Lightweight Convolutional Neural Network for Breast Cancer Classification Using RNA-Seq Gene Expression Data

Article *in* IEEE Access · December 2019

DOI: 10.1109/ACCESS.2019.2960722

CITATIONS
25

READS
928

4 authors, including:

Murtada Khalafallah Elbashir
Al-Jouf University
37 PUBLICATIONS **96** CITATIONS

SEE PROFILE

Mohanad Mohammed
University of KwaZulu-Natal
16 PUBLICATIONS **67** CITATIONS

SEE PROFILE

Said Saleh Saloum
Jouf University
2 PUBLICATIONS **32** CITATIONS

SEE PROFILE

IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received November 21, 2019, accepted December 10, 2019, date of publication December 18, 2019,
date of current version December 31, 2019.

Digital Object Identifier 10.1109/ACCESS.2019.2960722

Lightweight Convolutional Neural Network for Breast Cancer Classification Using RNA-Seq Gene Expression Data

MURTADA K. ELBASHIR[1,2], MOHAMED EZZ[1,3], MOHANAD MOHAMMED[2,4], AND SAID S. SALOUM[1]

[1]College of Computer and Information Sciences, Jouf University, Sakaka 72441, Saudi Arabia
[2]Faculty of Mathematical and Computer Sciences, University of Gezira, Wad Madani 11123, Sudan
[3]Faculty of Engineering, Al-Azhar University, Cairo 11651, Egypt
[4]School of Mathematics, Statistics and Computer Science, University of KwaZulu-Natal, Pietermaritzburg 3209, South Africa

Corresponding author: Murtada K. Elbashir (mkelfaki@ju.edu.sa)

This work was supported by the Jouf University under Project 39/751.

ABSTRACT Gene expressions are considered among the most used features in cancer classification. The available gene expression data has a small number of samples and a relatively big number of dimensions, and that makes it not suitable for deep Convolutional Neural Networks (CNN) architectures, which exhibit state-of-the-art performance in many fields. In this paper, we propose a lightweight CNN architecture for breast cancer classification using gene expression data downloaded from Pan-Cancer Atlas using ''Illumina HiSeq'' platform. The downloaded gene expression data is preprocessed and then transformed into 2D-images. We started the preprocessing by removing the outlier samples, which are determined based on the Array-Array Intensity Correlation (AAIC), which defines a symmetric square matrix of Spearman correlation. Then we applied a normalization process on the gene expression data to ensure that we can infer the expression level from it correctly and avoid biases in the expression measures. Finally, filtering is applied on the data. Model selection or a parameters search strategy is conducted to choose the values of the CNN hyper-parameters that give optimal performance. Our experiments show that our proposed method achieves an accuracy of 98.76%, which is the highest compared to other competing methods.

INDEX TERMS Tumor type classification, RNA-Seq, gene expression, convolutional neural network, edge detection.

I. INTRODUCTION

Biological systems' physiological status and gene activities can be revealed at the transcriptome level by gene expression i.e. the actively expressed genes at any given time are reflected by transcriptome [1]. The transcriptome of an organism can be measured using RNA-Seq or DNA microarrays [2]. It is used to denote all RNAs or just mRNA, which is a messenger RNA molecule. These molecules transfer genetic information from DNA, which encodes all the information needed to specify the features and functions of every single cell [3] to the ribosome. Ribosomes connect amino acids, which are the building blocks of the protein together in the order specified by mRNA that encode proteins through the genetic code [4]. RNA-Seq measures the transcription of a specific gene by converting long RNAs into a library of complementary DNA (cDNA) fragments that generate an expression profile. The genes that play a main role in the specification of the phenotype can be identified by comparing gene expression profiles from different tissues. i.e. comparing the diseased with the healthy tissues can reveal new insights over the genetic variables involved in pathology. Therefore, gene expression data can provide researchers with features that can be analyzed using computational methods to discover gene regularity targets, diagnosis disease, and develop drug [5]–[7]. Studies show that these data can provide very important information regarding tumor characteristics, which provide options for the treatment, care, and management of the patient [5], [6], [8]. Tumor characteristics offered deep insight into cancer detection problems [9]. Identifying genes that are highly expressed in tumor cells but not in normal ones using gene expression data is considered as a challenge

The associate editor coordinating the review of this manuscript and approving it for publication was Ran Su.

that needs to be addressed using computational methods. Gene expression data itself revealed other challenges for the use of the computational methods due to the high dimensionality associated with these data and the relatively very small number of samples as well as the high amount of noise [8], [10].

Many unsupervised and supervised learning methods have been developed for cancer classification using gene expression data. The unsupervised methods include the work of Alon *et al.* [11] and Li *et al.* [12], which use cluster analysis and decision trees for cancer classification respectively. Also, K-nearest neighbors (KNN) as unsupervised learning is used for breast cancer classification [13]. The supervised learning methods that are designed for cancer classification can be divided into statistical-based methods [14] and machine learning methods. The statistical methods include the work of Guha *et al.* [15], who proposed a two-stage method using hidden Markov Model (HMM) and Bayesian linear variable selection procedures for cancer classification using copy number data. The machine learning methods include the work of Gregory and William [16], and the work of Dwivedi [17], which are based on artificial neural network, besides support vector machines methods which include the works of Furey *et al.* [18] and Vanitha *et al.* [19].

Recently, a part of a broader family of machine learning known as deep learning, which uses more complicated algorithms to model the features of the data with a high level of abstraction has achieved state of the art in many classification fields such as image classification [20]–[22], and speech recognition [23]. The algorithms of deep learning normally use raw features from large data set such as a collection of images and utilize these features to create a model based on hidden patterns buried inside and organized in many levels that learn to convert the input data into composite and more abstract representation [24], [25].

In spite of the fact that the application of deep learning in bioinformatics is limited due to the huge amount of dimensions compared to the relatively small size of samples [26], [27], there are many methods that use deep learning for cancer classification. These methods include the work of Kong and Yu [28] who presented a deep feedforward network classifier embedding feature graph information for cancer classification using gene expression data. Their network architecture consists of an input layer, graph embedded layer with ReLU activation function, two hidden layers with ReLU activation functions, and an output layer with softmax activation function. Lyu and Haque [29] embedded gene expression data into 2D-image of size 102×102 to be used as input for a Convolutional Neural Network (CNN) to classify the tumor types. Sevakula *et al.* [30] presented a method based on transfer learning for cancer classification, they used feature selection and normalization techniques in conjunction with stacked sparse auto-encoders on gene expression data. Danaee *et al.* [8], presented a deep learning approach for cancer detection and identification of genes critical for breast cancer diagnosis. In their method, they deeply extracted functional features from high dimensional gene expression data using Stacked Denoising Autoencoder then they used supervised classification models to evaluate the performance of the extracted representation. Then they analyzed the Stacked Denoising Autoencoder connectivity matrices to identify a set of highly interactive genes.

In this paper, we constructed a lightweight CNN architecture for breast cancer classification using RNA-Seq gene expression data. The gene expression data is downloaded from Pan-Cancer Atlas [31] using ''Illumina HiSeq'' platform under R software via the TCGAbiolinks package [32]. The downloaded gene expression data is preprocessed and then transformed into 2D-images. We started the preprocessing by removing the outlier samples, which are determined based on the Array-Array Intensity Correlation (AAIC), which defines a symmetric square matrix of Spearman correlation between samples. Then we applied a normalization process on the gene expression data to ensure that we can infer the expression level from it correctly and avoid biases in the expression measures. Finally, filtering is applied on the data. To determine the best values of the CNN hyper-parameters that give optimal performance, we conducted a parameters search strategy using a grid search. Our constructed CNN shows improvement compared with previous work on breast cancer classification using Pan-Cancer Atlas gene expression data.

II. MATERIALS AND METHODS

A. DATA SET

The gene expression data set for breast cancer is downloaded from Pan-Cancer Atlas [31] using R studio, where the query is formulated using the GDCquery function of the TCGAbiolinks library [32]. GDC is an abbreviation for NCI's Genomic Data Commons that provides the research community with a unified data storage that enables data sharing across cancer genomic studies. There are many parameters that should be passed to the GDCquery function. These parameters include (project, legacy, data.category, data.type, platform, file.type, experimental.strategy, and sample.type). The project argument refers to the project that should be downloaded from the list of valid projects in the Pan-Cancer Atlas. For breast cancer, the value for this argument will be ''TCGA-BRCA''. The legacy argument can be set to true or false, in our case we set it to be true, which means that the query should look into the legacy repository when fetching the data. The legacy repository will make the unmodified copy of data that was previously stored in the TCGA Data Portal available for download. Each project has a different data category, thus data.category refers to the category of the data in the specific project. In our case, we should set the data category to ''Gene expression''. The data.type argument refers to the data type that we can use for filtering the files to download. In our case, we set it to ''Gene expression quantification''. Gene can be quantified by counting the number of reads that map to each gene using RNA-Seq [33], [34]. For the platform, we can select one of several platforms and the value

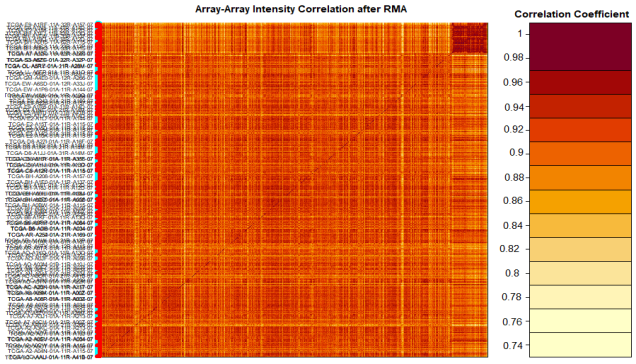

FIGURE 1. Array-Array Intensity Correlation (AAIC) visualization.

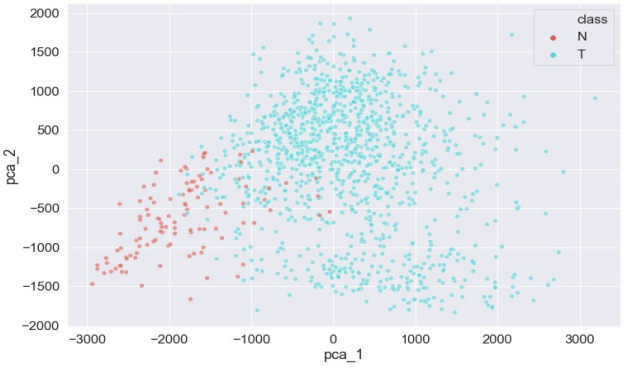

FIGURE 2. Gene expression data visualization using PCA.

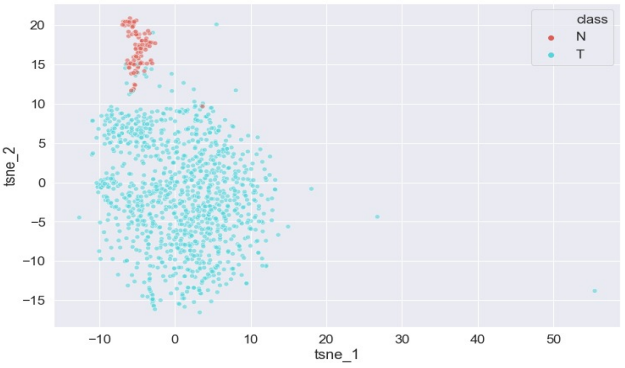

FIGURE 3. Gene expression data visualization using t-SNE.

that we have chosen is ''Illumina HiSeq''. File type argument refers to the type that can be used in the legacy database. We used ''results'' as a value for the file.type argument. There are different experimental strategies that we can select from e.g., RNA-Seq, miRNA-Seq, and Genotyping Array. We have selected RNA-Seq to generate our expression profile. Such a profile can be called RNA-Seq gene expression profile [35], [36]. Finally, the sample type refers to the type of sample that can be used for filtering the data to download. In our case, we set the sample type to ''c(''Primary solid Tumor'',''Solid Tissue Normal'')'' i.e. to download the gene expression associated with normal cases and cases with a tumor only. The downloaded BRCA data is transformed into a form of a matrix. The columns in this matrix represent the samples or the cases and the rows represent the genomic ranges of interest [37], [38]. The BRCA data set has 1208 clinical samples with respect to 19948 genes. Because of this large number of genes i.e. the dimensions, it is important to reduce them, because this large number of genes will make the data vulnerable to noise, which can affect the performance of the classifier. Therefore, preprocessing steps are applied to the BRCA data in order to reduce the number of genes and select the genes that contribute positively to the accuracy of the classification.

B. DATA PREPROCESSING

We started the preprocessing step by constructing AAIC, which defines a symmetric square matrix of Spearman correlation between samples to identify problematic arrays [39]. We visualized the AAIC as shown in Figure 1. The colors represent the strength of the correlation between the samples. A higher correlation is indicated in a dark color and lower correlation in a light color. According to this symmetric matrix, the samples that are considered as outliers will be removed. The correlation cut-off equal to 0.6 is used to determine the outliers, then normalization process is applied to BRCA gene expression data to ensure that we can infer the expression level from it correctly and avoid biases in the expression measures [40]–[43]. TCGAanalyze_Normalization function of the TCGAbiolinks library is used to carry out the normalization process. The method we used in this function is GC-content, which refers to the proportion of nucleotides in the strand of nucleic acid that holds either guanine (G) or cytosine (C). Therefore, the normalization process will remove the dependence in which GC-richer genes tend to be truly DE or the strong CG-Content bias [40]. We finalized the data preprocessing step by applying filtering on our gene expression data set using threshold value equal to 0.25 i.e. to select mean values across all samples that are higher than 0.25. The final obtained data set after the preprocessing has 1208 clinical samples with respect to 14477 genes. In the data set, there are 113 Negative samples and 1095 positive samples.

Before we start applying our lightweight CNN on the obtained data after the preprocessing, we visualized our preprocessed data by projecting it in a low dimensional space using Principal Component Analysis (PCA), which is a linear projection and t-Distributed Stochastic Neighbor Entities (t-SNE), which is a non-linear projection to capture the linear and non-linear dependencies as depicted in Figure 2 and Figure 3 respectively. From these figures, it is clear that our preprocessed data forms two clusters according to the class (Normal (N), Tumor (T)), which means that it contains the genes that discriminate between the normal and tumor cases. Because of the non-linearity in the gene expression, we see clearly that t-SNE performs better than PCA, but we can not rely on the figure to discriminate between the positive and negative classes because there are points that are hidden due to the overlapping.

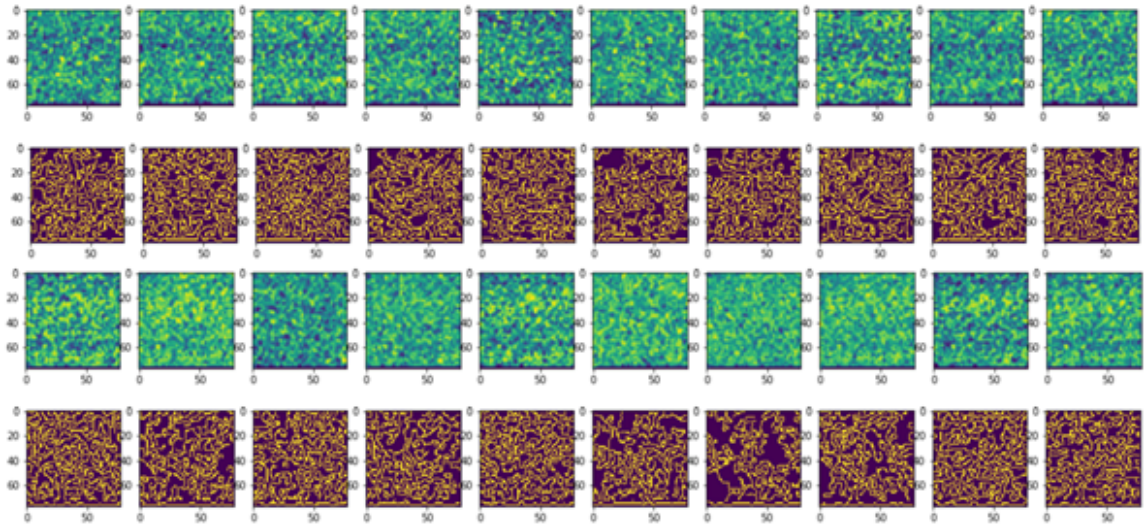

FIGURE 4. A sample of the 2D-images that are obtained from the data set. The first and the second rows represent the negative class before and after edge detection. The third and fourth rows represent the positive class before and after edge detection.

C. TRANSFORMING THE GENE EXPRESSION INTO 2D-IMAGES

The preprocessed data are transformed into 2D-images by reshaping them into a matrix of size (127×114) to be suitable for the use of the convolutional layer of the CNN. Transformation of data into images is adopted by many researches e.g. [29]. The data set contains a number of columns that cannot be transformed into (127×114) 2D-matrix, therefore, we appended the last column of the array with zeroes. This modification of array is commonly used to make the size of the input adjusted to the requirement. We applied the edge detection technique to have a better visualization for the obtained 2D-images. We visualized 10 negative (primary melanomas) and 10 positive (melanoma metastases) samples before and after edge detection as depicted in Figure 4. From the figure, it is clear that there is a recognizable pattern that can differentiate the two classes.

D. CONVOLUTIONAL NEURAL NETWORK

Convolutional Neural Networks (CNN) are a classes of neural networks mostly applied for image classification. They were inspired by the biological process, where the connectivity between neurons is similar to the arrangement of the animal visual cortex. In CNN, spatial and temporal dependencies in an image can be captured by using relevant filters. The CNN normally reduces the image features into an arrangement that is easier to process without dropping features which are very important for getting a good classification. The architecture of the CNN consists of a sequence of layers, where each layer uses a differentiable function to transform one volume of activations to another. Normally three types of layers are stacked to build a CNN model; these layers are: convolutional layer, pooling layer, and fully-connected layer. Two types of pooling can be used in the pooling layer. The first type is the max pooling, which returns the maximum value from the part of the image covered by the kernel and it removes the noisy activations and performs de-noising beside dimensionality reduction. The second type is the average pooling, which simply reduces the dimensions and it uses this reduction as a noise suppressing mechanism [44]. Therefore, max pooling is better than average pooling when it comes to performance. Usually, CNN takes tensor of order 3 as input i.e. an image matrix of n rows, m columns, and 3 color channels (R, G, and B) and it takes the spatial structure of the images into account. The input goes through the convolution layer, pooling layer, and the fully-connected layer sequentially, where the output of each layer acts as input for the layer that comes after it. The network starts with 3*m*n input neurons which are used to encode the pixel intensities for the input features of the 2D-image followed by a convolutional layer using a f $\times$ f local receptive field or filter size and 3 feature maps that represents color channels. The result is a layer of 3 $\times$ (m-f+1) $\times$ (n-f+1) hidden feature neurons when a stride of 1 is used. The pooling layer will be applied to 2 $\times$ 2 regions, across each of the 3 feature maps to obtain 3 $\times$ (m-r+1)/2 $\times$ (n-r+1)/2 hidden features neurons. Normally the feature map is generated by using convolution operation in which the elements of the filter or the kernel are multiplied by the elements of the input matrix element-wise, and then the result is summed up to obtain one pixel of the feature map. The rest of the features are generated by sliding the filter across the input matrix. The convolution operation can be written mathematically as depicted in equation 1 below.

$$O(i,j) = \sum_{k=1}^{f}(\sum_{l=1}^{f} input(i+k-1, j+l-1)kernel(k,l)) \tag{1}$$

where i runs from 1 to $m-f+1$ and j runs from 1 to $n-f+1$.

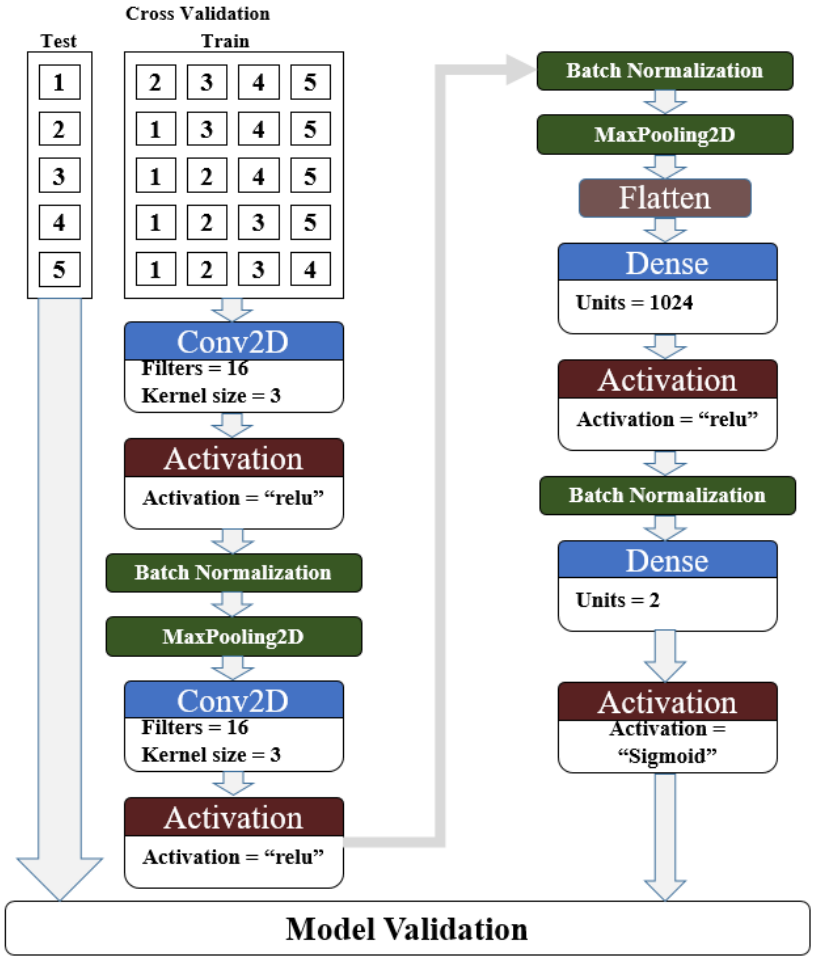

FIGURE 5. The pipeline representation of the constructed convolutional neural network model for breast cancer classification and the process of cross-validation.

E. THE CONSTRUCTED CNN MODEL

After converting the cases in the gene expression data into 2D-matrix i.e. image like data as shown in Figure 4, the visualization of the data shows that there is a recognizable pattern that can differentiate the two classes. The images that we obtained from the data have no complex shapes therefore; we constructed a lightweight CNN architecture using python programming language under the TensorFlow environment using Keras library. The pipeline representation of the constructed CNN architecture is depicted in Figure 5. This architecture consists of two convolutional layers where the first layer can look for low-level features such as edges, corners, and curves in the 2D-images and the second layer can build up high-level features based on the recognized low-level features that are obtained from the first layer. Two activation functions are used with our constructed CNN architecture namely the Rectified Linear Unit (ReLU), and sigmoid functions. ReLU is very popular and it has very good experimental results yet it has a very simple formula. It can be written mathematically as follows:

$$f(x) = max(0, x) \tag{2}$$

This means that it will return 0 for the negative values an the value back for positive values x. The sigmoid activation function can be written mathematically as follows:

$$f(x) = \frac{1}{1 + e^{-x}} \tag{3}$$

Thus the sigmoid function is nonlinear in nature and the value of f(x) will change significantly in response to change in x in the range of -2 to 2 so it has a tendency to bring f(x) value to either end of the curve and thus makes a clear distinction in the classification, therefore, we used it after the last dense layer. The order in which we use these two activation functions is shown in Figure 5.

The max pooling layer normally comes after the activation function that comes after the convolutional layer. the convolutional layer applies learned filters to the input features to create feature maps that take the exact location of features in the input, which means that slight movements in the location of the feature in the input features will end up with a different feature map. The max pooling layer can address this problem by down-sampling, which creates a lower resolution version of an input signal that still preserves the essential structural

element and thus summarizes the features detected in the convolutional layer. Our two max pooling2D layers that are depicted in Figure 5 are applied using a stride of (2,2) in 2×2 patches of the feature map i.e. they reduce the size of each feature map by a factor of 2.

To connect the convolutional layers to the dense layers we used a flatten layer, which converts the multidimensional output into linear. The dense layer is a fully connected layer that connects each input to each output.

In our CNN architecture, batch normalization is used after the activation functions by adjusting and scaling the activations to speed up the learning process and reduce the value of the hidden layer shift around (covariance shift). This allows each layer to learn independently of other layers and enables us to use less dropout and hence we will not lose a lot of information. During the batch normalization process two trainable parameters will be added, these parameters are the mean parameter β and the standard deviation parameter γ [45], where the inputs x are the values that are obtained from the activation function over mini-batch B and the output is $y_i = BN_{\beta\gamma}(x_i)$, which is a batch normalization for these values and it can be obtained as shown in the following equations.

$$\mu B \leftarrow \frac{1}{m}\sum_{i=1}^{m}(x_i) \tag{4}$$

$$\sigma^2 B \leftarrow \frac{1}{m}\sum_{i=1}^{m}(x_i - \mu B)^2 \tag{5}$$

$$\hat{x_i} \leftarrow \frac{x_i - \mu B}{\sqrt{\sigma^2 B + \epsilon}} \tag{6}$$

$$y_i \leftarrow \gamma \hat{x_i} + \beta = BN_{\beta\gamma}(x_i) \tag{7}$$

Our CNN has 3 hyper-parameters that are not known beforehand, these parameters are kernel size, number of filters, and dense units as shown in equation 9. Also, different learning rates are used during the training process in order to select the optimal accuracy within 100 epochs. The values of these hyper-parameters cannot be estimated from the data, and they should be obtained or set before building the CNN model. Therefore, we conducted a model selection or parameter search strategy to identify these parameters so that our CNN can predict unknown data, which is the testing data accurately. The values ranges that are used are depicted in equation 9. For the cancer classification, small and local features can differentiate the different class therefore; we used small sizes of kernels or filters in our grid search as shown in equation 9.

$$models = m : models(k, f, d) | k \in R^{kernel\_size}, \quad f \in R^{filter\_size}, \quad d \in R^{dens\_units} \tag{8}$$

$$best\_model = b : models | isMax(b.accurecy)$$
$$R^{kernel\_size} \in \{3, 5, 7\}$$
$$R^{filter\_size} \in \{8, 16, 32, 64\}$$
$$R^{dens\_units} \in \{128, 256, 512, 1024\} \tag{9}$$

Our parameters search strategy is based on a grid search using a stratified V fold cross-validation approach where the value of V is equal to 5. In the stratified V fold cross-validation data are split into V folds where V-1 folds are used as train sets and the remaining one is used as a testing set and it should fulfill the following criteria: First, the percentage of samples are preserved in each class when preparing the folds. Second, two different folds cannot contain the same group. Various values of our parameters are tried and the values that give the best cross-validation accuracy are selected.

F. PERFORMANCE MEASURES

We evaluated the performance of our constructed CNN using five measures. These measures are accuracy, sensitivity, specificity, precision, and F-measure. They are considered to be among the most frequent measures that are used to evaluate the machine learning performance and they are obtained using the four values true positive (TP), true negative (TN), false positive (FP), and false negative (FN), which refer to the number of the cases that are correctly classified as positive class, the number of the cases that are correctly classified as negative class, the number of cases that are in the negative class and incorrectly classified as positive class, and the number of cases that are in the positive class and incorrectly classified as negative class respectively.

The classification accuracy refers to the percentage of correctly classified cases and it is calculated as follows:

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{10}$$

Sensitivity (also known as recall or true positive rate) refers to the percentage of the samples that are correctly classified as positive (having cancer) among those observed as positive. In other words, it represents the proportion of the total positive samples that are correctly classified and it is calculated as follows:

$$Sensitivity = \frac{TP}{TP + FN} \tag{11}$$

Specificity (also known as true negative rate) measures the percentage of negative samples that are correctly identified as negative (not having the breast cancer). It can be calculated as follows:

$$Specificity = \frac{TN}{TN + FP} \tag{12}$$

Precision also is known as the positive predictive value. It is the percentage of the samples that are correctly classified as positive among the classified ones and is calculated as follows:

$$Precision = \frac{TP}{TP + FP} \tag{13}$$

F-measure also is known as F-score. It states the balance between precision and sensitivity and has a range of 0 to 1. 0 indicates that there are no true positives whereas 1 indicates

TABLE 1. Sample of the parameters' values and their results from the stratified 5 fold cross-validation.

Kernel Size	Number of Filters	Dens units	Accuracy	Accuracy Variance	F-Measure	F variance
3	16	1024	99.9	0.21	99.84	0.21
3	64	128	99.9	0.21	99.84	0.21
3	64	1024	99.9	0.21	99.84	0.21
5	8	128	99.9	0.21	99.84	0.21
5	8	256	99.9	0.21	99.84	0.21
5	16	256	99.9	0.21	99.84	0.21
5	32	256	99.9	0.21	99.84	0.21
5	32	128	99.79	0.41	99.77	0.28
3	64	512	99.69	0.62	99.7	0.39
3	8	128	99.59	0.39	99.67	0.32
5	16	128	97.21	2.57	98.24	1.67
9	8	256	96.94	3.20	98.23	1.87
3	32	128	96.58	6.33	97.84	3.79
9	32	128	96.11	4.15	97.73	2.49
7	64	128	91.57	14	94.28	9.79

that there are neither false negatives nor false positives. F-measure can be calculated as follows:

$$F - measure = \frac{2 * precision * sensitivity}{precision + sensitivity} \quad (14)$$

III. RESULTS AND DISCUSSION

As referred to earlier the constructed CNN model contains two convolutional layers. Before we start the training we tuned the hyper-parameters of the CNN using grid search. The best parameters' values i.e. the values the give high performance are selected to be used with our constructed model. Our goal is to construct a model that generalizes or classify with high performance previously unseen data. The hyper-parameters that give the best results and the results they yielded as evaluated by the accuracy and F-measure are shown in Table 1.

Table 1 shows that kernel size, number of filters, and dense units of 3, 16, and 1024 respectively yielded a very high accuracy and F-measure, these values are used to build the CNN model that is depicted in Figure 5. To evaluate the accuracy of our model we used leave one out cross-validation test. We first started by dividing our data set into five approximately equal sets and then we removed one set out of the five sets to represent the testing set and the remaining four sets are combined together to represent the training set. We repeated this process five times by removing one set in each time to represent the testing set. This way we will be having a different set for testing each time so that we can evaluate the generalization ability of our model. The average of the results from the five testing sets is taken to represent the final result. Our model Obtains an accuracy of 98.76%, a sensitivity of 91.43%, a specificity of 100%, a precision of 100%, and F-measure of 0.955.

The test and training accuracies and loss curves of the model are presented in Figure 6 and Figure 7 respectively.

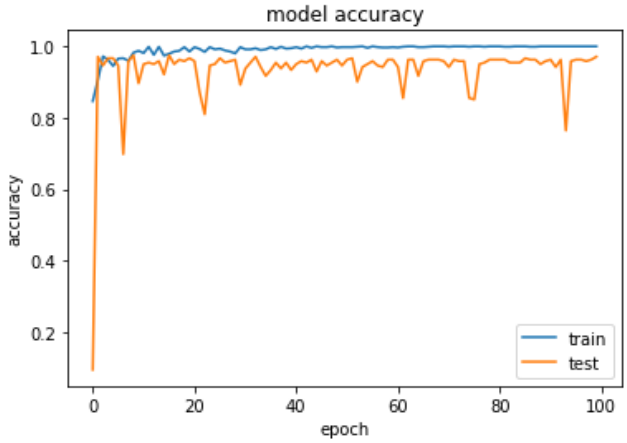

FIGURE 6. Training accuracy Curve.

These figures show that the more our model performs a training task the better its accuracy will be. Given that our training and testing are repeated 100 times (100 epochs). Both the test and training accuracies become stable at an accuracy that is more than 99%, which means that it can generalize very well when predicting unseen data. Figure 8 shows the F-measure training curve. The ROC curve of our model is depicted in Figure 9. The ROC curve is a threshold independent measure and it shows the effectiveness of our classification. The area under the ROC curve (AUC) is an important measure that reflects the classifier reliability. Our model shows AUC of 0.998 i.e. very close to 1, which indicates that it is a good classifier. Random classifier normally has an area around 0.5.

Table 2 shows the comparison between our lightweight convolutional neural network, Padideh Danaee and Reza Ghaeini method, and Yunchuan Kong and Tianwei Yu method. The last two methods are applied to the same RNA-

TABLE 2. The comparison of our method with other breast cancer classification methods.

Classification method	Accuracy	Sensitivity	Specificity	Precision	F-measure
Our Method	98.76%	91.43%	100%	100%	0.955
Padideh Danaee and Reza Ghaeini [8]	94.78%	93.04%	97.5%	97.20%	0.951
Yunchuan Kong and Tianwei Yu [28]	93.8%	N/A	N/A	N/A	N/A

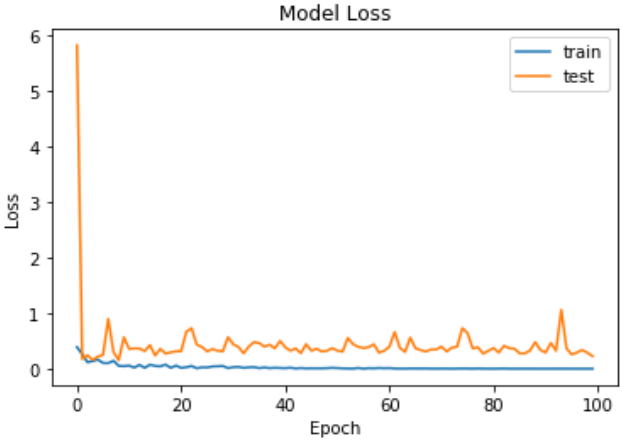

FIGURE 7. Training loss curve.

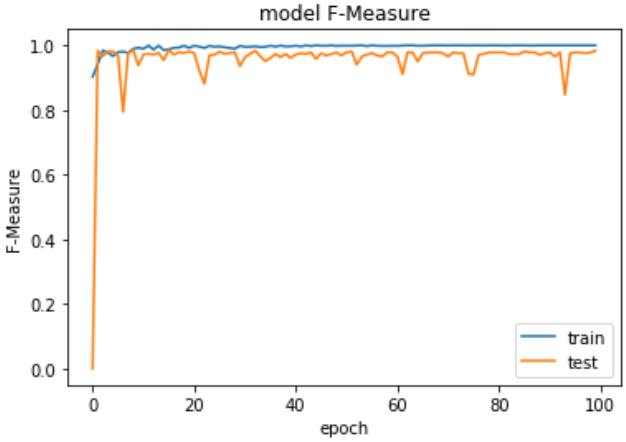

FIGURE 8. Training F-measure curve.

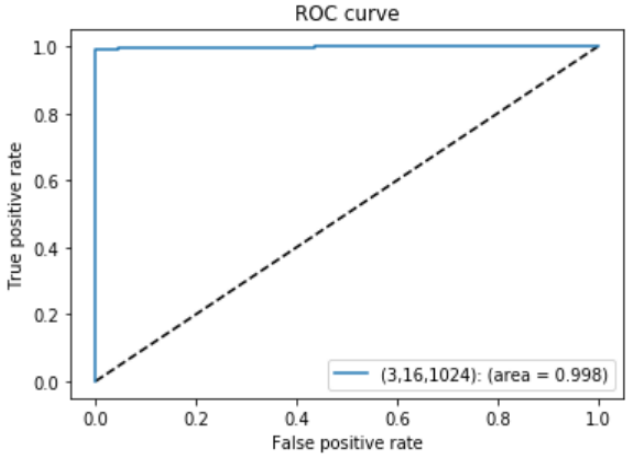

FIGURE 9. The ROC curve of our designed model for breast cancer classification.

Seq gene expression data of breast cancer from The Cancer Genome Atlas (TCGA) database that we used. As mentioned earlier in this section that our method achieves prediction accuracy = 98.76%, sensitivity = 91.43%, specificity = 100%, precision = 100%, and F-measure = 0.955. The accuracy of our method is the highest among the existing methods that use the same breast cancer data set that we used; i.e. Padideh Danaee and Reza Ghaeini's method and the method of Yunchuan Kong and Tianwei Yu, which achieved accuracy of 94.78%, and 93.8% respectively. The differences in the accuracy between our method and these methods are 3.98%, and 4.96 %. We highlight that these differences are reasonably large given that the baseline accuracy is around 91%, which could be obtained by merely regarding all samples as positive samples (the data set has 1095 positive samples vs 113 negative samples). i.e., our method provides 7.76/9 = 86.22% error rate reduction, while Padideh Danaee and Reza Ghaeini's method and the method of Yunchuan Kong and Tianwei Yu provide 3.78/9 = 42% and 2.8/9 = 31.11% error rate reduction respectively. Our method shows 100% precision and specificity compared to Padideh Danaee and Reza Ghaeini, which achieves 97.20% and 97.5% for precision and specificity respectively. 100% precision means that all the observed positive samples are correctly classified. And 100% specificity means that all negative samples are classified correctly.

We applied other convolutional neural network architectures, namely AlexNet and VGG16 architectures on our obtained breast cancer data set. AlexNet [20] consists of five convolutional layers and three fully-connected layers. Some of the convolutional layers are followed by max pooling layers. It uses ReLU activation after each convolutional and fully connected layer. It also utilizes dropout before the first and the second fully connected layers. VGG16 which consists of 16 convolutional layers was proposed by K. Simonyan and A. Zisserman [21]. Recently it was used by [46] for predictions of cancer types based on gene expression data. The results we obtained when we applied these CNN architectures are depicted in Table 3.

We also compared our results with SVM because it is considered among the most successful methods for cancer classification using gene expression data [47]. We used LIBSVM package [48] to train and build the SVM classification model using the breast cancer data set. Before building the model we

TABLE 3. The results of AlexNet, t-SNE + SVM, VGG16, SVM, and PCA + SVM on breast cancer classification.

CNN Architecture	Accuracy	Sensitivity	Specificity	Precision	F-measure
Our Method	98.76%	91.43%	100%	100%	0.955
AlexNet	96.69%	96.89%	94.12%	99.54	0.982
t-SNE + SVM	95.87%	100%	51.0%	95.96%	0.970
VGG16	95.04%	95.59%	86.67%	99.09%	0.973
SVM	92.23%	100%	0.00%	92.24%	0.959
PCA + SVM	90.59%	100%	0.00%	91.74%	0.957

performed a grid search to find the optimal parameters C and gamma of the LIBSVM tool and then we trained the SVM model. We used a five-fold cross-validation approach for training and testing our built model. the results we obtained are shown in table 3. Table 3 also shows the results of using t-SNE and PCA for reducing the features' dimensions before using the SVM classification method.

From the results, we can see that the SVM and t-SNE + support vector machine achieved a sensitivity of 100%. Because sensitivity concentrates on the actual positives that are correctly predicted and since our data set is highly imbalanced (9% negative class and 91% positive class), one can achieve very high sensitivity by predicting all the cases to be a positive class. In other words, classifiers are biased towards detecting the majority class and less sensitive to the minority class, and this leads to bias in classification [49], [50]. In such highly imbalanced data, the main interest will be to correctly classify the minority class. Therefore, specificity, which measures the proportion of the actual negatives that are correctly classified is an important measure i.e. a good classifier that has high credibility should have high sensitivity and specificity at the same time. As shown in the table that our proposed method obtained specificity of 100%, which is very high compared to SVM and t-SNE + SVM (they obtained Specificity of 0 and 51% respectively), which means that they don't classify negative samples well. At the same time, our method obtained a very high value of sensitivity which is 91.43% besides that our method provides the highest accuracy rate compared to the other methods that are shown in table 3. t-SNE + support vector machine provides $4.87/9 = 54.11\%$ error rate reduction compared to 86.22%, which is provided by our method. Regarding the deep learning methods, we see that AlexNet and VGG16 provide $5.69/9 = 63.22$, and $4.04/9 = 44.89$ error rate reduction respectively compared to 86.22% that is provided by our method. Also, our method is better than these two methods in classifying the minority class.

IV. CONCLUSION

In this work, we designed a new CNN architecture for breast cancer classification based on RNA-Seq gene expression data. The gene expression is downloaded from Pan-Cancer Atlas using R studio. The platform used for downloading the data is "Illumina HiSeq". The downloaded data is preprocessed by removing the outlier samples, which are determined based on the AAIC, which defines a symmetric square matrix of Spearman correlation between samples. A normalization process is applied on the data to ensure that we can infer the expression level from it correctly and avoid biases in the expression measures. Finally, filtering is applied on the obtained data. To overcome the problems of small size and high dimensionality of the gene expression data, we designed a special CNN architecture that contains only two convolutional layers. To improve the CNN performance, we selected its hyper-parameters carefully using a grid search approach with five-fold cross-validation, the values of the hyper-parameters that give the highest performance are selected to be used in our designed architecture. The accuracy of our CNN shows improvement compared with previous work on cancer classification. As of future work, we intend to use CNN for cancer multi-class classification.

ACKNOWLEDGMENT

The authors would like to acknowledge Jouf University for all the support that it provides.

REFERENCES

[1] S. Liu, C. Xu, Y. Zhang, J. Liu, B. Yu, X. Liu, and M. Dehmer, "Feature selection of gene expression data for Cancer classification using double RBF-kernels," *BMC Bioinform.*, vol. 19, no. 1, p. 396, 2018.
[2] Z. Wang, M. Gerstein, and M. Snyder, "RNA-Seq: A revolutionary tool for transcriptomics," *Nature Rev. Genet.*, vol. 10, no. 1, p. 57, 2009.
[3] F. Finotello and B. D. Camillo, "Measuring differential gene expression with RNA-seq: Challenges and strategies for data analysis," *Briefings Funct. Genomics*, vol. 14, no. 2, pp. 130–142, 2015.
[4] K. R. Kukurba and S. B. Montgomery, "RNA sequencing and analysis," *Cold Spring Harbor Protocols*, vol. 2015, no. 11, pp. 951–969, 2015.
[5] M. Maienschein-Cline, J. Zhou, K. P. White, R. Sciammas, and A. R. Dinner, "Discovering transcription factor regulatory targets using gene expression and binding data," *Bioinformatics*, vol. 28, no. 2, pp. 206–213, 2012.
[6] E. E. Schadt, J. Lamb, X. Yang, J. Zhu, S. Edwards, D. GuhaThakurta, S. K. Sieberts, S. Monks, M. Reitman, and C. Zhang, "An integrative genomics approach to infer causal associations between gene expression and disease," *Nature Genet.*, vol. 37, no. 7, p. 710, 2005.
[7] K. Shabana, K. A. Nazeer, M. Pradhan, and M. Palakal, "A computational method for drug repositioning using publicly available gene expression data," *BMC Bioinf.*, vol. 16, no. 17, p. S5, 2015.
[8] P. Danaee, R. Ghaeini, and D. A. Hendrix, "A deep learning approach for cancer detection and relevant gene identification," in *Proc. Pacific Symp. Biocomput.*, 2017, pp. 219–229.
[9] J. Krammer, K. Pinker-Domenig, M. E. Robson, M. Gonen, B. Bernard-Davila, E. A. Morris, D. A. Mangino, and M. S. Jochelson, "Breast cancer detection and tumor characteristics in BRCA1 and BRCA2 mutation carriers," *Breast Cancer Res. Treat.*, vol. 163, no. 3, pp. 565–571, 2017.
[10] A. A. Alizadeh, M. B. Eisen, R. E. Davis, C. Ma, I. S. Lossos, A. Rosenwald, J. C. Boldrick, H. Sabet, T. Tran, and X. Yu, "Distinct types of diffuse large B-cell lymphoma identified by gene expression profiling," *Nature*, vol. 403, no. 6769, p. 503, 2000.
[11] U. Alon, N. Barkai, D. A. Notterman, K. Gish, S. Ybarra, D. Mack, and A. J. Levine, "Broad patterns of gene expression revealed by clustering analysis of tumor and normal colon tissues probed by oligonucleotide arrays," *Proc. Nat. Acad. Sci. USA*, vol. 96, no. 12, pp. 6745–6750, 1999.

[12] J. Li, H. Liu, S.-K. Ng, and L. Wong, "Discovery of significant rules for classifying cancer diagnosis data," *Bioinformatics*, vol. 19, no. 2, pp. ii93–ii102, 2003.
[13] S. A. Medjahed, T. A. Saadi, and A. Benyettou, "Breast cancer diagnosis by using *k*-nearest neighbor with different distances and classification rules," *Int. J. Comput. Appl.*, vol. 62, no. 1, pp. 1–5, 2013.
[14] K. J. Cios, R. W. Swiniarski, W. Pedrycz, and L. A. Kurgan, "Supervised learning: Statistical methods," in *Data Mining*. Boston, MA, USA: Springer, 2007, pp. 307–379.
[15] S. Guha, Y. Ji, and V. Baladandayuthapani, "Bayesian disease classification using copy number data," *Cancer Inf.*, vol. 13, no. 2, pp. 83–91, 2014.
[16] J. S. Gregory and D. William, "Efficient cancer detection using multiple neural networks," *IEEE J. Transl. Eng. Health Med.*, vol. 5, 2017, Art. no. 2800607.
[17] A. K. Dwivedi, "Artificial neural network model for effective cancer classification using microarray gene expression data," *Neural Comput. Appl.*, vol. 29, no. 12, pp. 1545–1554, Jun. 2016.
[18] T. S. Furey, N. Cristianini, N. Duffy, D. W. Bednarski, M. Schummer, and D. Haussler, "Support vector machine classification and validation of cancer tissue samples using microarray expression data," *Bioinformatics*, vol. 16, no. 10, pp. 906–914, 2000.
[19] C. D. A. Vanitha, D. Devaraj, and M. Venkatesulu, "Gene expression data classification using support vector machine and mutual information-based gene selection," *Procedia Comput. Sci.*, vol. 47, pp. 13–21, 2015.
[20] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," in *Proc. Adv. Neural Inf. Process. Syst.*, 2012, pp. 1097–1105.
[21] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," 2015, *arXiv:1409.1556*. [Online]. Available: https://arxiv.org/abs/1409.1556
[22] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, "Going deeper with convolutions," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2014, pp. 1–9.
[23] G. Hinton, D. Li, Y. Dong, G. E. Dahl, A. Mohamed, N. Jaitly, A. Senior, V. Vanhoucke, P. Nguyen, T. N. Sainath, and B. Kingsbury, "Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups," *IEEE Signal Process. Mag.*, vol. 29, no. 6, pp. 82–97, Nov. 2012.
[24] Y. Bengio, A. Courville, and P. Vincent, "Representation learning: A review and new perspectives," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 35, no. 8, pp. 1798–1828, Aug. 2013.
[25] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, no. 7553, p. 436, 2015.
[26] S. Min, B. Lee, and S. Yoon, "Deep learning in bioinformatics," *Brief. Bioinform.*, vol. 18, no. 5, pp. 851–869, 2017.
[27] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, and M. Bernstein, "ImageNet large scale visual recognition challenge," *Int. J. Comput. Vis.*, vol. 115, no. 3, pp. 211–252, Dec. 2015.
[28] Y. Kong and T. Yu, "A graph-embedded deep feedforward network for disease outcome classification and feature selection using gene expression data," *Bioinformatics*, vol. 34, no. 21, p. 3727, 2018.
[29] B. Lyu and A. Haque, "Deep learning based tumor type classification using gene expression data," in *Proc. ACM Int. Conf. Bioinf., Comput. Biol., Health Inform. (BCB)*, 2018, pp. 89–96.
[30] R. K. Sevakula, V. Singh, K. N. Verma, C. Kumar, and Y. Cui, "Transfer learning for molecular cancer classification using deep neural networks," *IEEE/ACM Trans. Comput. Biol. Bioinf.*, vol. 16, no. 6, pp. 2089–2100, Nov./Dec. 2019.
[31] J. N. Weinstein, E. A. Collisson, G. B. Mills, K. R. M. Shaw, B. A. Ozenberger, K. Ellrott, I. Shmulevich, C. Sander, and J. M. Stuart, "The cancer genome atlas pan-cancer analysis project," *Nature Genet.*, vol. 45, no. 10, p. 1113, 2013.
[32] A. Colaprico, T. C. Silva, C. Olsen, L. Garofano, C. Cava, D. Garolini, T. S. Sabedot, T. M. Malta, S. M. Pagnotta, and I. Castiglioni, "TCGAbiolinks: An R/Bioconductor package for integrative analysis of TCGA data," *Nucleic Acids Res.*, vol. 44, no. 8, p. e71, 2015.
[33] I. Van der Auwera, S. J. Van Laere, G. G. Van den Eynden, I. Benoy, P. van Dam, C. G. Colpaert, S. B. Fox, H. Turley, A. L. Harris, E. A. Van Marck, P. B. Vermeulen and L. Y. Dirix, "Increased angiogenesis and lymphangiogenesis in inflammatory versus noninflammatory breast cancer by real-time reverse transcriptase-PCR gene expression quantification," *Clin. Cancer Res.*, vol. 10, no. 23, pp. 7965–7971, 2004.
[34] G. M. Silva and C. Vogel, "Quantifying gene expression: The importance of being subtle," *Mol. Syst. Biol.*, vol. 12, no. 10, p. 885, 2016.
[35] J. Choi, T. M. Baldwin, M. Wong, J. E. Bolden, K. A. Fairfax, E. C. Lucas, R. Cole, C. Biben, C. Morgan, and K. A. Ramsay, "Haemopedia RNA-seq: A database of gene expression during haematopoiesis in mice and humans," *Nucleic Acids Res.*, vol. 47, no. D1, pp. D780–D785, 2018.
[36] A. McDermaid, B. Monier, J. Zhao, B. Liu, and Q. Ma, "Interpretation of differential gene expression results of RNA-seq data: Review and integration," *Briefings Bioinf.*, pp. 1–11, Aug. 2018.
[37] M. Morgan, V. Obenchain, J. Hester, and H. Pagès, "Summarized-experiment: Summarized-experiment container," 2018. [Online]. Available: https://bioconductor.org/packages/SummarizedExperiment. Rpackageversion1.12.0
[38] W. Huber, V. J. Carey, R. Gentleman, S. Anders, M. Carlson, B. S. Carvalho, H. C. Bravo, S. Davis, L. Gatto, and T. Girke, "Orchestrating high-throughput genomic analysis with bioconductor," *Nature Methods*, vol. 12, no. 2, p. 115, 2015.
[39] S. Yang, X. Guo, Y.-C. Yang, D. Papcunik, C. Heckman, J. Hooke, C. D. Shriver, M. N. Liebman, and H. Hu, "Detecting outlier microarray arrays by correlation and percentage of outliers spots," *Cancer Informat.*, vol. 2, pp. 351–360, 2006.
[40] D. Risso, K. Schwartz, G. Sherlock, and S. Dudoit, "GC-content normalization for RNA-seq data," *BMC Bioinf.*, vol. 12, no. 1, p. 480, 2011.
[41] J. H. Bullard, E. Purdom, K. D. Hansen, and S. Dudoit, "Evaluation of statistical methods for normalization and differential expression in MRNA-seq experiments," *BMC Bioinf.*, vol. 11, no. 1, p. 94, 2010.
[42] K. D. Hansen, R. A. Irizarry, and Z. Wu, "Removing technical variability in RNA-seq data using conditional quantile normalization," *Biostatistics*, vol. 13, no. 2, pp. 204–216, 2012.
[43] W. Zheng, L. M. Chung, and H. Zhao, "Bias detection and correction in RNA-sequencing data," *BMC Bioinf.*, vol. 12, no. 1, p. 290, 2011.
[44] C.-Y. Lee, P. W. Gallagher, and Z. Tu, "Generalizing pooling functions in convolutional neural networks: Mixed, gated, and tree," in *Proc. 19th Int. Conf. Artif. Intell. Statist. (AISTATS)*. Cadiz, Spain: JMLR: W&CP, 2016, pp. 464–472.
[45] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," 2015, *arXiv:1502.03167*. [Online]. Available: https://arxiv.org/abs/1502.03167
[46] M. Karim, M. Cochez, O. Beyan, S. Decker, and C. Lange, "OncoNet-Explainer: Explainable predictions of cancer types based on gene expression data," 2019, *arXiv:1909.04169*. [Online]. Available: https://arxiv.org/abs/1909.04169
[47] M. Mohammed, H. Mwambi, B. Omolo, and M. K. Elbashir, "Using stacking ensemble for microarray-based cancer classification," in *Proc. Int. Conf. Comput., Control, Elect., Electron. Eng. (ICCCEEE)*, 2018, pp. 1–8.
[48] C.-C. Chang and C.-J. Lin, "LIBSVM: A library for support vector machines," *ACM Trans. Intell. Syst. Technol.*, vol. 2, no. 3, p. 27, 2011.
[49] P. Fergus, D.-S. Huang, and H. Hamdan, "Prediction of intrapartum hypoxia from cardiotocography data using machine learning," in *Applied Computing in Medicine and Health*. Amsterdam, The Netherlands: Elsevier, 2016, pp. 125–146.
[50] S. Iram, F.-B. Vialatte, and M. I. Qamar, "Early diagnosis of neurodegenerative diseases from gait discrimination to neural synchronization," in *Applied Computing in Medicine and Health*. Amsterdam, The Netherlands: Elsevier, 2016, pp. 1–26.

MURTADA K. ELBASHIR received the B.Sc. degree (Hons.) in computer/statistics from the University of Gezira, Wad Madani, Sudan, in 2000, the M.Sc. degree (Hons.) in computer information systems from the University of the Free State, Bloemfontein, South Africa, in 2003, and the Ph.D. degree in computer science and technology from Central South University, Changsha, China, in 2013. In 2016, he promoted to an Associate Professor with the University of Gezira. He is currently an Associate Professor with the College of Computer and Information Sciences, Jouf University, Sakaka, Saudi Arabia. His current research interests include machine learning and bioinformatics.

MOHAMED EZZ received the B.Sc., M.Sc., and Ph.D. degrees in systems and computers engineering from the Faculty of Engineering, Al-Azhar University. He is currently an Associate Professor with the Faculty of Engineering, Al-Azhar University and also a Visiting Professor with the College of Computer and Information Sciences, Jouf University. He has published 20 scientific articles in various national and international journals and conferences. He has contributed in more than 16 mega software projects in Electronic banking EBPP, EMV, mobile banking, and e-commerce. His areas of interests include pattern recognition, applied machine learning, security applications, intrusion detection, and semantic web.

MOHANAD MOHAMMED received the B.Sc. degree (Hons.) from the University of Gezira, Wad Madani, Sudan, and the M.Sc. degree from the University of KwaZulu-Natal, South Africa. He is currently pursuing the Ph.D. degree with the School of Mathematics, Statistics, and Computer Science, University of KwaZulu-Natal. He is currently a Teacher Assistant with the Faculty of Mathematical and Computer Sciences University of Gezira. From his M.Sc. degree, he published an article. His researches focus on cancer disease diagnosis and prognosis using gene expression data, and next-generation sequencing data via statistical and machine learning techniques. His current researches focus on combining the gene expression and next-generation sequence data (Omics data) for cancer prediction using Poisson regression model, negative binomial linear discriminant analysis, and convolutional neural networks among other machine learning methods. He received three senate prizes for academic excellence from the Deanship of Academic Affairs, University of Gezira. He was awarded a prize of academic excellence owing to his outstanding performance during his M.Sc. degree.

SAID S. SALOUM was born in Irbed, Jordan. He received the master's degree in radio-physics and electronics from K0SU, Russia, in 1995, and the Ph.D. degree in computer engineering from IzSTU, Russia, in 2004. He is currently working as an Assistant Professor with the Computer Engineering and Networks Department, Jouf University, Saudi Arabia. His research interests include image processing, machine learning, and deep learning.