0972	0.8860	0.9802	0.9393
Lasso_4096	0.9851	0.9132	0.9634	0.8603	0.9951	0.1048	0.8628	0.9876	0.9439
Log_256	**0.9951**	0.9132	0.9870	0.9750	0.9912	0.9531	0.9859	0.9863	0.9934
Log_512	0.9947	0.9132	0.9871	0.9738	0.9959	0.8473	0.9809	0.9876	0.9911
Log_1024	0.9935	0.9132	0.9830	0.9747	**0.9984**	0.6214	0.9747	0.9868	0.9214
Log_2048	0.9923	0.9132	0.9728	0.9771	**0.9984**	0.4158	0.9747	0.9827	0.9871
Log_4096	0.9923	0.9132	0.9744	0.9746	0.9959	0.2276	0.9746	0.9841	0.9778
GSEnet_256	0.9880	0.9891	0.9863	0.9824	0.9880	0.9891	0.9827	0.9878	0.9905
GSEnet_512	0.9878	0.9828	0.9854	0.9890	0.9907	0.9878	0.9852	0.9856	0.9865
GSEnet_1024	0.9774	0.9735	0.9814	0.9759	0.9761	0.9776	0.9791	0.9825	0.9756
GSEnet	0.9926	**0.9898**	**0.9910**	**0.9906**	0.9906	**0.9896**	**0.9926**	**0.9917**	0.9914
GSEnet_4096	0.9807	0.9781	0.9797	0.9807	0.9820	0.9793	0.9735	0.9799	0.9770

6. Conclusions

In this paper, a novel deep learning model, namely GSEnet, is proposed. It combines ResNet and SENet, and is constructed to improve the extraction of instructive features from gene expression data. The proposed model has been evaluated on the GSE99095 dataset with 9 representative classifiers. Experimental results show advantages of the proposed model on performance improvement of different classifiers compared with t-test, analysis of variance, lasso and Logistic Regression feature selection methods, GSEnet introduces the best performance on DT, RF, NB, KNN, ADA and GDBT classifiers.

Acknowledgments

This work was supported by the Natural Science Foundation of Liaoning Province under grant 2021-MS-085. The authors would like to thank the handling editor and anonymous reviewers very much for their constructive suggestions on improving quality of this paper.

Conflict of Interest

No potential conflict of interest was reported by the authors.

References

1. A. K. Shukla, P. Singh, M. Vardhan, A two-stage gene selection method for biomarker discovery from microarray data for cancer classification, *Chemometr. Intell. Lab. Syst.*, **183** (2018), 47–58. https://doi.org/10.1016/j.chemolab.2018.10.009
2. S. Hautaniemi, O. Yli-Harja, J. Astola, P. Kauraniemi, A. Kallioniemi, M. Wolf, et al., Analysis and visualization of gene expression microarray data in human cancer using self-organizing maps, *Mach. Learn.*, **52** (2003), 45–66. https://doi.org/10.1023/A:1023941307670
3. J. H. Hong, S. B. Cho, The classification of cancer based on dna microarray data that uses diverse ensemble genetic programming, *Artif. Intell. Med.*, **36** (2006), 43–58. https://doi.org/10.1016/j.artmed.2005.06.002
4. M. Hollstein, D. Sidransky, B. Vogelstein, C. C. Harris, p53 mutations in human cancers, *Science*, **253** (1991), 49–53. https://doi: 10.1126/science.1905840
5. T. Latkowski, S. Osowski, Data mining for feature selection in gene expression autism data, *Expert Syst. Appl.*, **42** (2015), 864–872. https://doi.org/10.1016/j.eswa.2014.08.043
6. Y. Wang, F. S. Makedon, J. C. Ford, J. Pearlman, Hykgene: a hybrid approach for selecting marker genes for phenotype classification using microarray gene expression data, *Bioinformatics*, **21** (2005), 1530–1537. https://doi.org/10.1093/bioinformatics/bti192
7. W. Hu, W. Hu, S. Maybank, Adaboost-based algorithm for network intrusion detection, *IEEE Trans. Syst. Man Cybern. B Cybern.*, **38** (2008), 577–583. https://doi.org/10.1109/TSMCB.2007.914695
8. C. L. Huang, C. J. Wang, A ga-based feature selection and parameters optimizationfor support vector machines, *Expert Syst. Appl.*, **31** (2006), 231–240. https://doi.org/10.1016/j.eswa.2005.09.024

9. A. K. Jain, R. P. W. Duin, J. Mao, Statistical pattern recognition: A review, *IEEE Trans. Pattern Anal. Mach. Intell.*, **22** (2000), 4–37. https://doi.org/10.1109/34.824819
10. L. Li, T. A. Darden, C. Weingberg, A. Levine, L. G. Pedersen, Gene assessment and sample classification for gene expression data using a genetic algorithm/k-nearest neighbor method, *Comb. Chem. High Throughput Screening*, **4** (2001), 727–739. https://doi.org/10.2174/1386207013330733
11. X. Huang, L. Zhang, B. Wang, F. Li, Z. Zhang, Feature clustering based support vector machine recursive feature elimination for gene selection, *Appl. Intell.*, **48** (2018), 594–607. https://doi.org/10.1007/s10489-017-0992-2
12. R. Díaz-Uriarte, S. A. De Andres, Gene selection and classification of microarray data using random forest, *BMC bioinformatics*, **7** (2006), 1–13. https://doi: 10.1186/1471-2105-7-3
13. I. Guyon, J. Weston, S. Barnhill, V. Vapnik, Gene selection for cancer classification using support vector machines, *Mach. Learn.*, **46** (2002), 389–422. https://doi.org/10.1023/A:1012487302797
14. L. Vinh, S. Lee, Y. T. Park, B. J. dAuriol, A novel feature selection method based on normalized mutual information, *Appl. Intell.*, **37** (2012), 100–120. https://doi.org/10.1007/s10489-011-0315-y
15. R. Ruiz, J. C. Riquelme, J. S. Aguilar-Ruiz, Incremental wrapper-based gene selection from microarray data for cancer classification, *Pattern Recognit.*, **39** (2006), 2383–2392. https://doi.org/10.1016/j.patcog.2005.11.001
16. S. Szedmak, J. Shawe-Taylor, C. J. Saunders, D. R. Hardoon, Multiclass classification by l1 norm support vector machine, in *Pattern recognition and machine learning in computer vision workshop*, **5** (2004).
17. E. Lotfi, A. Keshavarz, Gene expression microarray classification using PCA–BEL, *Comput. Biol. Med.*, **54** (2014), 180–187. https://doi.org/10.1016/j.compbiomed.2014.09.008
18. K. Y. Yeung, W. L. Ruzzo, Principal component analysis for clustering gene expression data, *Bioinformatics*, **17** (2001), 763–774. https://doi.org/10.1093/bioinformatics/17.9.763
19. L. Sun, W. Wang, J. Xu, S. Zhang, Improved lle and neighborhood rough sets-based gene selection using lebesgue measure for cancer classification on gene expression data, *J. Intell. Fuzzy Syst.*, **37** (2019), 5731–5742. https://doi.org/10.3233/JIFS-181904
20. L. Sun, J. Xu, W. Wang, Y. Yin, Locally linear embedding and neighborhood rough set-based gene selection for gene expression data classification, *Genet. Mol. Res.*, **15** (2016), 15038990. http://dx.doi.org/10.4238/gmr.15038990
21. K. He, X. Zhang, S. Ren, J. Sun, Deep residual learning for image recognition, in *Proceedings of the IEEE conference on computer vision and pattern recognition*, (2016), 770–778. http://doi.org/10.1109/CVPR.2016.90
22. J. Hu, L. Shen, G. Sun, Squeeze-and-excitation networks, in *Proceedings of the IEEE conference on computer vision and pattern recognition*, (2018), 7132–7141. http://doi.org/10.1109/CVPR.2018.00745
23. X. Zhao, S. Gao, Z. Wu, S. Kajigaya, X. Feng, Q. Liu, et al., Single-cell rna-seq reveals a distinct transcriptome signature of aneuploid hematopoietic cells, *Blood*, **130** (2017), 2762–2773. http://doi.org/10.1182/blood-2017-08-803353

AIMS Press

©2022 Author (s), licensee AIMS Press. This is an open access article distributed under the terms of the Creative Commons Attribution License (http://creativecommons.org/licenses/by/4.0)