Novel hybrid DCNN–SVM model for classifying RNA-sequencing gene expression data*

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

aInformation Technology Faculty, An Giang University, An Giang, Vietnam; bCollege of Information Technology, Can Tho University, Can Tho, Vietnam; cUMI UMMISCO 209 (IRD/UPMC), UPMC, Sorbonne University, Pierre and Marie Curie University, Paris, France

ABSTRACT
In recent years, cancer is one of the leading causes of death worldwide. Therefore, there are more and more studies that have been conducted to find effective solutions to diagnose and treat cancer. However, there are still many challenges in cancer treatment because possible causes of cancer are genetic disorders or epigenetic alterations in the cells. RNA sequencing is a powerful technique for gene expression profiling in model organisms and it is able to produce information for diagnosing cancer at the biomolecular level. Gene expression data are used to build a classification model which supports treatment of cancer. Nevertheless, its characteristic is very-high-dimensional data which lead to over-fitting issue of classifying model. In this paper, we propose a new gene expression classification model of support vector machines (SVM) using features extracted by deep convolutional neural network (DCNN). In our approach, the DCNN extracts latent features from gene expression data, then they are used in conjunction with SVM that efficiently classify RNA-Seq gene expression data. Numerical test results on RNA-Seq gene expression datasets from The Cancer Genome Atlas (TCGA) illustrate that our proposed algorithm is more accurate than state-of-the-art classifying models including DCNN, SVM and random forests.

1. Introduction
In recent years, precision medicine has become a potential method using genetic information systems to optimize treatment and revolutionize care. Next generation sequencing technologies are used to resolve many important issues for personalized medicine (Snyder, 2016). Gene expression data generated from these technologies could measure

CONTACT Phuoc-Hai Huynh hphai@agu.edu.vn
*This paper is an extended version of 11th Asian Conference on Intelligent Information and Database Systems (ACIIDS 2018) Paper (Huynh, Nguyen, & Do, 2018a). In this paper, in addition to ACIIDS 2018 paper, we have been built DCNN–SVM model RNA-Seq gene expression data classification. This model extracts features from RNA-Seq gene expression data. After tuning hyper parameters, the proposed model performs quite comparatively well on the task tested on RNA-Seq gene expression datasets from The Cancer Genome Atlas (TCGA). 25 small and medium RNA-Seq datasets and a large RNA-Seq dataset of 36 cancer type are also added to the experiment.

© 2019 The Author(s). Published by Informa UK Limited, trading as Taylor & Francis Group
This is an Open Access article distributed under the terms of the Creative Commons Attribution License (http://creativecommons.org/licenses/by/4.0/), which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.
the level of activity of genes in the cells as well as provide a lot of useful information about the complex activities within the corresponding cells. In fact, these technologies are used to compare gene transcription in cancer cells versus normal tissues in thousands of genes. Gene expression data are used for preventing, diagnosing, treating cancer because this disease is associated with multiple genetic and regulatory aberrations of tumour progression and is reflected by gene expression data (Aluru, 2005). Therefore, researchers can obtain better insight into the cancer pathology (Tan & Gilbert, 2003). Many gene expression classification studies have revealed distinct tumour subtypes and uncovered expression patterns that were associated with clinical outcomes (Bhattacharjee et al., 2001; Reis-Filho & Pusztai, 2011).

The dominant contemporary techniques, RNA sequencing (RNA-Seq) and DNA microarray are major technologies for performing high-throughput analysis of gene expression. DNA microarray technology measures the abundances of a defined set of transcripts via their hybridization to an array of complementary probes. In recent decades, there have been many classification algorithms using DNA microarray gene expression data which provide a lot of contributions in cancer research such as (Golub et al., 1999; Guyon, Weston, Barnhill, & Vapnik, 2002). However, this technology has several limits because the background hybridization limits accuracy of expression measurements, particularly for transcripts present in low abundance. Furthermore, probes differ considerably in their hybridization properties, and arrays are limited to interrogating only those genes for which probes are designed (Zhao, Fung-Leung, Bittner, Ngo, & Liu, 2014). On the other hand, RNA-Seq technology provides insight into the transcriptome in a cell. It provides far higher coverage and greater resolution than DNA microarray of the dynamic nature of the transcriptome (Kukurba & Montgomery, 2015). Therefore, it has become a powerful technology and may replace DNA microarray for transcriptome profiling (Wang, Gerstein, & Snyder, 2009). RNA-Seq technology is now being widely used for discovering multiple facets of transcriptome to analyse gene expression data. Moreover, it also produces visibility to previously undetected changes occurring in disease states, in response to therapeutics under different environmental conditions and other studies (Han, Gao, Muegge, Zhang, & Zhou, 2015). In fact, the cost of RNA-Seq is higher than DNA microarray but this technology can generate a very large quantity of sequencing data. Therefore, gene expression data are created by RNA-Seq that is also suitable for deep convolutional neural network (DCNN) and support vector machines (SVM) which use to classify RNA-Seq gene expression data.

Classifying RNA-Seq gene expression data has provided useful information for diagnosing cancer and drug discovery (Li et al., 2017). Gene expression can be simply defined as a function of one or more factors of the environment, lifestyle, and genetics. RNA-Seq technology has become a prevalent approach to quantify gene expression that is expected to gain better insights to a number of biological and biomedical questions, compared to DNA microarray technology (Johnson, Dhroso, Hughes, & Korkin, 2018). The processing of RNA-Seq gene expression includes many stages to obtain data matrix (RNASeqV2 level 3 expression data) (MIT and Harvard, 2016). The gene expression data matrix type contains gene expression values taken under given sampling conditions. In this data structure, each row represents a gene expression profile and each column corresponds to an RNA-Seq experiment. A characteristic of gene expression data is that the number of variables (genes) $n$ far exceeds the number of samples $m$, commonly known as 'curse of
dimensionality’ issue (Clarke et al., 2008). The issue leads to statistical and analytical challenges and conventional statistical methods which give improper result due to the high dimension of gene expression data with a limited number of patterns (Köppen, 2000). During the past decade, many algorithms have been used to classify gene expression data including support vector machines (SVM) (Furey et al., 2000), neural network (Khan et al., 2001), k nearest neighbours (Li, Weinberg, Darden, & Pedersen, 2001), decision trees (Netto et al., 2010), random forests (Díaz-Uriarte & De Andres, 2006), random forests of oblique decision trees (Do, Lenca, Lallich, & Pham, 2010), bagging and boosting (Dettling, 2004; Tan & Gilbert, 2003), and random ensemble oblique decision stumps (Huynh, Nguyen, & Do, 2018b). Although there have been many studies for classifying gene expression data, there remains a critical need for better accuracy improvement. DCNN and SVM are two successful approaches for pattern recognition (Christopher, 2016). On the one hand, the SVM has several advantages for classifying high-dimensional data. The main idea of this algorithm is to maximize the margin and to minimize an upper bound on the generalization error (Vapnik, 1995). SVM resolves this issue by convex optimization problem to find globally optimal solutions. Although SVM often outperforms other algorithms, it is shallow architecture models that has a single adjustable layer. On the other hand, deep convolutional neural network (DCNN) is deep architecture that learns latent representations. Tradition DCNN architecture uses the multinomial logistic regression (softmax activation) at the top layer for classifying. In fact, SVM is widely used alternative to softmax for classifying (Boser, Guyon, & Vapnik, 1992) to improve classification performance.

In this paper, we propose a new learning algorithm for precise classification of RNA-Seq gene expression data of SVM using features extracted by DCNN (call DCNN–SVM). The algorithm performs the training task with two steps. First of all, we use a new DCNN model to learn latent features from RNA-Seq gene expression data. The new features can improve the dissimilarity power of gene expression representations and thus obtain a higher accuracy rate than original features. Secondly, SVM is used to classify new features extraction from DCNN. Results of 25 small and medium RNA-Seq gene expression datasets as well as a large pan-cancer dataset (36 classes) from TCGA repository illustrate that DCNN–SVM is more accurate than the stage-of-the-art classifying models including deep convolutional neural network, linear support vector machines (LSVM) (Vapnik, 1998), and random forests (Breiman, 2001).

This paper is organized as follows. Section 2 discusses related works. Section 3 gives a brief overview of SVM, DCNN and our approach. Section 4 shows the results and the conclusions are presented in the final session.

2. Related works

In recent years, deep convolutional neural network (DCNN) has achieved remarkable results in computer vision (Krizhevsky, Sutskever, & Hinton, 2012), text classification (Kim, 2014). Besides, DCNN is also used for genomics signal processing and medical imaging instrumentation (Min, Lee, & Yoon, 2017). In recent years, the development of next generation sequencing technologies has contributed to big data infrastructure to support the future of personalized medicine. However, the classification algorithms often face several limitations in processing very-high-dimension data. To tackle this
issue, hybrid approaches are used to combine the advantages of feature extraction of DCNN and effective of support vector machines (SVM) or random forests (RF). In fact, this idea was used in image, text, signal classification. To emphasize this idea, we would like to mention that a hybrid model of multilayer perceptron and SVM was initially proposed in Suykens and Vandewalle (1999). Another model is proposed for handwritten digit recognition in Bellili, Gilloux, and Gallinari (2001) and Niu and Suen (2012). In these studies, the DCNN is used feature extraction and the SVM use a non-linear kernel to classify. It is noticeable that error classification rate has been gained by the hybrid model has achieved better results. In addition, the hybrid model of Nagi, Di Caro, Giusti, Nagi, and Gambardella (2012) has been used for recognition for mobile swarm robotic systems. Moreover, DCNN and RF are combined to build a hybrid model for electron microscopy images segmentation (Cao, Wang, Wei, Yin, & Yang, 2013). On the other hand, there are many studies use feature selection method to select important features, then these features use SVM or RF to classify. To illustrate this approach, in study (Guyon et al., 2002) which used a selection of relevant genes for sample classification. In addition, an unbiased feature selection method and random forests are used for high-dimensional data in Nguyen, Huang, and Nguyen (2015). A combined SVM and feature selection approach based on the method optimization involving the zero-norm has been proposed in Le Thi, Le, and Dinh (2015).

Although DCNN is a popular classification model, it is not common in research on RNA-Seq. Some related works are researched by Fakoor, Ladhak, Nazi, and Huber (2013), Urda, Montes-Torres, Moreno, Franco, and Jerez (2017). Therefore, we aim to use DCNN and SVM models to classify RNA-Seq gene expression data. Our approach differs from these previous ones as we build a coupling model instead of single classifiers. As far as we know, this method is not previously investigated for RNA-Seq gene expression data. The data in the relevant previous work is image data such as handwritten digit, medical image, and video datasets.

### 3. Methods

In this section, we briefly describe the deep convolutional neural network and the support vector machine. We focus on inspecting their internal structures to provide insights into their respective strengths and weaknesses on the present task. The below-mentioned analysis outlines reasons to propose our algorithm.

#### 3.1. Deep convolutional neural network

Deep convolutional neural network (DCNN) consists of many neural network layers. In network structure, the sequential layers are designed to learn progressively higher-level features, until the last layer which produces categories. In training process of network, all the layers are trained simultaneously and the feature extraction is an integral part of the classification system. Once training processing is completed, the last layer, which is a linear classified operating on the features extracted by the previous layers. Since those features are the result of the integrated training procedure and they have been optimized to satisfy the requirements and limitations of the last layer (Huang & LeCun, 2006).
DCNN is designed to process multiple data types, especially two-dimensional images that are directly inspired by the visual cortex of the human brain. In the human brain, there is a hierarchy of two basic cell types: simple cells and complex cells (Hubel & Wiesel, 1963). On the one hand, simple cells react to primitive patterns in sub-regions of visual stimulator. One the other hand, complex cells synthesize the information from simple cells to identify more intricate forms. Therefore, the visual cortex is a powerful and natural visual processing system. DCNN is applied to imitate three key ideas including local connectivity, invariance to location, and invariance to local transition (LeCun, Bengio, & Hinton, 2015).

The feature extraction can be seen as a building of stack of convolution layers and sub-sampling layers. The convolution layers compute convolutions over the previous layers $L_{in}$ with some small trainable convolution kernels $k$:

$$L_{out} = f\left(\sum L_{in} \ast k_i + b\right)$$

where $f$ is a non-linear function such as a hyperbolic tangent (TANH), rectified linear unit, and sigmoid. $b$ is bias. With each value of the kernel coefficients $k$, the convolution operation can implement a local edge detector or a low-pass filter. On each layer, multiple convolution kernels can be used to create several different feature maps.

The sub-sampling layers take the average or maximum of a $n \times n$ pixel block, multiply it by a trainable scalar $\beta$, add a bias, and pass the result through a sigmoid:

$$L_{out} = f\left(\sum L_{in}^{n\times n} + b\right)$$

The overall effect of these layers is extract a feature vector $v$ from the input $x$, written as $v = c(x)$. The last layer is a linear classification which operates from features extracted by the previous layers. This layer computes the product of the feature vector $v$ with a weight matrix $W$, adds a bias vector, and passes the result through activation functions. The classic DCNN uses softmax activation function at the top network. Gradient descent algorithms are used for the optimization during the training of a network.

3.2. Support vector machines

Support vector machine (SVM) was proposed by Vapnik (Vapnik, 1995), which is systematic and properly motivated by statistical learning theory. SVM is the most popular as the class of learning algorithms using the idea of kernel substitution. SVM and kernel-based methods have shown practical relevance for classification, regression (Burges, 1998). The SVM algorithm finds the best separating plane furthest from the different classes. In order to achieve this purpose, SVM algorithm tries to simultaneously maximize 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 Figure 1), with $m$ datapoints $x_i (i = 1, \ldots, m)$ in the $n$-dimensional input space $R^n$, having corresponding labels $y_i = \pm 1$. The SVM algorithms (Vapnik, 1995) try to find the best hyper-plane farthest from both $class +1$ and $class -1$. It can compute by simply maximize the distance or the margin between the supporting planes for each class ($x.w - b = +1$ for $class +1$, $x.w - b = -1$ for $class -1$). The
margin between these supporting planes is $2/\|w\|$ (where $\|w\|$ is the 2–norm of the vector $w$). Any point $x_i$ falling on the wrong side of its hyper plane is considered to be an error. Therefore, SVM has optimized the margin and minimize the error. The standard SVM with linear kernel is given by the following quadratic program (1):

$$
\min_{\alpha} \left( \frac{1}{2} \sum_{i=1}^{m} \sum_{j=1}^{m} y_i y_j \alpha_i \alpha_j K(x_i, x_j) - \sum_{i=1}^{m} \alpha_i \right)
$$

s.t.

$$
\sum_{i=1}^{m} x_i \alpha_i = 0
$$
$$
0 \leq \alpha_i \leq C \quad \forall i = 1, 2, \ldots, m
$$

where $C$ is a positive constant used to tune the margin and the error and a linear kernel function $K(x_i, x_j) = K(x_i, x_j)$.

The support vectors (for which $\alpha_i > 0$) are given by the solution of the quadratic program (1), and then, the separating surface and the scalar $b$ are determined by the support vectors. A new data point $x$ is classified based on the SVM model as follows:

$$
predict(x, SVM Model) = \text{sign} \left( \sum_{i=1}^{n_{SV}} y_i \alpha_i K(x_i, x_j) - b \right)
$$

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 Cristianini and Shawe-Taylor (2000).

For multiclass, one-versus-all (Vapnik, 1998) and one-versus-one (Kressel, 1999) 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 $k \times (k - 1)/2$ classifiers, using all the binary pairwise combinations of the $k$ classes. After that the class is predicted with a majority vote.
SVM has been successfully applied to high-dimensional problems arising. For high-dimensional data, conventional classifiers like logistic regression, maximum likelihood classification, etc. tend to overfit the model using training data and run the risk of achieving lower accuracies on validation data. Similar observations were reported in the field of microarray gene expression data classification in Pirooznia, Yang, Yang, and Deng (2008), Statnikov, Wang, and Aliferis (2008) where SVM was trained directly on the original high-dimensional input spaces. SVM was employed and compared with other classifiers like k nearest neighbours, C4.5 decision trees. The results show that SVM outperforms the traditional algorithms.

3.3. Classifying gene expression data of SVM using features extracted by DCNN

The proposed algorithm is an effective combination of two algorithms DCNN and SVM. The algorithm performs the training task with two main phases. The workflow of our method is shown in Figure 2.

Our model has taken advantage of DCNN by learning latent features from very-high-dimensional input spaces. This process can be viewed as a projection of data from higher dimensional space to a lower dimensional space. These new features can improve the dissimilarity power of gene expression representations and thus obtain higher accuracy rate than original features. Conversely, although the non-linear SVM can not learn complex invariances, it produces good decision surfaces by maximizing margins using soft-margin approaches (Huang & LeCun, 2006). The combination of DCNN and SVM has been proposed by Huynh et al. (2018a) in the past as part of a deep learning process.

First of all, we implement a new DCNN that extracts new features from origin RNA-Seq gene expression data. To extract features, we design a new architecture of DCNN to extract latent features from RNA-Seq gene expression data. We found global average pooling increased model stability. Its architecture consists of two convolutional layers, two pooling layers, and a fully connected layer (Figure 3). The layers are respectively named Conv1, kernel size: (3x3) 4 feature maps, Pooling1, kernel size: (3x3), Conv2, kernel size: (3x3) 2 feature maps, Pooling2, kernel size: (3x3), and the output layer uses non-linear support vector machine with RBF kernel classify.

Figure 2. The workflow of our method.

Figure 3. In this illustration, the convolutional layers already include non-linearities and, thus, a convolutional layer actually represents four layers. The feature maps of the final sub sampling layer are then fed into the actual classifier consisting of an arbitrary number of fully connected layers. The output layer uses non-linear support vector machine with RBF kernel classify.
Conv1, Pooling1, Conv2, Pooling2, and output (numbers indicate the sequential position of the layers). The input layer receives the gene expression in the 2-D matrix format. We embedded the high-dimension expression data (20531x1) into a 2-D image (142x142) by adding some zeros at the last line of the image. The first Conv1 layer contains 4 feature maps and kernel size (3x3). The second layer, Pooling1 layer, is taken as input of the average pooling output of the first layer and filter with (2x2) sub-sampling layer. Conv2 uses convolution kernel size (3x3) to output 2 feature maps Pooling2 is a (2x2) sub-sampling layer. We propose to use the Tanh activation function as neurons. The final layer has a variable number of maps that combine inputs from all map in Pooling2. The output of networks is a 2312 dimensional vector. These vectors are the features extracted by sequence of Conv1 to Pooling2 layers.

Last but not least, the new feature extracted by DCNN following which the SVM learns to classify gene expression data efficiently in this phase. In our approach, we propose to use RBF kernel type in SVM model because it is general and efficient (Burges, 1998).

In our algorithm, DCNN is used to extract features. The training and testing samples are fed through the trained network, and the output of the last layer are extracted as the features. The feature sets from the training samples are then used as input to train a SVM in the usual way. We find that this hybrid system not only improve the classification performance significantly, but also are computed very efficiently.

4. Evaluation

We are interested in the classification performance (accuracy and training time) of our proposal (DCNN–SVM) for classifying RNA-Seq gene expression data. Therefore, we report the comparison of the classification performance obtained by DCNN–SVM and the best state-of-the-art algorithms including linear support vector machines (LSVM) (Vapnik, 1995), random forests (RF) (Breiman, 2001), and deep convolutional neural network (DCNN). In addition, we also compare various version of DCNN–SVM (DCNN–LSVM, DCNN–RF) with LSVM, RF. These results are used to evaluate performance of classifiers after using DCNN.

In order to evaluate the effectiveness in classification tasks, we have implemented DCNN–SVM and its version in Python using TensorFlow (Abadi et al., 2015) and Scikit (Pedregosa, 2011) libraries. Random forests of C4.5 decision trees use Scikit library (Pedregosa, 2011). We use the highly efficient standard SVM algorithm LibSVM (Chang & Lin, 2011) with one-versus-all for multi-class. Student’s test is used to access classification results of algorithms.

All experiments are run on machine Linux Mint, Intel(R) Xeon(R) CPU 3.07GHz, 8 cores and 8 RAM.

4.1. Datasets

To evaluate performance of our approach on various dataset sizes, we use RNA-Seq gene expression datasets from TCGA data portal (January 2016). Data retrieved from TCGA (http://gdac.broadinstitute.org).

Regard on datasets for binary class classification, we use 25 datasets which are small- and medium-sample sizes ranging from 66 to 1100 samples. Each sample has 20,531 features. Table 1 shows the overall description of 25 datasets.
Besides the full pan-cancer dataset is used, which contains 12,181 data samples representing 36 tumour types. It contains all RNA-Seq level 3 expression data for 11,355 patients representing 35 tumour types and for 826 normal samples. The large dataset contains the new updated samples of tumours in RNA-Seq gene expression data. Recently, there have been few studies about classification of tumours on this dataset.

### 4.2. Experiments setup

As for 25 datasets, we use 10-fold cross-validation protocol that remains the most widely to evaluate the performance (Kohavi, 1995). The total classification accuracy measure is used to evaluate the classification models.

On the large pan-cancer dataset, we randomly divide the data into a training set (75% of the samples) and a testing set (25% of the samples) with samples drawn proportionally from each tumour type without replacement.

Regard to DCNN–SVM configuration, the gene expression matrix is scaled to the range $[-0.9, 0.9]$. To training model, we use Adam for optimization (Kingma & Ba, 2014) with batch size is 16 to 128. We start to train with a learning rate of 0.00002 for all layers, and then increased it manually every time when the validation error rate stopped improving. We use cross-entropy for loss function, the epochs size is 200. However, the network structure hyper-parameters could not be fixed across tasks. Therefore, for DCNN, we examine a variety of architectures on the validation set, and pick the one with the best performance, then re-train the whole network on the training set and report the test accuracy. We tune the hyper-parameter of RBF kernel and the cost $C$ (a trade off between the margin size and the errors) to obtain the best correctness. The cost $C$ is chosen in $1, 10, 10^2, 10^3, 10^4$ and the hyper-parameter $\gamma$ of RBF kernel is tried among $1.E - 1, 1.E - 2, 1.E - 3, 1.E - 4, 1.E - 5$.

With other algorithms, the cost constant $C$ of LSVM is set to $10^3$. The number of trees in random forests is 200 trees.

### 4.3. Classification results of small and medium 25 RNA-Seq gene expression datasets

Table 2 gives results of classifying algorithms on 25 binary-class gene expression datasets. The plot chart in Figures 4 and 5 also visualize classification results. The results of DCNN–
SVM show in column 6 of Table 2. Results of other algorithms are shown in columns 5 to 7 respectively.

Overall, using paired Student ratio test, it is clear that DCNN–SVM outperformed LSVM, RF and DCNN. DCNN–SVM has 18 wins, 5 ties, 2 defeats (\( p\)-value = .0046) when compared to LSVM in column 1 on Table 2. In addition, DCNN–SVM has 18 wins, 6 ties, 1 defeat (\( p\)-value = .0012) compared to RF in column 2. Besides, when compared to DCNN, our proposed model is outperformed 15 out of 25 datasets (15 wins, 1 tie, 9 defeats, \( p\)-value = .034).

**Figure 4.** Classification accuracy comparison of DCNN–SVM, linear SVM and random forests on 25 RNA-Seq gene expression datasets.
Theses column 5, 6 and 7 of Table 2 show the performance of three classifiers (LSVM, SVM and RF). The RBF kernel function is quite good for LSVM (21 defeats, 4 wins) and better than RF (18 wins, 6 ties, 1 defeat). The parameters of $C$ and $\gamma$ of DCNN–SVM have been tuned and the best values of $C$ and $\gamma$ are $10^4$ and $10^{-4}$ on most datasets except 4 datasets which are ESCA ($C = 10$, $\gamma = 0.001$), LIHC ($C = 1$, $\gamma = 0.01$), KIRC ($C = 1$, $\gamma=0.01$) and STES ($C = 100$, $\gamma = 0.001$).

Figure 4 shows that our method archives a higher performance for the large dataset. It is true that with small sample sizes RNA-Seq gene expression datasets, classifying algorithms can also face the overfitting problem. Therefore, DCNN–SVM’s performance has more effect on medium and large datasets. In fact, with datasets from 13 to 25, the proposed model has classification results outperformed the other models.

The running time of a hybrid model includes two parts: the time to train the deep convolutional networks for extracting the features, and the training time for the classifier on the new extracted feature set. The average time of the first part on 25 datasets is 80.72 s. The average time of the second part for SVM with RBF kernel, linear SVM, RF in the hybrid model are, respectively, 0.07, 0.24 and 1.12 s. While the running time of linear SVM and RF are 10.85 and 4.98 s.

4.4. Classification results of the large RNA-Seq gene expression dataset

Table 3 and Figure 6 show results of DCNN–SVM, various versions of DCNN–SVM (DCNN–LSVM, DCNN–RF), LSVM, RF and DCNN on the large RNA-Seq gene expression dataset. It is clear that DCNN–SVM has the best accuracy, the next model is DCNN–LSVM. On this dataset, DCNN–SVM has higher accuracy than other models because it is take advantage of DCNN and SVM. In addition, between various version of DCNN–SVM, DCNN–SVM has the

| Dataset  | Linear SVM | RF   | DCNN | DCNN–LSVM | DCNN–SVM | DCNN–RF |
|----------|------------|------|------|-----------|----------|---------|
| Pan-Cancer | 72.39      | 73.44| 74.09| 76.03     | 76.33    | 72.05   |
best classifier performances (76.33%), because of efficient of RBF kernel. It is observed that our proposed DCNN and SVM combined outperforms the DCNN, LSVM and RF. These results clearly shows that DCNN–SVM is the best classification model on large RNA-Seq gene expression datasets.

For TCGA dataset, the time for learning the DCNN and for extracting the features is 64 minutes and 25 s. The time to train the classifiers of SVM with RBF kernel, LSVM, RF in the hybrid model are, respectively, 3 min and 8 s, 2 min and 55 s, 3 min and 16 s. While the running time of LSVM and RF are 43 and 18 min. With the significant dimension reduction in the hybrid model, the time to train the classifier is reduced.

5. Conclusions

We have presented a hybrid model combining DCNN and non-linear SVM to classify very-high-dimensional RNA-Seq gene expression data. The DCNN extracts latent features from gene expression data, then they are used in conjunction with SVM that efficiently classify RNA-Seq gene expression data. The latent features are learned through a convolution process and then sent as input to the SVM classifier using RBF kernel. After modifications through specified hyper parameters, this model performs quite comparatively well on the task tested on RNA-Seq gene expression datasets from The Cancer Genome Atlas (TCGA). Numerical test results on RNA-Seq gene expression datasets from TCGA show that our proposed approach DCNN–SVM has the most accurate, when compared to classical DCNN, SVM and random forests.

In the near future, we will attempt to build larger architectures so as to further improve the classification accuracy, while taking advantage of the constant increase of available computational power. Our proposal can be effectively paralleled. A parallel implementation that exploits Graphical Processing Units (GPUs) can greatly speed up the learning and predicting tasks.

Disclosure statement

No potential conflict of interest was reported by the authors.
Notes on the contributors

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 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.

ORCID

Phuoc-Hai Huynh http://orcid.org/0000-0001-8348-9267
Van-Hoa Nguyen http://orcid.org/0000-0003-0582-2129
Thanh-Nghi Do http://orcid.org/0000-0002-9328-3358

References

Abadi, M., Agarwal, A., Barham, P., Brevdo, E., Chen, Z., Citro, C., & Ghemawat, S. (2015). TensorFlow: Large-scale machine learning on heterogeneous systems. Software available from tensorfl ow.org. Aluru, S. (2005). Handbook of computational molecular biology. Chapman & Hall/CRC.

Bellili, A., Gilloux, M., & Gallinari, P. (2001). An hybrid MLP-SVM handwritten digit recognizer. In Proceedings of sixth international conference on document analysis and recognition (pp. 28–32). IEEE.

Bhattacharjee, A., Richards, W. G., Staunton, J., Li, C., Monti, S., Vasa, P., & Loda, M. (2001). Classification of human lung carcinomas by mRNA expression profiling reveals distinct adenocarcinoma subclasses. Proceedings of the National Academy of Sciences, 98(24), 13790–13795.

Boser, B. E., Guyon, I. M., & Vapnik, V. N. (1992). A training algorithm for optimal margin classifiers. In Proceedings of the fifth annual workshop on computational learning theory, COLT ’92 (pp. 144–152). New York, NY: ACM.

Breiman, L. (2001). Random forests. Machine learning, 45(1), 5–32.

Burges, C. J. (1998). A tutorial on support vector machines for pattern recognition. Data mining and knowledge discovery, 2(2), 121–167.

Cao, G., Wang, S., Wei, B., Yin, Y., & Yang, G. (2013). A hybrid CNN-RF method for electron microscopy images segmentation. J Biomim Biomater Tissue Eng, 18, 2.

Chang, C.-C., & Lin, C.-J. (2011). Libsvm: A library for support vector machines. ACM Transactions on Intelligent Systems and Technology, 2(3), 27:1–27:27.

Christopher, M. B. (2016). Pattern recognition and machine learning. New York, NY: Springer-Verlag.

Clarke, R., Ressom, H. W., Wang, A., Xuan, J., Liu, M. C., Gehan, E. A., & Wang, Y. (2008). The properties of high-dimensional data spaces: implications for exploring gene and protein expression data. Nature Reviews Cancer, 8(1), 37.
Cristianini, N., & Shawe-Taylor, J. (2000). *An introduction to support vector machines and other kernel-based learning methods*. Cambridge, U.K: Cambridge University Press.

Dettling, M. (2004). BagBoosting for tumor classification with gene expression data. *Bioinformatics (Oxford, England)*, 20(18), 3583–3593.

Díaz-Uriarte, R., & De Andres, S. A. (2006). Gene selection and classification of microarray data using random forest. *BMC Bioinformatics*, 7(1), 3.

Do, T.-N., Lenca, P., Lallich, S., & Pham, N.-K. (2010). Classifying very-high-dimensional data with random forests of oblique decision trees. *Advances in Knowledge Discovery and Management* (pp. 39–55). Springer, Berlin, Heidelberg.

Fakoor, R., Ladhak, F., Nazi, A., & Huber, M. (2013). Using deep learning to enhance cancer diagnosis and classification. In *Proceedings of the international conference on machine learning* (Vol. 28). New York, USA: ACM.

Furey, T. S., Cristianini, N., Duffy, N., Bednarski, D. W., Schummer, M., & Haussler, D. (2000). Support vector machine classification and validation of cancer tissue samples using microarray expression data. *Bioinformatics (Oxford, England)*, 16(10), 906–914.

Golub, T. R., Slonim, K. D., Tamayo, P., Huard, C., Gaasenbeek, M., Mesirov, J. P., & Caligiuri, M. A. (1999). Molecular classification of cancer: Class discovery and class prediction by gene expression monitoring. *Science (New York, N.Y.)*, 286(5439), 531–537.

Guyon, I., Weston, J., Barnhill, S., & Vapnik, V. (2002). Gene selection for cancer classification using support vector machines. *Machine learning*, 46(1–3), 389–422.

Han, Y., Gao, S., Muegge, K., Zhang, W., & Zhou, B. (2015). Advanced applications of RNA sequencing and challenges. *Bioinformatics and Biology Insights*, 9, BBI–S28991.

Huang, F., & LeCun, Y. (2006). Large-scale learning with svm and convolutional network for generic object recognition. In *2006 IEEE computer society conference on computer vision and pattern recognition* (pp.284-291). IEEE Computer Society.

Hubel, D. H., & Wiesel, T. (1963). Shape and arrangement of columns in cat’s striate cortex. *The Journal of Physiology*, 165(3), 559–568.

Huynh, P.-H., Nguyen, V. H., & Do, T.-N. (2018a). 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* (pp. 233–243). Springer, Cham.

Huynh, P.-H., Nguyen, V. H., & Do, T.-N. (2018b). Random ensemble oblique decision stumps for classifying gene expression data. In *Proceedings of the ninth international symposium on information and communication technology*, SoICT 2018 (pp. 137–144). New York, NY: ACM.

Johnson, N. T., Dhroso, A., Hughes, K. J., & Korkin, D. (2018). Biological classification with RNA-seq data: Can alternatively spliced transcript expression enhance machine learning classifiers?. *RNA (New York, N.Y.)*, 24(9), 1119–1132.

Khan, J., Wei, J. S., Ringner, M., Saal, L. H., Ladanyi, M., Westermann, F., & Meltzer, P. S. (2001). Classification and diagnostic prediction of cancers using gene expression profiling and artificial neural networks. *Nature Medicine*, 7(6), 673.

Kim, Y. (2014). Convolutional neural networks for sentence classification. In *Proceedings of the 2014 conference on empirical methods in NLP (EMNLP)* (pp. 1746–1751). Doha, Qatar.

Kingma, D. P., & Ba, J. (2014). Adam: A method for stochastic optimization. In *Proceedings of the 3rd international conference on learning representations* (ICLR). San Diego, CA, USA.

Kohavi, R. (1995). A study of cross-validation and bootstrap for accuracy estimation and model selection. In *IJCAI* (Vol. 14, pp. 1137–1145). Montreal, Canada.

Köppen, M. (2000). The curse of dimensionality. In *5th online world conference on soft computing in industrial applications* (WSCS) (pp. 4–8). IEEE Finland Section.

Kressel, U. H.-G. (1999). Advances in kernel methods. chapter Pairwise Classification and Support Vector Machines (pp. 255–268). Cambridge, MA: MIT Press.

Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2012). Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, (pp. 1097–1105). Curran Associates, Inc.
Kukurba, K. R., & Montgomery, S. B. (2015). RNA sequencing and analysis. *Cold Spring Harbor protocols*, 2015(11), 951–969.

LeCun, Y., Bengio, Y., & Hinton, G. (2015). Deep learning. *Nature*, 521(7553), 436.

Le Thi, H. A., Le, H. M., & Dinh, T. P. (2015). Feature selection in machine learning: An exact penalty approach using a difference of convex function algorithm. *Machine Learning*, 101(1–3), 163–186.

Li, Y., Kang, K., Krahm, J. M., Croutwater, N., Lee, K., Umbach, D. M., & Li, L. (2017). A comprehensive genomic pan-cancer classification using the cancer genome atlas gene expression data. *BMC Genomics*, 18(1), 508.

Li, L., Weinberg, C. R., Darden, T. A., & Pedersen, L. G. (2001). Gene selection for sample classification based on gene expression data: study of sensitivity to choice of parameters of the GA/KNN method. *Bioinformatics (Oxford, England)*, 17(12), 1131–1142.

Min, S., Lee, B., & Yoon, S. (2017). Deep learning in bioinformatics. *Briefings in Bioinformatics*, 18(5), 851–869.

MIT and Harvard (2016). Broad institute TCGA genome data analysis center (2016): Analysis overview for 28 January 2016.

Nagi, J., Di Caro, G. A., Giusti, A., Nagi, F., & Gambardella, L. M. (2012). 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. Boca Raton, Florida, USA.

Netto, O. P., Nozawa, S. R., Mitrowsky, R. A. R., Macedo, A. A., Baranauskas, J. A., & Lins, C. (2010). Applying decision trees to gene expression data from DNA microarrays: A leukemia case study. In XXX congress of the Brazilian computer society, X workshop on medical informatics (p. 10). Belo Horizonte, MG.

Nguyen, T.-T., Huang, J. Z., & Nguyen, T. T. (2015). Unbiased feature selection in learning random forests for high-dimensional data. *The Scientific World Journal*, 2015, 471371.

Pedregosa, F. (2011). Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12, 2825–2830.

Reis-Filho, J. S., & Pusztai, L. (2011). Gene expression profiling in breast cancer: Classification, prognostication, and prediction. *The Lancet*, 378(9805), 1812–1823.

Snyder, M. (2016). *Genomics and personalized medicine: What everyone needs to know*. New York: Oxford University Press.

Statnikov, A., Wang, L., & Aliferis, C. F. (2008). A comprehensive comparison of random forests and svm for microarray-based cancer classification. *BMC Bioinformatics*, 9(1), 319.

Vapnik, V. (1995). *The nature of statistical learning theory*. Berlin, Heidelberg: Springer science & business media.

Vapnik, V. (1998). An overview of statistical learning theory. *IEEE Transactions on Neural Networks*, 10(5), 988–999.

Wang, Z., Gerstein, M., & Snyder, M. (2009). RNA-seq: A revolutionary tool for transcriptomics. *Nature Reviews Genetics*, 10(1), 57.

Zhao, S., Fung-Leung, W.-P., Bittner, A., Ngo, K., & Liu, X. (2014). Comparison of RNA-seq and micro-array in transcriptome profiling of activated t cells. *PloS One*, 9(1), e78644.