https://doi.org/10.3311/PPee.23081

Periodica Polytechnica Electrical Engineering and Computer Science, 68(2), pp. 202–212, 2024

Ovarian Cancer Detection Based on Elman Recurrent Neural Network

Hameed R. Farhan[1]*, Muayad S. Kod[1], Arwa M. Taqi[1], Suleiman A. Ghazi[1]

[1] Department of Electrical and Electronic Engineering, College of Engineering, University of Kerbala, P.O.B. 367, 56001 Kerbala, Iraq
* Corresponding author, e-mail: hameed.r.f@uokerbala.edu.iq

Received: 28 July 2023, Accepted: 07 November 2023, Published online: 28 February 2024

Abstract

The early detection of cancers increases the possibility of health recovery and prevents the disease from becoming a silent killer. This study introduces an effective method for identifying ovarian cancer (OC) using Elman Recurrent Neural Network (ERNN), which can recognize cancer via mass spectrometry data. The network has a topology of 100 input neurons for receiving data, five neurons for hidden and context layers, and two output nodes to indicate the status. The proposed method uses reduced-size features, including ion concentration levels at specific mass/charge values, which are trained using various learning algorithms to determine the suitable one that achieves the best results. The experimental results show that all the training algorithms achieve about 100% performance rate, with the Levenberg Marquardt (LM) being the most accurate and fastest algorithm, which converges after six epochs and achieves 0.0035, 0.0045 and 0.0045 mean square errors for training, validation, and test performances, respectively. Based on comparative results, the proposed LM-ERNN method outperforms other OC detection methods and holds promise for detecting other types of cancer.

Keywords

ovarian cancer, Elman Recurrent Neural Network, deep learning, Levenberg Marquardt algorithm

1 Introduction

Nowadays, cancer has become a significant risk that affects our life. One type of cancer that infects the women's generation system is ovarian cancer (OC). According to the latest study on women, OC contributes to more than 4% of cancer mortality. Countries with a high human development index have the highest cancer incidence; however, the mortality rate curve reverses [1]. The occurrence of OC is influenced by several factors, with genetic factors being the most significant. Other factors, such as lactation, oral contraceptive pills and pregnancy, reduce the possibility of OC infection [2].

The recent electronics revolution has led to the creation of different computer systems for various applications. These systems are superior to humans in speed, accuracy, and efficiency. Computers are widely used in medical applications such as diagnosing diseases, which can be accomplished in two different methods according to the type of application. The first method utilizes the same pattern recognition procedure as [3], i.e., a dedicated algorithm trains certain images and different images are examined for decision-making. An example of this type of disease diagnosis is breast cancer detection, which depends on mammographic images [4]. The second method depends on the data file, which includes information about specific tests such as genes, hormones, proteins, and blood components. An example of this can be seen in the classification of genomics [5]. It is important to mention that there are two methods for diagnosing OC. The first method uses ultrasound Doppler images of the ovaries [6–8], whereas the second method involves analyzing the serum proteomic data profile [9]. In this particular case, the proposed work will use the second method due to its simplicity, accuracy, and speed.

Patient samples are differentiated using serum proteomic pattern diagnostics to decide whether the patient is infected. Surface-Enhanced Laser Desorption and Ionization (SELDI) mass spectrometry can produce protein profile data patterns. This technology can potentially improve clinical diagnostics tests for cancer pathologies [9]. Data profiles contain immense data points that should be analyzed using elegant analytical methods. Bioinformatics is widely used to analyze the outcomes of the physiological

Cite this article as: Farhan, H. R., Kod, M. S., Taqi, A. M., Ghazi, S. A. "Ovarian Cancer Detection Based on Elman Recurrent Neural Network", Periodica Polytechnica Electrical Engineering and Computer Science, 68(2), pp. 202–212, 2024. https://doi.org/10.3311/PPee.23081

and microarrays of cluster genes. This pattern helps distinguish veiled samples from healthy women [9].

In this study, we aim to address the challenge of detecting OC using data protein profiles by applying Machine Learning (ML) techniques. Specifically, we focus on evaluating the performance of several training algorithms within the realm of deep learning to identify the most effective approach for OC detection. OC is a significant health concern, and early detection is critical in improving patient outcomes. Traditional methods of OC detection often need to be revised in accuracy and efficiency. Therefore, there is a pressing need to explore advanced computational techniques, such as deep learning, to enhance the precision and reliability of OC detection. A comprehensive examination of different training algorithms frequently utilized in deep learning is conducted. Performance metrics such as accuracy rate and number of iterations are used to evaluate algorithms carefully. The proposed work aims to choose the best algorithm that achieves the highest detection rate with the lowest computational complexity, which is the demand for real-time applications. Our proposed system presents a precise and reliable approach to OC detection using the advanced capabilities of deep learning algorithms and utilizing data protein profiles. The reduced complexity ensures that the system can be implemented efficiently and effectively within clinical settings, paving the way for improved early detection and timely intervention.

2 Related work

Researchers have made efforts for OC detection and presented valuable insights through rigorous experimentation and analysis. Ongoing research in this field aims to develop more effective methods for obtaining optimal results. Below are some recent approaches that address the prediction of OC.

An OC diagnosis method based on Fuzzy Neural Network (FNN) was introduced in [10]; the data used in this method were taken from the DNA microarray used to identify gene expressions. To reduce huge features, the researchers used the sparse logistic regression method to choose only nine features that can be used to classify the four classes of OC diagnosis. Although the method achieved an accuracy rate of about 84.72%, it is not within the ambition for OC detection. In [11], an Error Guided Artificial Bee Colony (EABC) model was introduced to detect OC. The model used a neural network with 100 inputs, two hidden layers containing 20 neurons each, and a single output. The model was iterated 200 times during training to attain the optimal outcome. However, an accuracy rate of approximately 91.2% is still needed to reach the desired detection rate. The authors in [12] utilized Deep Convolutional Neural Networks (DCNN), and a pre-trained CNN model called AlexNet to classify OC from cytological images. They improved the accuracy from 72.76% to 78.20% using the training data of augmented cytological images. A model of Artificial Neural Network (ANN) with 15 neurons and a sigmoid activation function was used in [13] to detect OC, where it achieved a 98.7% of classification rate. Optimal Recurrent Neural Networks (ORNN) and Self Organizing Maps (SOM) were used for OC detection in [14]. In addition, an Adaptive Harmony Search Optimization (AHSO) algorithm was used to optimize the weights of RNN, where the detection rate of this method was about 96.27%. Principal Component Analysis (PCA) was used in [15] to construct 2D barcodes of converted serum glycopeptides expression features. AlexNet was applied to detect epithelial OC early-stage, which introduced 95% accuracy of cancer detection. In [16], the authors examined four optimization methods on three types of feature selection for choosing the best genes that can be used to identify OC. In addition, they examined five algorithms to classify OC and data mining that can be used to predict OC recurrence. They concluded that using the multi-layer perceptron (MLP) with GDX achieved 98.96% classification accuracy. Besides, using the Kruskal Wallis test for feature extraction, Genetic Bee Colony (GBC) for optimization, Radial Basis Function (RBF) with and Support Vector Machine (SVM) for classification achieved an accuracy rate of about 99.48%. An ML-based system was introduced in [17] to predict OC using the most concerning features and a decision tree model, where two biomarkers have been suggested, human epididymis protein 4 (HE4) and carcinoembryonic antigen (CEA). The experimental results showed that CEA is a suitable biomarker for predicting OC in patients with low HE4, and ML is better than the risk of ovarian malignancy algorithm, where the method achieved a 94.9% accuracy rate. In [18], different ML methods were utilized to classify the testing women where they have OC, Ovarian LMP, Fallopian Tube Cancer, Peritoneal Cancer, and no-cancer. The most successful outcome was achieved using the random forest (RF) algorithm, which attained an accuracy rate of approximately 72%. Also, the RF algorithm was proven the best classifier in [19], with the obtained accuracy of 90.5% using the median imputation. In [20], the SVM model was employed in conjunction with K-Nearest Neighbor (KNN) model to predict OC, where the use of SVM achieved a better accuracy rate than the KNN, which was about 97.16%. The authors in [21] utilized an algorithm called classification and regression trees (CART) for diagnosing OC, where the binary decision tree achieved 98.1% accuracy using the CPTAC dataset. In addition, the researchers tested the data from another database on the first database's trained model, obtaining an accuracy rate of 98.2%.

3 Background

3.1 Elman Recurrent Neural Network (ERNN)

ANN is a popular classification method that comprises several types: recurrent, backpropagation, feedforward, etc. The RNN structure includes single or multiple hidden layers of feedback loops. These loops connect the neurons of two layers using a unit delay element (z^{-1}), representing the main difference compared with feedforward connections [22]. The role of the delay unit is to work as a memory location; it keeps the current activity value and displays it to the next event, as shown in Fig. 1. This process involves propagating through a recurrent backward loop and manipulating the activation with the input of the next pattern before presenting the result at the output layer. Jordan and Elman are maybe the most known for using the RNN as a full and partial mechanism [23].

In 1990, Jeffrey L. Elman proposed a simple structure of RNN called Elman RNN (ERNN) [24]. It is a back-propagation ANN comprising input, output, and hidden layers. The ERNN model contains a unique layer known as the context layer. This layer establishes connections between the outputs and inputs of the hidden layer, allowing for feedback. The input and context nodes activate the hidden nodes, which then activate the output and context nodes through feedforward and feedback connections. The process is illustrated in Fig. 2 [25].

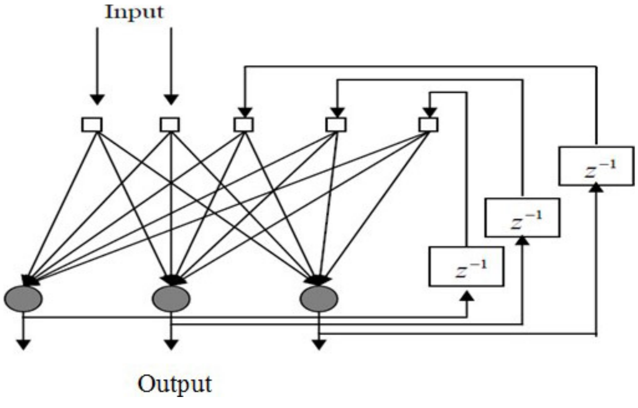


Fig. 1 RNN structure [22]

When the signal moves towards the output nodes, it goes through the hidden neurons. These neurons have transfer functions that can be either linear or nonlinear. On the other hand, the context nodes are employed to save the output values of the hidden nodes, where each hidden node transmits its output to the corresponding context node, which memorizes the received value and then returns it to the same hidden node. As a result, the context nodes are activated by the previous inputs of the hidden nodes and keep their information for the next iteration [26].

The ERNN structure can detect and produce transitory patterns due to the saved values in the context nodes from the earlier steps. Therefore, the number of iterations and feedback of the two Elman networks, which may have the same inputs, biases, and weights at a given time, can produce different outputs. Besides, ERNN can train patterns that may be altered with time because of its ability to store data used for future estimation. ERNN with nonlinear state-space is expressed as follows [26]:

$$Y(t) = g\left(w^3\left(X(t)\right)\right), \tag{1}$$

$$X(t) = f\left(w^1 X_c(t) + w^2\left(u(t-1)\right)\right), \tag{2}$$

$$X_c(t) = X(t-1), \tag{3}$$

where:

- $u(t-1)$, $Y(t)$, $X(t)$ and $X_c(t)$ are the network input, network output, output of the hidden layer, and the feedback vector, respectively;

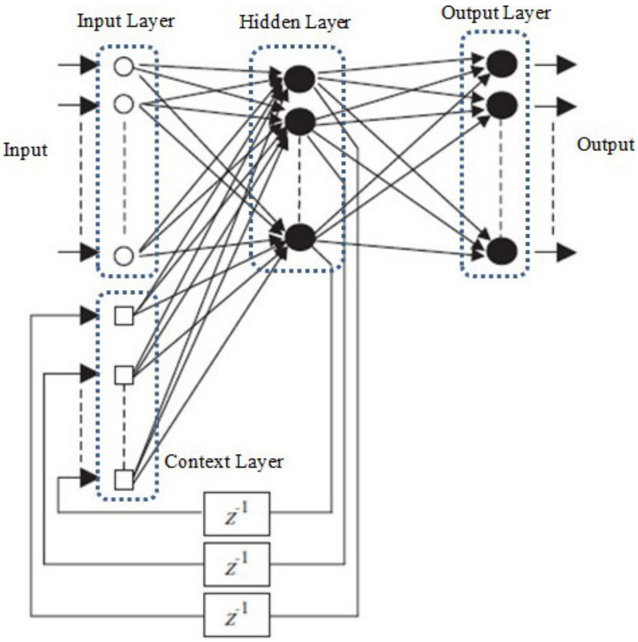


Fig. 2 ERNN model [25]

- w^1, w^2 and w^3 are the connections' weights between the hidden layer and the context, input and output layers, respectively;
- g and f are the activation functions of the output layer and hidden layer, respectively.

ERNN uses the linear (pure-lin) activation function in the output layer and the hyperbolic tangent sigmoid (tan-sigmoid) function in the hidden layer as denoted in Eqs. (4) and (5), respectively:

$$g(x) = x, \tag{4}$$

$$f(x) = \frac{1-e^{-2x}}{1+e^{-2x}}, \tag{5}$$

where x is the weighted input for output or hidden nodes.

From Eqs. (2) and (3), one can deduce the expression that refers to the time-varying status of ERNN as follows:

$$X_c(t) = f\left(w^1(t-1)X_c(t-1) + w^2(t-1)u(t-2)\right). \tag{6}$$

In addition, Eq. (6) illustrates the dependence of feedback values on the weights estimated from different periods.

Although ERNN has advantages, it requires more neurons in the hidden layer than other ANNs. Besides, the training process is accomplished using the error gradient approximation. The squared error function $E(w)$ is used to update the weights, as illustrated in Eq. (7) [26].

$$E(w) = \sum_{k=1}^{n}\left[\boldsymbol{y_k}(\boldsymbol{w}) - \boldsymbol{\tilde{y}_k}(\boldsymbol{w})\right]^2, \tag{7}$$

where $\boldsymbol{y_k}(\boldsymbol{w})$ and $\boldsymbol{\tilde{y}_k}(\boldsymbol{w})$ are the actual and target output vectors.

3.2 Training algorithms

The ANN training algorithms aim to obtain a decision function that can update the network weights, where several algorithms have been used and achieved different results. Examining all the training algorithms is a time-consuming and laborious task. It requires a lot of effort and patience, but according to the previous work, the best training algorithms are:

- Gradient Descent Adaptive Learning backpropagation (GDA);
- Gradient Descent momentum (GDX);
- Levenberg Marquardt (LM);
- Adaptive learning backpropagation with Resilient Backpropagation (R_{PROP}).

Therefore, this study has used these algorithms to determine the best one that meets the performance rate and speed requirements.

3.2.1 Gradient Descent Adaptive Learning backpropagation (GDA)

Backpropagation networks automatically receive raw data and can achieve a specific function during the training by adjusting the weights of connections between nodes. The network undergoes multiple training iterations until the difference between the target and actual outputs approaches zero. The trained network with a supervised model accepts many pairs of input-target vectors. The word "adaptive" refers to increasing the network training by subsequently changing the biases and weights of the network according to the matching between the output-target vectors [27].

Gradient descent (GD) has been employed in the backpropagation model, where the partial derivatives and chain rule are used to calculate the cost function or the difference error (Δ) [28]:

$$\Delta = \frac{1}{2}(Y_d - Y_l)^2, \tag{8}$$

where Y_d and Y_l are the desired and actual outputs, respectively. In the output layer, the partial error derivative is:

$$\frac{\partial E}{\partial Y_l} = Y_l - Y_d, \tag{9}$$

After each learning iteration, the weights are modified to reduce the cost function, where this process is iterated until no further reduction can be achieved.

3.2.2 Gradient-Descent momentum (GDX)

The learning rate is used in the standard stochastic GD to update the weights, but it can cause a convergence delay in the case of a noisy gradient or a potential minimum overshoot if the gradient is much steeper. In physics, the momentum introduces an exponentially decreasing velocity related to the average gradient. The same concept can be applied to the ERNN to avoid high descent in the unwanted direction. The velocity update is given by [28]:

$$v \leftarrow \alpha v - \epsilon d. \tag{10}$$

In addition, the actual update is shown as in Eq. (11):

$$\theta \leftarrow \theta + v, \tag{11}$$

where v is the velocity, α is the momentum parameter; $\alpha \in (0,1)$, d and θ are the gradient parameters, and ϵ is the learning rate.

3.2.3 Levenberg Marquardt (LM)

The Levenberg-Marquardt (LM) algorithm uses the Gauss-Newton and GD methods to solve non-linear least squares problems. In a curve-fitting problem, points of the function with the least-squares errors between the actual and existing data points are fitted. The Gauss-Newton method aims to minimize the error by finding the minimum quadratic function assumed to be locally quadratic [28]. It has fast local convergence for problems that are somewhat non-linear and almost consistent. However, it should be noted that it might not be able to converge locally on highly non-linear problems or have large residuals [29]. Besides, GD reduces the sum of the squared errors by modifying the function parameters towards the abrupt descent [28].

Let $\hat{Y}(t; q)$ be the fitting function of an independent variable (t) and a vector ($\boldsymbol{q}$) of n parameters to a set of m data points (t_i, Y_i). If $\boldsymbol{W}$ is a diagonal weighted matrix, the squared error $E^2(q)$ is given by [28]:

$$E^2(\boldsymbol{q}) = Y^T\boldsymbol{W}Y - 2Y^T\boldsymbol{W}\hat{Y} + \hat{Y}^T\boldsymbol{W}\hat{Y}. \tag{12}$$

The GD of E^2 with respect to the q parameters is denoted by:

$$\frac{\partial}{\partial q}E^2 = -2\left(Y-\hat{Y}\right)^T \boldsymbol{WJ}, \tag{13}$$

where $\boldsymbol{J}$ is the Jacobian ($m \times n$) matrix that represents $\partial\hat{Y}/\partial q$, where $\hat{Y} = Eq$ in the linear models. The GD update h_{gd} is denoted by:

$$h_{gd} = \alpha\boldsymbol{J}^T\boldsymbol{W}\left(Y-\hat{Y}\right), \tag{14}$$

where the step length is determined by the positive scalar α in the steepest-descent direction.

The Gauss-Newton update h_{gn} is given by:

$$\left[\boldsymbol{J}^T\boldsymbol{WJ}\right]h_{gn} = \boldsymbol{J}^T\boldsymbol{W}\left(Y-\hat{Y}\right). \tag{15}$$

The LM update h_{lm} adaptively varies the parameters between the GD and Gauss-Newton as in Eq. (16):

$$\left[\boldsymbol{J}^T\boldsymbol{WJ} + \lambda\, diag\left(\boldsymbol{J}^T\boldsymbol{WJ}\right)\right]h_{lm} = \boldsymbol{J}^T\boldsymbol{W}\left(Y-\hat{Y}\right), \tag{16}$$

where the damping parameter λ is scaled by the diagonal of the Hessian $\boldsymbol{J}^T\boldsymbol{WJ}$ for each parameter. If λ has a high value, the update method is considered GD. However, if the value of λ is low, the update method is considered Gauss-Newton [28].

3.2.4 R_{PROP}

R_{PROP} is a resilient propagation that directly updates the weights according to the local gradient calculations. Each weight is assigned a unique value that determines how much it will be adjusted during the training based on the error function [30].

Let $\boldsymbol{W}_{ij}$ be the weight assigned from j to i neurons and t represents the learning iteration. The new weight is updated as in Eq. (17) [31]:

$$\boldsymbol{W}_{ij}^{(t+1)} = \boldsymbol{W}_{ij}^{(t)} + \Delta\boldsymbol{W}_{ij}^{(t)}. \tag{17}$$

The weight update direction depends on the sign of $\partial E/\partial \boldsymbol{W}_{ij}$, where E is an arbitrary error measure. The R_{PROP} technique is unique because it uses a step-size Δ_{ij}, which is not dependent on the absolute value of $\partial E/\partial \boldsymbol{W}_{ij}$. If $\partial E/\partial \boldsymbol{W}_{ij} > 0$, the weight is decreased, i.e., $\boldsymbol{W}_{ij} = -\Delta_{ij}$, while if $\partial E/\partial \boldsymbol{W}_{ij} < 0$, the weight is increased, i.e., $\boldsymbol{W}_{ij} = +\Delta_{ij}$.

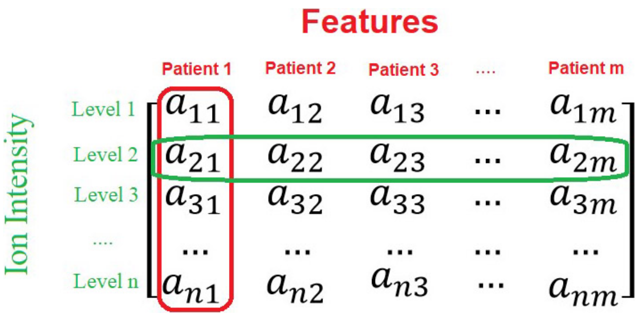
4 The Proposed ERNN Model

This study utilizes an ERNN as a classifier to distinguish between normal and cancer patients by selecting a reduced set of features that represent intensity levels of serum proteomics at specific charge/mass values. The serum proteomic pattern samples are taken from a group of patients, and different features are identified to determine whether the patient has ovarian cancer. The input for the network is a variable matrix ($\boldsymbol{u}$) containing data for each patient. Each column in the $\boldsymbol{u}$ matrix represents one patient's data, while the rows indicate the levels of ion intensity at a specific charge/mass, as illustrated in Fig. 3.

The patient's status is based on the ERNN output, represented by a matrix called $\boldsymbol{y}$, which has two rows and the same number of columns as in $\boldsymbol{u}$.

The proposed ERNN model comprises input, hidden, context, and output layers, as shown in Fig. 4.

The input layer receives the features of one patient's data at a time. It has 100 nodes according to the number of features of the database used. To optimize the performance



$$\begin{array}{c|ccccc} & \text{Patient 1} & \text{Patient 2} & \text{Patient 3} & \dots & \text{Patient m} \\ \hline \text{Level 1} & a_{11} & a_{12} & a_{13} & \dots & a_{1m} \\ \text{Level 2} & a_{21} & a_{22} & a_{23} & \dots & a_{2m} \\ \text{Level 3} & a_{31} & a_{32} & a_{33} & \dots & a_{3m} \\ \dots & \dots & \dots & \dots & \dots & \dots \\ \text{Level n} & a_{n1} & a_{n2} & a_{n3} & \dots & a_{nm} \end{array}$$

Fig. 3 Matrix $\boldsymbol{u}$ that comprises the patients' data

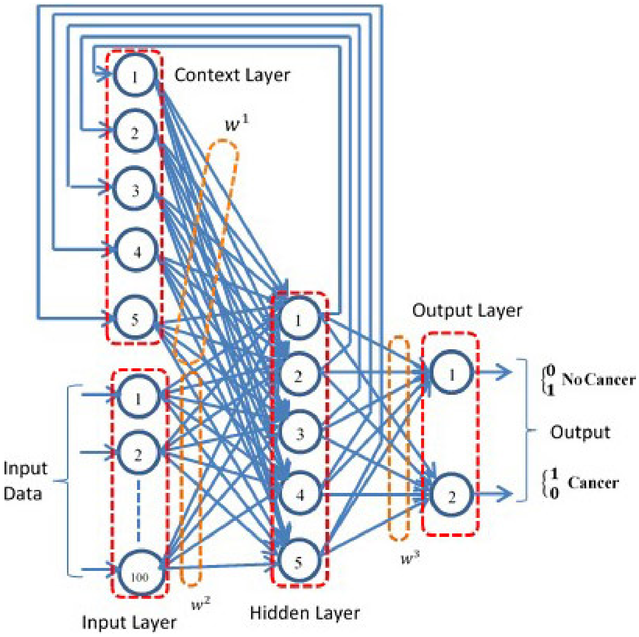


Fig. 4 Proposed ERNN model

and save time, we have chosen to use only five nodes in each hidden and context layer. This fixed parameter allows us to improve other aspects, such as weights and the number of epochs. The trial-and-error method can also determine the number of nodes, but it consumes more time since other parameters need to be iteratively changed. Finally, the output layer includes two nodes with a binary indication (0 or 1). The hidden and output layers use the tan-sigmoid mentioned in Eq. (5) as an activation function. The model is trained using the LM, GDA, GDX and R_{PROP} algorithms to find the proper one. The connections' weights between the hidden layer and the context, input and output layers are denoted by w^1, w^2 and w^3, respectively.

The pairs input/output ($\boldsymbol{u}$, $\boldsymbol{y}$) are generated to train the ERNN on the target output. The weights w^1, w^2 and w^3 are stored after accomplishing the optimum training performance. Furthermore, according to the trained cases, ERNN can predict the output ($\boldsymbol{y}$) for each new input ($\boldsymbol{u}$) that has never been trained to indicate cancer cases. The procedure of ERNN training is demonstrated in Fig. 5.

The ERNN is tested by entering the new patient's charge/mass values, which give the output ($\boldsymbol{y}$) that indicates cancer or a normal case. The judgment of whether a person is ill or healthy is based on the value of the two rows of $\boldsymbol{y}$ (namely [1; 0] or [0; 1] representing a cancer patient or a healthy person, respectively).

5 Experimental results

During the experiments, a computational environment with a 2.4 GHz Core i7 processor, 8 GB of RAM, a 64-bit

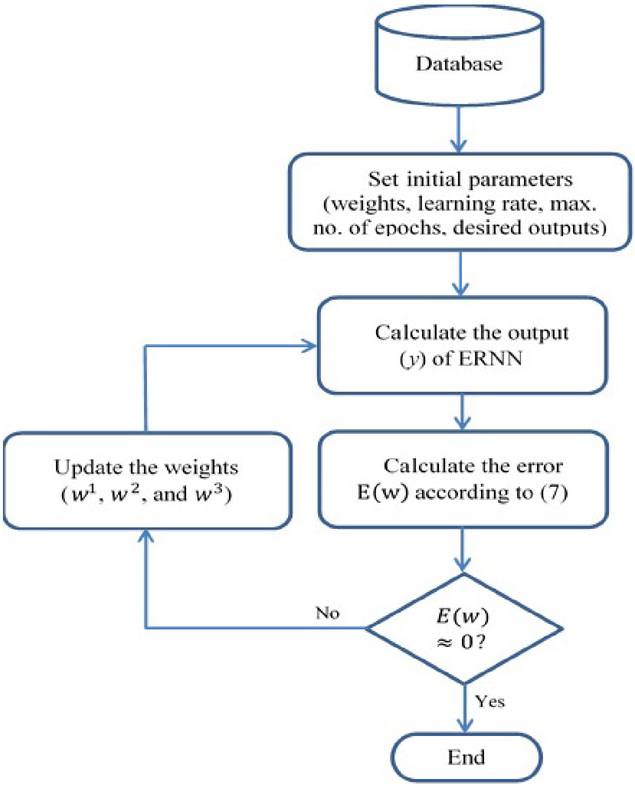

Database
Set initial parameters (weights, learning rate, max. no. of epochs, desired outputs)
Calculate the output (y) of ERNN
Calculate the error E(w) according to (7)
Update the weights (w^1, w^2, and w^3)
$E(w) \approx 0$?
No
Yes
End

Fig. 5 ERNN training procedure

operating system (Windows 10), and MATLAB (R2020b) with the neural network toolbox has been utilized. The proteomic spectra were taken from the Clinical Proteomics Program Databank (FDA-NCI), and the data profile has been created by SELDI mass spectroscopy [2].

Based on the ERNN classifier, the OC detection system was tested for 216 samples of women with 100 features, where 95 out of them are uninfected persons, and 121 are ovarian cancer patients. The data samples were grouped into training (70%), validation (15%), and testing (15%) sets. The training set was used for training the network and updating its parameters, whereas the validation set was used to evaluate the performance of the training process. On the other hand, the testing set was evaluated independently. Once the training process is done, the ERNN is tested on 32 new individuals not part of the training set. The performance of ERNN was evaluated using the mean square error (MSE) between actual and target values as in Eq. (18):

$$\mathrm{MSE} = \frac{1}{n}\sum_{i=1}^{n}\left(\boldsymbol{y}_{i(\mathrm{desired})} - \boldsymbol{y}_{i(\mathrm{actual})}\right)^2 \qquad (18)$$

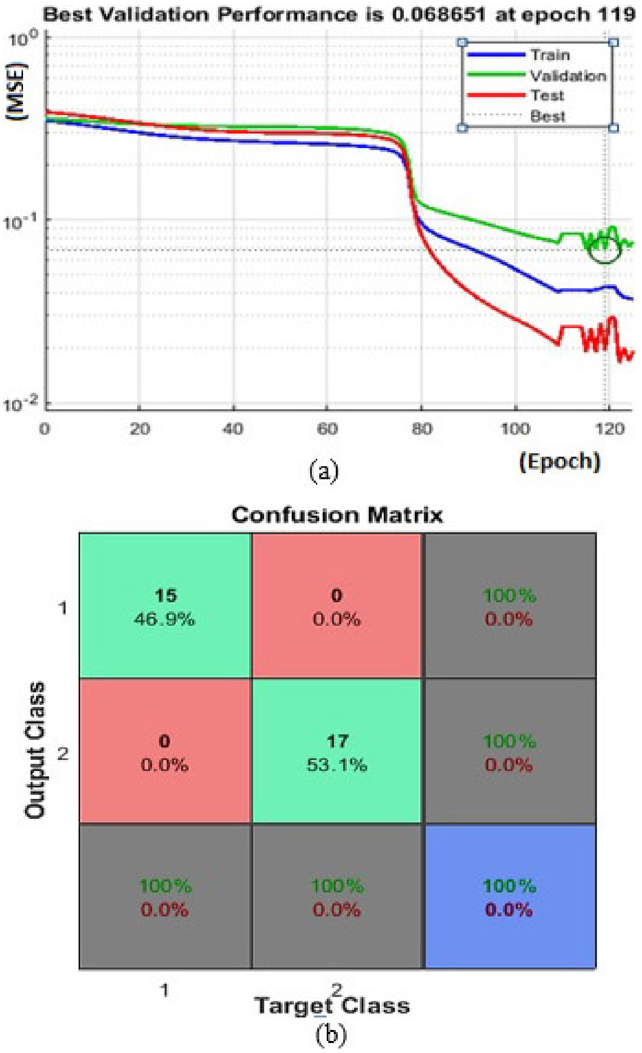
The performances of the four mentioned algorithms used in training ERNN are presented in the following experiments.

5.1 GDA algorithm

The ERNN was trained using the GDA algorithm, and data from 32 individuals were classified as healthy or cancerous. The results of this model are shown in Fig. 6.

The structure of the confusion matrix is illustrated as follows: The green, red and blue squares indicate the percentage of correct, incorrect, and total classification rates, respectively. The gray square in the first row shows the classification rates of cancer cases, whereas the healthy case rates are indicated in the second row's gray square. Finally, the true and false prediction rates for all positive and negative cases are displayed in the 3rd row's 1st and 2nd gray squares, respectively. The top values in the blue and gray squares refer to the accuracy rates, whereas the bottom values indicate the percentage error.

As illustrated in Fig. 6, the implementation of ERNN and GDA achieved a remarkable performance evaluation of about 100%. However, the best performance is achieved after taking 119 epochs. The training, validation, and test performances got an MSE of 0.0433, 0.0686, and 0.0192 at the same epoch.



(a)

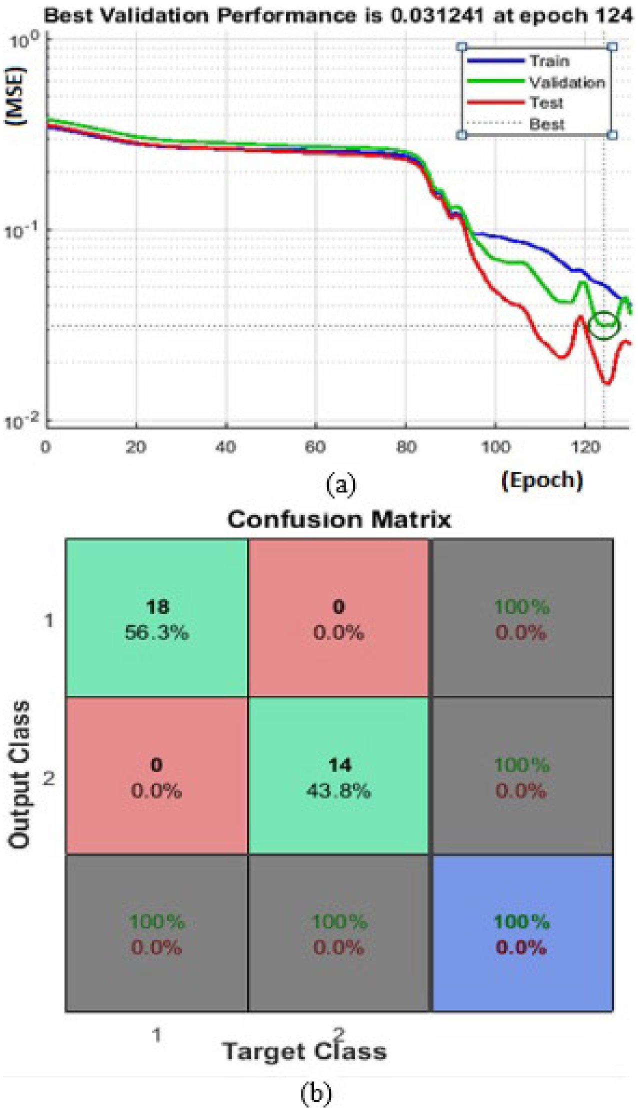
(b)

Fig. 6 Results of using the GDA algorithm; (a) Training performance; (b) Confusion matrix

5.2 GDX algorithm

When utilizing the GDX algorithm to train the ERNN, the network was tested on new data belonging to 32 patients, and the results are shown in Fig. 7.

As seen in Fig. 7, the classification accuracy is 100%, and the optimal performance is achieved after 124 epochs. The best training performance was 0.0511, the best validation performance was 0.0312, and the best test performance was 0.0159. Therefore, the GDX-ERNN method is consistent with the GDA training method.



(a)

(b)

Fig. 7 Results of using the GDX algorithm; (a) Training performance (b) Confusion matrix

5.3 LM algorithm

Similarly, as in the previous experiments, a set of 32 samples not used in the training phase was tested according to the trained ERNN, where the model performs excellently. The LM model was trained and examined using different numbers of epochs, and the results are displayed in Fig. 8.

According to Fig. 8, the best performance of the training algorithm was reached at epoch 6 with an MSE of 0.0035, while the best validation and test performances were 0.0045. Besides, the accuracy of using the LM algorithm to train the ERNN was about 100 %.

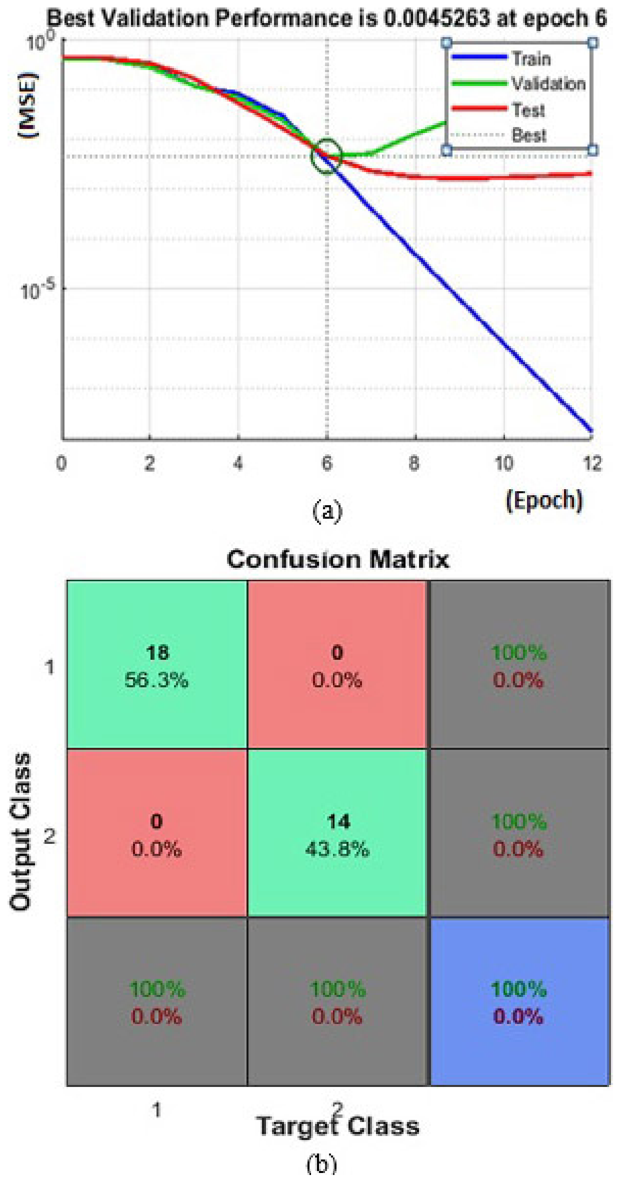
5.4 R_{PROP} algorithm

This experiment was conducted using the R_{PROP} algorithm to train the ERNN. The testing process was conducted on the data used in the previous algorithms, representing 32 samples not used in the training step, and the results are shown in Fig. 9.

From Fig. 9, one can observe the highest accuracy rate achieved at epoch 9 using the R_{PROP} algorithm. The best training performance was 0.0715, the best validation performance was 0.0401, and the best test performance was 0.0242.

5.5 Comparative results

To discriminate between the LM, GDA, GDX, and R_{PROP} algorithms that are used to train the ERNN, a comparison among these algorithms is made in Table 1.

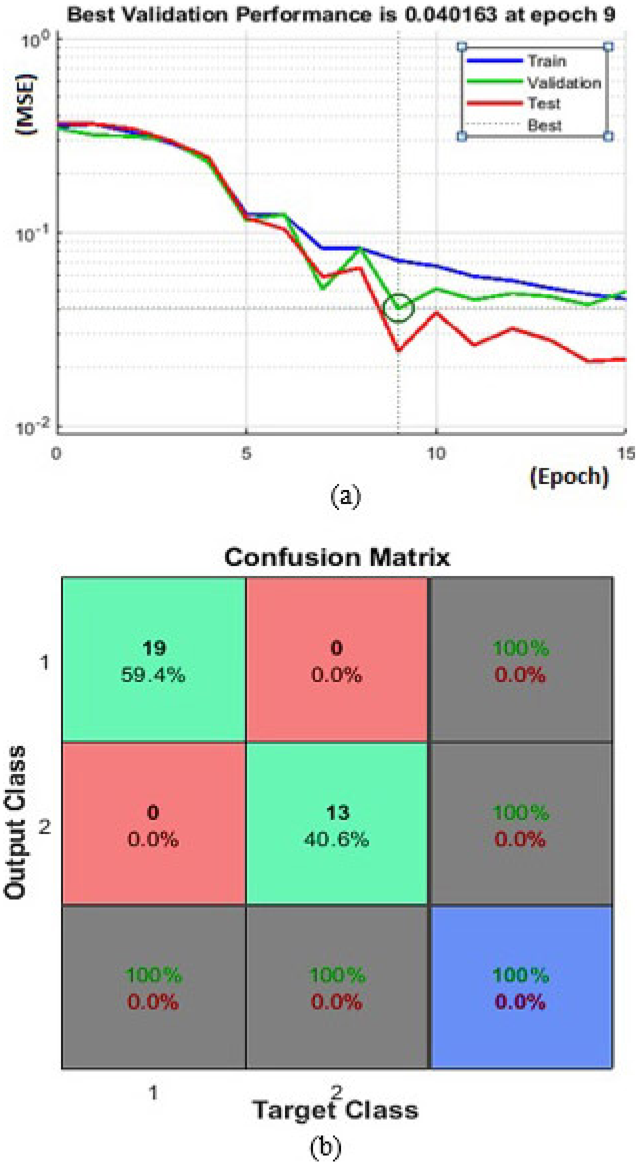


(a)

18 56.3%	0 0.0%	100% 0.0%
0 0.0%	14 43.8%	100% 0.0%
100% 0.0%	100% 0.0%	100% 0.0%

(b)

Fig. 8 Results of using the LM algorithm; (a) Training performance; (b) Confusion matrix



(a)

19 59.4%	0 0.0%	100% 0.0%
0 0.0%	13 40.6%	100% 0.0%
100% 0.0%	100% 0.0%	100% 0.0%

(b)

Fig. 9 Results of using the RPROP algorithm; (a) Training performance; (b) Confusion matrix

It is evident from Table 1 that the LM algorithm has advantages over the others in terms of MSE and the number of training iterations. Therefore, it offers the fastest training backpropagation algorithm, where the speed of this algorithm was modified to be compatible with the speed of the second-order system. Although the R_{PROP} algorithm reaches the best results after nine epochs, the MSE of the training performance is the highest compared to the others.

5.6 Comparison with other approaches

It is better to evaluate the proposed work by comparing it with the previous approaches for detecting OC. Table 2 displays such a comparison based on two key performance metrics: classification accuracy and the number of epochs.

It is obvious from Table 2 that the proposed LM-ERNN method is capable of achieving the highest accuracy rate using just six epochs. Although four epochs were used for [10], it achieved a poor classification accuracy of about 84.27%. The work introduced in [16] achieved an accuracy rate of 98.96 using MLP with GDX and 99.48% using GBC with SVM-RBF, where the rate of 99.48% is the closest to that of the proposed work. However, using multi-stage feature extraction, training algorithm and classifier adds more complexity to the system and consumes extra time. Therefore, the proposed method outperforms the related work regarding accuracy, simplicity, and number of training iterations.

Table 1 Comparative results of ERNN performances using different training algorithms

Training algorithm	LM	GDA	GDX	R_{PROP}
Training performance	0.0035	0.0433	0.0511	0.0715
Validation performance	0.0045	0.0686	0.0312	0.0401
Test performance	0.0045	0.0192	0.0159	0.0242
No. of epochs	6	119	124	9
Accuracy rate (%)	100	100	100	100

Table 2 Comparison between the proposed and related methods

Ref. No.	Method	Accuracy (%)	No. of epochs
[10]	FNN	84.72	4
[11]	EABC	91.2	200
[12]	DCNN	78.2	NA
[13]	15-Neuron Feed Forward	98.7	10
[14]	SOM and ORNN	96.27	10000
[15]	CNN	95	30
[16]	MLP and GDX	98.96	NA
	GBC and SVM-RBF	99.48	NA
[17]	CEA	94.9	NA
[18]	Random Forest	72	NA
[19]	Random Forest	90.5	NA
[20]	SVM	97.16	NA
[21]	CART	98.2	NA
This work	LM-ERNN	100	6

6 Conclusion

Deep learning is used in various medical applications to detect abnormal cases. This study proposes an appropriate neural network architecture with the following advantages:

1. achieving excellent performance,
2. decreasing the processing time,
3. minimizing the capacity of memory usage.

Furthermore, this research has a potential impact on critical medical applications such as healthcare or safety-critical systems. ERNN has been used for OC detection using the mass spectrometry data on protein profiles, distinguishing between control patients and cancer states by selecting a short set of features. These features are the levels of ion intensity at specific mass values, which are used to produce neural network outputs 1 or 0 to indicate cancer or normal patient states, respectively. The ERNN model includes four layers:

- The input layer with 100 nodes;
- The context layer with five neurons;
- The hidden layer with five neurons;
- The output layer with two nodes.

Four backpropagation training algorithms were employed to train the ERNN: LM, GDA, GDX and R_{PROP}. The experiments showed that the mentioned algorithms achieved 100% classification accuracy. Furthermore, using only six epochs, the LM algorithm achieved the lowest MSE, about 0.0035, 0.0045, and 0.0045 for training, validation, and test performances, respectively. The proposed LM-ERNN method is compared to other related work, showing the superiority of the proposed work among other approaches to detect OC.

The proposed method is expected to be able to detect different cancer types and diseases in future works. In addition, some improvements to the network structure will be useful in reducing the processing time and utilizing memory.

References

[1] Momenimovahed, Z., Tiznobaik, A., Taheri, S., Salehiniya, H. "Ovarian cancer in the world: Epidemiology and risk factors", International Journal of Women's Health, 11, pp. 287–299, 2019. https://doi.org/10.2147/IJWH.S197604

[2] Conrads, T. P., Fusaro, V. A., Ross, S., Johann, D., Rajapakse, V., … Veenstra, T. D. "High-resolution serum proteomic features for ovarian cancer detection", Endocrine-Related Cancer, 11(2), pp. 163–178, 2004. https://doi.org/10.1677/erc.0.0110163

[3] Farhan, H. R., Al-Muifraje, M. H., Saeed, T. R. "A new model for pattern recognition", Computers & Electrical Engineering, 83, 106602, 2020. https://doi.org/10.1016/j.compeleceng.2020.106602

[4] Wu, N., Phang, J., Park, J., Shen, Y., Huang, Z., … Geras, K. J. "Deep Neural Networks Improve Radiologists' Performance in Breast Cancer Screening", IEEE Transections on Medical Imaging, 39(4), pp. 1184–1194, 2020. https://doi.org/10.1109/TMI.2019.2945514

[5] Eraslan, G., Avsec, Ž., Gagneur, J., Theis, F. J. "Deep learning: new computational modelling techniques for genomics", Nature Reviews Genetics, 20(7), pp. 389–403, 2019. https://doi.org/10.1038/s41576-019-0122-6

[6] Cohen, L. S., Escobar, P. F., Scharm, C., Glimco, B., Fishman, D. A. "Three-Dimensional Power Doppler Ultrasound Improves the Diagnostic Accuracy for Ovarian Cancer Prediction", Gynecologic Oncology, 82(1), pp. 40–48, 2001. https://doi.org/10.1006/gyno.2001.6253

[7] Khazendar, S., Al-Assam, H., Du, H., Jassim, S., Sayasneh, A., Bourne, T., Kaiser, J., Timmerman, D. "Automated classification of static ultrasound images of ovarian tumours based on decision level fusion", In: 2014 6th Computer Science and Electronic Engineering Conference (CEEC), Colchester, UK, 2014, pp. 148–153. ISBN 978-1-4799-6692-9 https://doi.org/10.1109/CEEC.2014.6958571

[8] Zhang, L., Huang, J., Liu, L. "Retraction Note: Improved Deep Learning Network Based in Combination with Cost-sensitive Learning for Early Detection of Ovarian Cancer in Color Ultrasound Detecting System", Journal of Medical Systems, 47(1), 20, 2023. https://doi.org/10.1007/s10916-023-01915-6

[9] Petricoin, E. F., Ardekani, A. M., Hitt, B. A., Levine, P. J., Fusaro, V. A., … Liotta, L. A. "Use of proteomic patterns in serum to identify ovarian cancer", The Lancet, 359(9306), pp. 572–577, 2002. https://doi.org/10.1016/S0140-6736(02)07746-2

[10] Tan, T. Z., Quek, C., Ng, G. S., Razvi, K. "Ovarian cancer diagnosis with complementary learning fuzzy neural network", Artificial Intelligence in Medicine, 43(3), pp. 207–222, 2008. https://doi.org/10.1016/j.artmed.2008.04.003

[11] Singh, A., Kumar, D. "Novel ABC based training algorithm for ovarian cancer detection using neural network", In: 2017 International Conference on Trends in Electronics and Informatics (ICEI), Tirunelveli, India, 2018, pp. 594–597. ISBN 978-1-5090-4258-6 https://doi.org/10.1109/ICOEI.2017.8300771

[12] Wu, M., Yan, C., Liu, H., Liu, Q., Yin, Y. "Automatic classification of cervical cancer from cytological images by using convolutional neural network", Bioscience Report, 38(6), pp. 1–9, 2018. https://doi.org/10.1042/BSR20181769

[13] Rahman, M. A., Muniyandi, R. C., Islam, K. T., Rahman, M. M. "Ovarian Cancer Classification Accuracy Analysis Using 15-Neuron Artificial Neural Networks Model", In: 2019 IEEE Student Conference on Research and Development (SCOReD), Bandar Seri Iskandar, Malaysia, 2019, pp. 33–38. ISBN 978-1-7281-2614-2 https://doi.org/10.1109/SCORED.2019.8896332

[14] Elhoseny, M., Bian, G., Lakshmanaprabu, S. K., Shankar, K., Singh, A. K., Wanging, W. "Effective features to classify ovarian cancer data in internet of medical things", Computer Networks, 159, pp. 147–156, 2019. https://doi.org/10.1016/j.comnet.2019.04.016

[15] Tanabe, K., Ikeda, M., Hayashi, M., Matsou, K., Yasaka, M., … Mikami, M. "Comprehensive Serum Glycopeptide Spectra Analysis Combined with Artificial Intelligence (CSGSA-AI) to Diagnose Early-Stage Ovarian Cancer", Cancers, 12(9), 2373, 2020. https://doi.org/10.3390/cancers12092373

[16] Prabhakar, S. K., Lee, S.-W. "An Integrated Approach for Ovarian Cancer Classification with the Application of Stochastic Optimization", IEEE Access, 8, pp. 127866–127882, 2020. https://doi.org/10.1109/ACCESS.2020.3006154

[17] Lu, M., Fan, Z., Xu, B., Chen, L., Zheng, X., Li, J., Znati, T., Mi, Q., Jiang, J. "Using machine learning to predict ovarian cancer", International Journal of Medical Informatics, 141, 104195, 2020. https://doi.org/10.1016/j.ijmedinf.2020.104195

[18] Akter, L., Akhter, N. "Ovarian Cancer Classification from Pathophysiological Complications using Machine Learning Techniques", In: 2012 12th International Conference on Computing Communication and Networking Technologies (ICCCNT), Kharagpur, India, 2021, pp. 1–6. ISBN 978-1-7281-8596-5 https://doi.org/10.1109/ICCCNT51525.2021.9580067

[19] Aditya, M., Amrita, I., Kodipalli, A., Martis, R. J. "Ovarian Cancer Detection and Classification Using Machine Leaning", In: 2021 5th International Conference on Electrical, Electronics, Communication, Computer Technologies and Optimization Techniques (ICEECCOT), Mysuru, India, 2021, pp. 279–282. ISBN 978-1-6654-3273-3 https://doi.org/10.1109/ICEECCOT52851.2021.9707954

[20] Taleb, N., Mehmood, S., Zubair, M., Naseer, I., Mago, B., Nasir, M. U. "Ovary Cancer Diagnosing Empowered with Machine Learning", In: 2022 International Conference on Business Analytics for Technology and Security (ICBATS), Dubai, United Arab Emirates, 2022, pp. 1–6. ISBN 978-1-6654-8841-9 https://doi.org/10.1109/ICBATS54253.2022.9759010

[21] Farinella, F., Merone, M., Bacco, L. et al. "Machine Learning analysis of high-grade serous ovarian cancer proteomic dataset reveals novel candidate biomarkers", Scientific Reports, 12(1), 3041, 2022. https://doi.org/10.1038/s41598-022-06788-2

[22] Chow, T. W., Cho, S. Y. "Neural Networks and Computing: Learning Algorithms and Applications", Series in Electrical and Computer Engineering, 7, Imperial College Press, 2007. https://doi.org/10.1142/p487

[23] Medsker, L. R., Jain, L. C. "Recurrent Neural Network: Design and Applications", International Series on Computational Intelligence, CRC Press, USA, 1999. ISBN 9781420049176

[24] Elman, J. L. "Finding Structure in Time", Cognitive Science, 14(2), pp. 179–211, 1990. https://doi.org/10.1016/0364-0213(90)90002-E

[25] Şeker, S., Ayaz, E., Türkcan. E. "Elman's recurrent neural network applications to condition monitoring in nuclear power plant and rotating machinery", Engineering Applications of Artificial Intelligence, 16(7–8), pp. 647–656, 2003. https://doi.org/10.1016/j.engappai.2003.10.004

[26] Ghaemi, N., Madaeni, S. S., Abolhasani, M., Zahedi, G., Rajabi, H. "Modeling the Presence of Humic Acid in Ultrafiltration of Xenobiotic Compounds: Elman Recurrent Neural Network", Chemical Engineering & Technology, 34(11), pp. 1891–1898, 2011. https://doi.org/10.1002/ceat.201100112

[27] Devasena, L., Latha, M. H. "Automatic classification of audio data using gradient descent neural network based algorithm", Journal of Theoretical and Applied Information Technology, 70(3), pp. 375–389, 2014. [online] Available at: http://www.jatit.org/volumes/Vol70No3/3Vol70No3.pdf [Accessed: 26 July 2023]

[28] Shrestha, A., Mahmood, A. "Review of Deep Learning Algorithms and Architectures", IEEE Access, 7, pp. 53040–53065, 2019. https://doi.org/10.1109/ACCESS.2019.2912200

[29] Björck, Å. "Numerical methods for Least Squares Problems", SIAM, 1996. ISBN 978-0-89871-360-2 https://doi.org/10.1137/1.9781611971484

[30] Riedmiller, M., Braun, H. "A Direct adaptive method for faster backpropagation learning: The RPROP algorithm", In: IEEE International Conference on Neural Networks, San Francisco, CA, USA, 1993, pp. 586–591. ISBN 0-7803-0999-5 https://doi.org/10.1109/ICNN.1993.298623

[31] Igel, C., Hüsken, M. "Empirical evaluation of the improved Rprop learning algorithms", Neurocomputing, 50, pp. 105–123, 2003. https://doi.org/10.1016/S0925-2312(01)00700-7