Deep Learning Hybrid with Binary Dragonfly Feature Selection for the Wisconsin Breast Cancer Dataset

Marian Mamdouh Ibrahim[1*], Dina Ahmed Salem[2], Rania Ahmed Abdel Azeem Abul Seoud[3]
PhD researcher - Faculty of engineering, Fayoum University, Egypt, Fayoum[1]
Assistant Professor - Computer Department - Faculty of Engineering
Misr University for Science and Technology (MUST).Egypt, Cairo[2]
Professor of digital signals at the Department of Electrical Engineering
Faculty of Engineering Fayoum University Egypt, Fayoum[3]

***Abstract*—Breast cancer is the world's top cancer affecting women. While the danger of the factors varies from a place, lifestyle, and diet. Treatment procedures after discovering a confirmed cancer case can reduce the risk of the disease. Unfortunately, breast cancers that arise in low and middle-income countries are diagnosed at a very late stage in which the chances of survival are impeded and reduced. Early detection is therefore required not only to improve the accuracy of discovering breast cancer but also to increase the chances of making the right decision on a successful treatment plan. There have been several studies tending to build software models utilizing machine learning and soft computing techniques for cancer detection. This research aims to build a model scheme to facilitate the detection of breast cancer and to provide the exact diagnosis. Improving the accuracy of a proposed model has, therefore, been one of the key fields of study. The model is based on deep learning that intends to develop a framework to accurately separate benign and malignant breast tumors. This study optimizes the learning algorithm by applying the Dragonfly algorithm to select the best features and perfect parameter values of the deep learning model. Moreover, it compares deep learning results against that of support vector machine (SVM), random forest (RF), and k nearest neighbor (KNN). Those classifiers are chosen as they are the most reliable algorithms having a solid fingerprint in the field of clinical data classification. Consequently, the hybrid model of deep learning combined with binary dragonfly has accurately classified between benign and malignant breast tumors with fewer features. Besides, deep learning model has achieved better accuracy in classifying Wisconsin Breast Cancer Database using all available features.**

Keywords*—Breast cancer; Wisconsin data set; classifiers; deep learning; feature selection; dragonfly

I. INTRODUCTION

Breast cancer is the most common cancer in women and, overall, the second most leading to death. In 2019, women were diagnosed with an estimated 268,600 new cases of invasive breast cancer and approximately 2,670 cases were diagnosed in men [1]. An accurate diagnosis for various sorts of cancer plays a great role for doctors to assist them in determining and choosing the proper treatment. Lately, the application of various artificial intelligence (AI) classification methods has been proven in aiding doctors to facilitate their decision-making process [2]. Recently, the use of AI classification techniques in the medical field generally and cancer detection particularly has grabbed the researchers' attention. AI is beneficial in reducing medical human-errors (because it minimizes possible errors) that might occur due to unskilled doctors [3].

More research is being done on breast cancer diagnosis using the Wisconsin Breast Cancer Database (WBCD)[4]. Many methods have been constantly developed to achieve accurate and efficient diagnosis results and several experiments were performed on the WBCD using multiple classifiers and feature selection techniques. Many of them show a good classification accuracy, for example, in [5] the performance criterion of supervised learning classifiers such as Naïve Bayes (NB), Support Vector Machine (SVM-RBF) kernel, and neural networks (NN) are compared to find the best classifier using the dataset (WBCD), and the SVM-RBF has the best outcome achieving 96.84%. The robustness of the least square Support Vector Machine (SVM) obtained a classification accuracy of 98.53% [6]. In [7] Linear Regression achieved an average training accuracy of 96.093%, whereas Multilayer Perceptron (MLP) is 99.038%, Softmax Regression has an average training accuracy of 97.366573% and the accuracy obtained by SVM (97.13%) is better than the accuracy obtained by KNN [8]. The prediction accuracy of the SVM (linear kernel) in [9] reaches 97.14%, an accuracy of 95.71% using RBF kernel, and 97.14% using RF classifier for Breast Cancer detection. The accuracy obtained from the system which combines rough set theory with backpropagation neural network in [10] is 98.6% on the breast cancer dataset. The first stage handles missing values to obtain a smooth data set and to select appropriate attributes from the clinical dataset by the indiscernibility relation method. The second stage is classification using a backpropagation neural network. The algorithm KNN for classification which is used in [11] with several different types of distances and classification rules is used in the diagnosis and classification of cancer, and these experiments are conducted on the database WBCD. The results advocate the use of the KNN algorithm with both types of Euclidean distance and Manhattan that give the best results (98.70% for Euclidean distance and 98.48% for Manhattan with k = 1), these values are not significantly affected even when k=1 is increased to 50. SVM and KNN individually used in [12] achieved the accuracy of 98.57% and 97.14%, respectively. This work aims to automatically design and modify the parameters of the deep learning model hybrid with the Dragonfly algorithm for breast cancer diagnosis.

*Corresponding Author

II. Materials and Methods

A. Machine Intelligence Library

The software, developed for implementation in this study is written by using Spyder which is an interactive development environment capable of advanced editing, interactive testing, debugging, and introspection for Python (version 3.7 was used) programming language. Also, Keras [13] neural network API was used for deep learning in the developed method. It is a high-level neural network API, supporting Python which can convert the results rapidly, highly modular, minimalist, and has extensible features. Keras with Google TensorFlow backend is used to implement the deep learning algorithms in this study, with the aid of other scientific computing libraries: matplotlib [14] is a comprehensive library for creating interactive, and animated visualizations in Python, NumPy [15] is a library for the Python programming language, adding support for big, multi-dimensional arrays and matrices, along with a huge collection of high-level mathematical functions to operate on these arrays, and scikit-learn [16] is a free software machine learning library for the Python programming language, where it emphasizes several classifications, regression, and clustering algorithms including support vector machines, k-means, random forests.

B. Dataset Description

The UCI machine learning repository has been used to download the WBCD [4] for breast cancer classification [17]. This dataset usage is more common among researchers who utilize machine learning methods for the classification of breast cancer. Each dataset is composed of a set of numerical attributes that were assessed by fine needle aspiration (FNA) from human breast cancer tissue. WBCD has 699 instances and 10 attributes including the class attribute. One of the two possible classes is found in each instance; malignant (M) or benign (B). Every attribute has been represented in the form of an integer between 1 and 10. These attributes include: (uniformity of cell size, clump thickness, uniformity of cell shape, single epithelial cell size, marginal adhesion, bare nuclei, normal nuclei, bland chromatin, and mitosis).

C. Data Preprocessing

Preparing data for use in a machine learning (ML) framework is significant, where data preparation requires at least 80 percent of the total time expected to create an ML system. Data preparation has three main phases: cleaning, normalizing, and encoding, and splitting. Each of the three phases has several steps. Equation (1) is used to normalize dataset attributes.

$$Z = \frac{X-\mu}{\sigma} \quad (1)$$

Where X represents the dataset attributes, μ represents the mean value for each dataset attribute x(i), and σ represents the corresponding standard deviation. This normalization technique was implemented using the Standard Scaler of scikit-learn.

D. Principal Component Analysis

Principal Component Analysis (PCA) [18] is a dimension reduction method that includes related features. Dimensionality reduction [19] is a process used in Data Mining where the numbers of random variables under consideration are reduced. An essential step in the efficient analysis of large high-dimensional data sets is the reduction of dimensions. PCA performs dimensionality reduction whilst maintaining maximum feasible arbitrariness in the high-dimensional space. PCA is probably the oldest and certainly the most popular technique for computing lower-dimensional representations of multivariate data. The technique is linear in the sense that the components are linear combinations of the original variables (features), but non-linearity in the data is preserved for effective visualization. The PCA is a method of statistical data analysis that transforms the initial set of variables into an assorted set of linear combinations, referred to as the principal components (PC), with variance-specific properties. This condenses the system's dimensionality while retaining the variable connections information. The analysis is carried out by calculating and analyzing the data covariance matrix on a data set, its eigenvalues along with its respective eigenvectors systematized in descending order.

E. Classification Techniques

The classification aims to develop a set of models that can correctly classify the class of different objects. There are three types of inputs to such models, which are: (a) a bunch of objects that are described as training data, (b) the dependent variables, and (c) classes that may be a group of variables describing various characteristics of the objects. Once a classification model is built, it tends to be utilized to classify the class of the objects to which class information is unidentified [20]. There are numerous sorts of classifiers that have been utilized for a cancer diagnosis; some of them are NN, SVM, KNN, NB, and RF. They are used to classify cancer datasets as malignant and benign tumors.

1) Support vector machine: Support vector machine (SVM) classifier is a type of supervised machine learning classification algorithm, it is applied in classifying cancer because it is a non-probabilistic binary and nonlinear statistical tool which works by separating space into two regions by a straight line or hyperplane in higher dimensions. It examines the data, recognizes the pattern, and classifies the data based on common attributes by using kernel tricks. The kernel is a set of numerical functions that are used in SVM. The kernel's function is to take data as an input and convert it into the form necessary. Various kinds of kernel functions were utilized by the SVM algorithm. These functions can be different types; for example, linear, nonlinear, polynomial, radial basis (RBF), and sigmoid functions.

2) Naïve Bayes: Naïve Bayes (NB) is a probabilistic classifier based on the Bayes theorem. Rather than predictions, it produces probability estimates. For the value of each class, it estimates the probability of each given instance belongs to that class. An advantage of the NB classifier is that it requires a small amount of training data in order to estimate the parameters that are mandatory for classification.

3) Artificial Neural Network: Artificial Neural Network (ANN) is a numerical model based on biological neural networks. It comprises an interconnected group of artificial

neurons, and it processes information employing a connectionist approach to the computing process. In most cases, an ANN is a robust framework that changes its structure based on outside or inner data that flows through the network during the learning phase. One of the fundamental advantages of ANN over conventional methods is its ability in capturing the complex and nonlinear interaction between prognostic markers and the outcome to be anticipated.

4) Random forest (RF): Random forest (RF) algorithm is a supervised classification algorithm that creates a forest with several trees. It is a flexible, easy to utilize machine learning algorithm that mostly produces a great result. Due to its simplicity, it is also one of the most used algorithms. The more trees in the forest the more robust the forest appears in general. In the same way in the random forest classifier, the higher the number of trees in the forest indicates the high accuracy results.

5) K-nearest Neighbors: K-nearest Neighbors (KNN) is one of the most used algorithms in machine learning. It is a method of learning based on instances that do not require a phase of learning. The model developed is the training sample, connected to a distance function and the choice function of the class based on the classes of the nearest neighbors. Before classifying a new element, it must be compared with other elements using a similarity measure. Its k-nearest neighbors consider the class that appears most among the neighbors is assigned to the element to be classified. Besides, the appropriate functioning of the method relies on the choice of some number of a parameter such as the k parameter represents the number of neighbors chosen to assign the new element to the class and the distance used.

III. Deep Learning

Deep learning (DL) is one of the numerous strategies found within machine learning (ML) as shown in Figure 1, where ML [21] is a discipline of artificial intelligence that ensures the software estimate results with better accuracy, without the need to write explicit codes to perform the task mentioned. DL methods are utilized in ML in terms of quick learning and implementation of large and complex data. DL is widely utilized in numerous software disciplines for example computer vision, speech and sound processing, bioinformatics, computer games, search engines, manufacturing, online advertising, and financing, etc. It is realized that DL provides highly successful results in processes of estimation and classification.

DL describes a bunch of computational models composed of many layers of data processing, which make it conceivable to learn by representing these data through several levels of abstraction [22] from a large amount of training data, these models discover recurrent structures by automatically refining their interior parameters via a backpropagation algorithm as shown in Figure 2. Each layer of the network transforms the signal nonlinearly to increase the selectivity and invariance of the representation. With a sufficient number of layers, the network can generate a hierarchy of representations that will make the model both sensitive to very small details and insensitive to large variations. The classification issue is an important component in the field of deep learning since it is focused on judging a new sample that belongs to which predefined sample category, according to a train set containing a certain number of known samples. The classification problem is also called supervised classification, since all samples in the train set are labeled, and all categories are predefined.

The output is defined by the following formula in (2):

$$Y = f(\sum_j w_j x_j + b) \quad (2)$$

Where w_j is the network weights, b is a bias term, and f is a specified activation function. Figure 3 shows a natural extension of this simple model is attained by combining multiple neurons to form a so-called hidden layer.

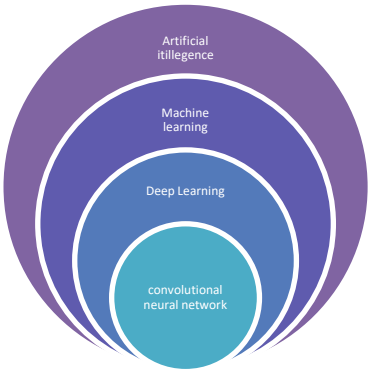


Fig. 1. The Relationship between Artificial Intelligence, Machine Learning, Deep Learning, and Artificial Neural Networks.

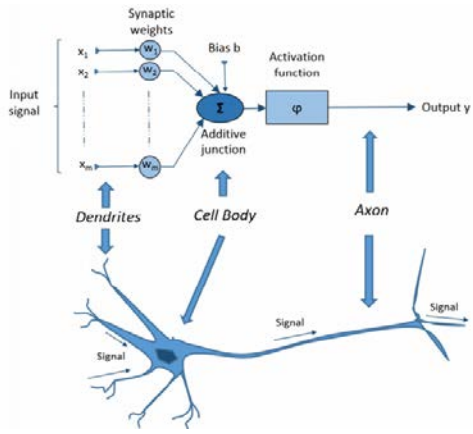


Fig. 2. The Analogy between an Artificial Neuron and a Biological Neuron. X Represents the Inputs, the Bias b, the Activation Function ϕ, and Weights w are Adjusted Automatically by the Network.

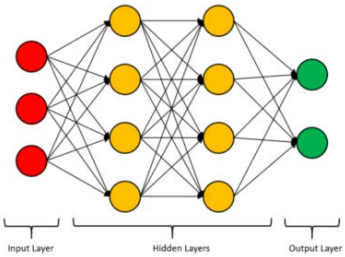


Fig. 3. Representation of Layers of Deep Learning.

IV. Feature Selection

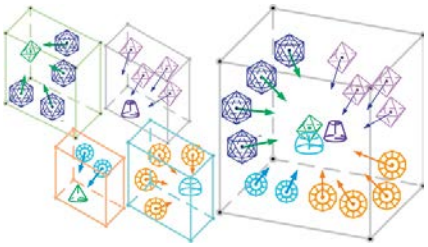
Feature selection (FS) is a pre-processing method that has been demonstrated to significantly affect the performance of the data mining techniques [23] (e.g. classification) in terms of either the quality of the extracted patterns or the running time required to analyze the complete dataset. It reduces the number of features, removes irrelevant, redundant, or noisy features, and brings about palpable effects for applications: speeding up a data mining algorithm, improving learning accuracy, and leading to better model comprehensibility's methods are arranged into filters and wrappers [24]. While a learning algorithm (e.g. classification) is approached by the wrapper in the evaluation of the feature subset, filters rely on the data itself to evaluate the feature subset using designated methods (e.g. information gain) [25].

Searching for an ideal subset of features is a major challenge when solving feature selection problems. The primary target when selecting a feature is to find a set of M features from an original set of N where M < N without information lose. Therefore, an impractical approach to this problem is to create all possible subsets. If the dataset includes N features, then there will be 2N subsets to be generated and evaluated, which are considered computationally expensive tasks [23]. This paper introduces Dragonfly as a feature selection and studies its effect on accuracy.

A. Dragonfly Algorithm (DA)

Dragonfly is an open-source python library for scalable Bayesian optimization. Bayesian optimization is utilized for optimizing black-box functions whose evaluations are usually expensive. Beyond vanilla optimization techniques, DA provides an array of tools to scale up Bayesian optimization to expensive large-scale problems. These include features that are especially suited for high dimensional optimization, parallel evaluations in synchronous or asynchronous settings which means conducting multiple evaluations in parallel, multi-fidelity optimization which using cheap approximations to speed up the optimization process, and multi-objective optimization which optimizing multiple functions simultaneously. It is compatible with Python2 (>= 2.7) and Python3 (>= 3.5) and has been tested on Linux, Mac OS, and Windows platforms.

DA is a recently well-established population-based optimizer proposed by Mirjalili in 2016 [26]. The hunting and migration strategies of dragonflies are the base of the DA algorithm. The hunting method is known as a static swarm (feeding), in which all members of a swarm can fly in small clusters over a limited space for discovering food sources. Dynamic swarming is considered the migration strategy of dragonflies (migratory). In this phase, the dragonflies are eager to take off in bigger clusters, and as a result, the swarm can migrate. Dynamic and static groups are shown in Figure 4. Moreover, in other swarm-based methods, the operators of DA perform two main concepts: intensification, encouraged by the dynamic swarming activities, and diversification, motivated by the static swarming activities.



(a) Static Swarm. (b) Dynamic Swarm.

Fig. 4. Dynamic and Static Dragonflies.

Five behaviors characterize DA, where X is the position vector, X_j is the j-th neighbor of the X, and N denotes the neighborhood size:

Separation: dragonflies use this strategy to separate themselves from other agents. This procedure is formulated as (3):

$$S_i = \sum_{j=1}^{N} X - X_i \quad (3)$$

Alignment: shows how an agent will set its velocity to the velocity vector of other adjacent dragonflies. This concept is modeled based on (4) Where Vj indicates the velocity vector of the j-th neighbor:

$$A_i = \frac{\sum_{j=1}^{N} Vi}{N} \quad (4)$$

Cohesion: shows members' inclination to move in the direction of the nearest mass center. This step is formulated as in (5):

$$C_i = \frac{\sum_{j=1}^{N} X_j}{N} - X \quad (5)$$

Attraction: illustrates the propensity of members to step towards the food source. The attraction tendency among the food source and the i-th agent is performed based on (6) Where Floc is the food source's location:

$$F_i = F_{loc} - X \quad (6)$$

Distraction: illustrates the proclivity of dragonflies to keep themselves away from a conflicting enemy. The distraction among the enemy and the i-th dragonfly is performed according to (7) Where Eloc is the enemy's location:

$$E_i = E_{loc} + X \quad (7)$$

In DA, the fitness of food source and position vectors are updated based on the fittest agent found so far. Moreover, the fitness values and positions of the enemy are calculated based on the worst dragonfly. This fact will help DA converge in the solution space towards more promising regions and in turn, avoid non-promising areas. The position vectors of dragonflies are updated based upon two rules: the position vector and the step vector (X). The step vector indicates the dragonflies' direction of motion and it is calculated as in (8):

$$X_{t+1} = (sS_i + aA_i + cC_i + f F_i + eE_i) + wX_t \quad (8)$$

Where s, w, a, c, f, and e show the weighting vectors of different components. The location vector of members is calculated as in Eq. (9), where t is iteration:

$$X_{t+1} = X_t + X_{t+1} \quad (9)$$

The pseudo-code of the DA algorithm is given in Algorithm 1 as follows:

```
Initialize the population Xi (i = 1, 2, ..., n)
Initialize ΔXi (i = 1, 2, ..., n)
while (t < Max_Iteration)
Evaluate each dragonfly
Update (F) and (E)
Update the main coefficients(i.,e., w, s, a, c, f, and e)
Calculate S, A, C, F, and E(using Eqs. (3) to (7))
Update step vectors using Eq. (8)
Calculate T(ΔX) using Eq. (9)
Update Xt+1 using Eq. (8)
Return the best agent
End while
```

V. Experimental Discussion

The data is split into a training set (80%), testing set (10%), and validation sets (10%) several times, and a Cross-Validation (CV) approach was utilized to evaluate the accuracy, sensitivity, and specificity of each of the classifiers with five folds.

A. Model

The proposed model in this study is represented in a block diagram as shown in Figure 5 explaining the process conducted within the model. It is planned with three phases first of them utilizing traditional classifiers for example SVM, NB, RF, and KNN, secondly applying deep learning for enhancing the accuracy of the detection of breast cancer, finally applying the principle of feature selection using DA with deep learning classification to improve the performance.

B. Missing Dataset Technique

The training of a model with a dataset containing missing values may significantly influence the quality of the deep learning model. For this reason, the utilization of WBCD in training was ensured by the correction of 16 incorrect data found with statistical missing value analysis. There are two techniques in handling the missing data: The mean Imputation technique and Missing Data Ignoring Technique. Mean Imputation technique works by calculating the mean value of readily available values in a column and then substituting the missing values in each column independently from each other [27]. Missing Data Ignoring Technique simply deletes the cases that contain missing data.

C. Normalization of Dataset

A normalization process between the ranges of 0-1 was applied in the data set for eliminating the long learning time caused by the size of the data set. The MinMaxScaler method was used in this process as shown in (1).

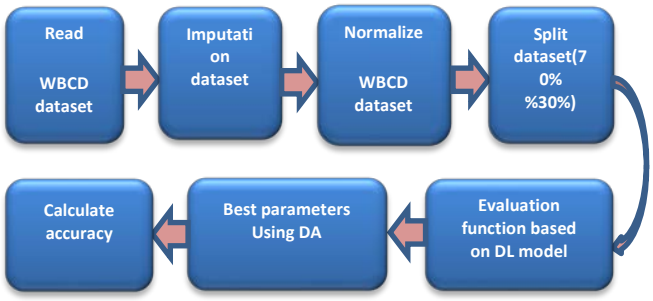


Fig. 5. Block Diagram of the Model.

D. Splitting the Dataset

The data used in experiments are separated into two groups as training and testing using (train_test_split) library in python. The allocation of available data among these three data sets is vital for the objectivity of success. Because of different tests, the data set in the suggested model was allocated as 80% (559 data) for training, 20% (140 data) for testing and validating. The process of allocation is shown in Figure 6. The cross-validation method was utilized in the implementation of this process [28].

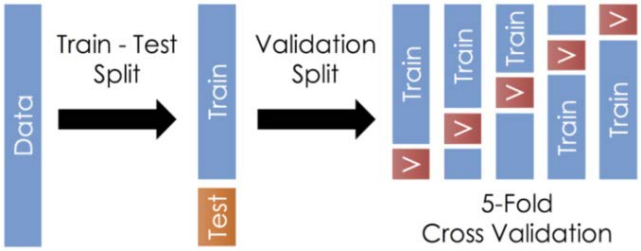


Fig. 6. Splitting Dataset using Cross-validation.

E. Neural Network Model

Any neural network (NN) model has multiple parameters that control its performance for example number of hidden layers, number of nodes in each layer, the type of activation function, number of epochs, and batch size.

F. Epoch

The NN learns the patterns of input data by reading the input dataset and applying various calculations to it. However, it does not make that only once, it learns, over and over, utilizing the input dataset and learning outcomes from the previous trials. An epoch is a process of learning from the input dataset in each trial. Expanding the number of epochs doesn't generally imply that the network will give better results, it may cause overfitting. Using the trial-and-error method, several epochs were chosen until the outcomes still the same after a very few cycles.

G. Activation Functions of the Neural Network

The activation functions used in the layers of the created neural network are described as follows:

1) Rectified Linear Unit (ReLU): ReLU activation function is utilized in the input layer and hidden layers of the neural network. ReLU as seen in Figure 7 is an activation

function that recently gained popularity for its practicality in deep learning. It enables the neural network to learn faster [21]. The numerical expression of the function is provided in (10).

$$f(x) = \max(0, x) \quad (10)$$

2) Sigmoid activation function: The sigmoid activation function is used in the output layer of the neural network. It is a function that gets a value between the ranges of (0, 1) as seen in Figure 8. The numerical expression of the function is given in (11).

$$\sigma(x) = \frac{1}{1+\exp(-x)} \quad (11)$$

3) Softmax activation function: The Softmax is a function that turns a vector of K real values into a vector of K real values that sum to 1. The input values can be positive, negative, zero, or greater than one, but the Softmax transforms them into values somewhere in the range 0 and 1 as shown in Figure 9, so that they can be interpreted as probabilities. Large multi-layer neural networks end in a penultimate layer that outputs real-valued scores that are not efficiently scaled and which makes working with them complicated. In the current study, the Softmax is very helpful as it turns the scores into a normalized probability distribution. Consequently, it is normal to append a Softmax function as the final layer of the neural network.

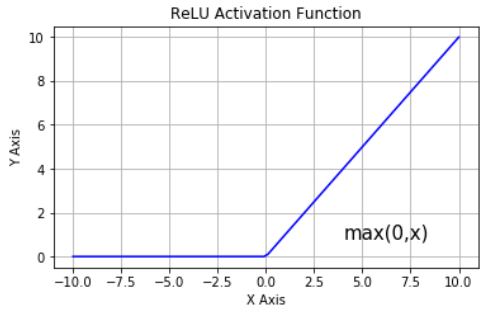


Fig. 7. Relu Activation Function.

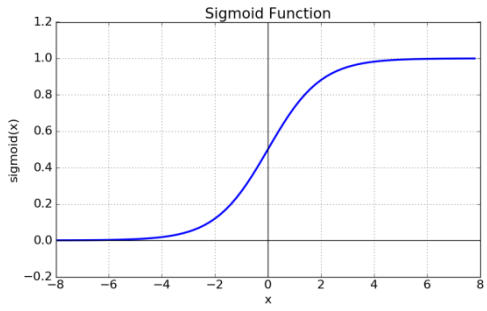


Fig. 8. Sigmoid Function Activation.

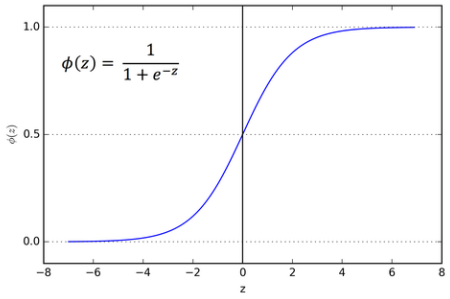
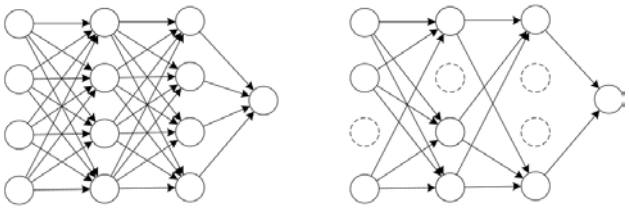


Fig. 9. Softmax Activation Function.

H. Dropout

Dropout is one of the methods that are utilized to prevent memorization. In each iteration, it randomly removes some neurons from a layer at a specified rate. The process of dropout is shown in Figure 10. They dropped the crossed units out of the network.



(a) Standard Neural Network. (b) Network after Dropout.

Fig. 10. Dropout Neural Network Model. (a) A Standard Neural Network. (b) After the Dropout is Applied, the Same Network. Dotted Lines Indicate a Node that has been Dropped.

I. Optimization

Optimization is a basic issue in the learning process in deep learning applications. Its techniques are utilized to find the optimum value in solving non-linear problems. RMSprop, adagrad, adadelta, adam, adamax. Moreover, there are differences between each of these algorithms in terms of performance and speed. In this study, the optimization algorithm of Adaptive Moment Optimization (Adam) was applied.

J. Loss Function

The loss function is a type of function that measures both the error rate and performance of a designed model. In DL, the last layer of a NN is the layer where the loss of function is defined. In DL applications, the function calculates the dissimilarity between the estimation of the designed model and the required real value. In case that a model with good estimation capability is designed, the difference between the real value and estimated value will be lower. An output of a higher loss value indicates that the designed model contains defects. In the literature, there are various loss functions such as mean squared error, mean absolute percentage error, mean squared logarithmic error, hinge, logcosh, sparse categorical cross-entropy, binary cross-entropy, kullback, poisson, and many others. In this study, the meager straight out cross-entropy misfortune work was utilized.

K. Early Stopping

In the models where training is made by iteration with data, the duration of learning must be terminated at the right time. Otherwise, if training is not stopped, all of the samples in the data set for training will be memorized by the system. These outcomes in a decrease in the capability of estimation of unknown samples. In case of early termination, the performance of the system will decline in that it could not fully analyze the data. The same outcome will also arise in the case of over-training. In case of an overfitting possibility for the program, a parameter of early stopping was defined; the training will be stopped regardless of the number of iterations.

VI. Results and Discussion

In this section, performance evaluation is discussed using accuracy which is used as the percentage of correct predictions. Table 1 shows the comparative study of different classifiers which easily analyses KNN, RBF, SVM, and NB. Some experiments handled missing data by Mean Imputation technique and others by Missing Data Ignoring Technique. It declares that RF gives a better result when using 10 trees, and KNN with 3 neighbors which reduces the complexity of the model and consumes less processing time. PCA+SVM with RBF kernel when using missing data ignoring technique considered as a better classifier as compared to others which achieved 99%.

A. Deep Learning Usage with the Dataset

The proposed model utilizes two layers at the start, then eventually experiments with more layers which have been observed that the convergence time is larger for deeper networks. Many parameters control the deep learning model. One of them is the number of hidden layers, if data is less complex and is having fewer features then neural networks with 1 to 2 hidden layers will work, but if data is having large features, so to get an optimum solution, 3 to 5 hidden layers can be used. It should be noticed that increasing hidden layers will also increase the complexity of the model which may sometimes lead to overfitting. Another one is the number of hidden neurons; it should be between the size of the input layer and the size of the output layer. It may be 2/3 the size of the input layer, plus the size of the output layer and it should be less than twice the size of the input layer [29]. The experiments are based on using batch size 16 and 9 neurons in each layer; the result is as shown in Table 2.

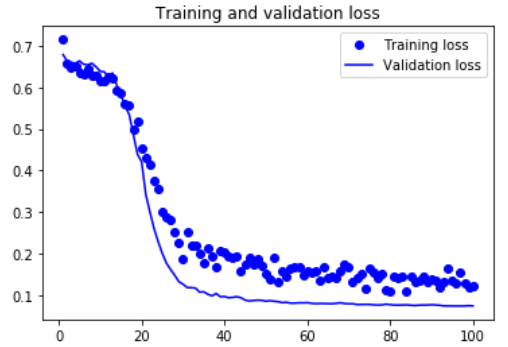
As shown above, in Table 2, the best accuracy achieved is 99.3% with 2 hidden layers and epochs 2000 while the accuracy reduces to 99% with 250 epochs only. More epochs mean more iteration and more consumption of time and resources. However, the difference in accuracy is not significantly considerable to endure more time consumption. Also, the same accuracy level of 99.3% is attained using 4 hidden layers and only 100 epochs. Besides, plots of the characteristic of the 4 hidden layers model are shown in Figure 11. In graph (a) the training accuracy visibly increases over time, until it reaches nearly 95%, while the validation accuracy reaches a plateau at a range of 98–99.3% after 21epochs. Moreover, the validation loss, presented in a graph (b), reaches its minimum after 50 epochs and then halts, while the training loss keeps decreasing exponentially until it drops to nearly 0.

TABLE I. Results of Traditional Classifiers

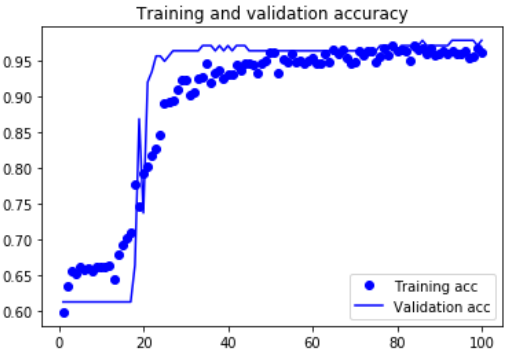
classifier	**Missing values**	**PCA**	**accuracy**
RF (100)	Mean		97
RF (10)	Mean		95
RF (10)	Mean	1	**98**
RF (100)	Mean	1	98
RF (10)	Remove		98
RF (100)	remove		95
KNN (10)	Mean	1	97
KNN (3)	Mean	1	**98**
KNN (3)	Remove		96
NB	Mean	1	97
Svm (rbf)	Remove	1	**99**
Svm (rbf)	Mean		96
Svm (rbf)	Remove		97
Svm (rbf)	Mean	1	98
Svm (linear)	Mean	1	97

TABLE II. Deep Learning Results

Number Of layers	Epochs	Activation functions	Dropout	accuracy
2	250	Sigmoid,Softmax	0.5	**99**
2	100	Sigmoid,Softmax	0.3	98.54
2	100	Sigmoid,Softmax	0.5	97.85
2	2000	Relu,Sigmoid		**99.3**
3	150	Sigmoid,Sigmoid,Softmax	0.3	98
3	150	Relu,Relu,Softmax	0.3	97
3	250	Relu,Relu,Softmax	0.3	97.08
3	1000	Relu,Relu,Softmax	0.3	97.08
3	100	Relu,Sigmoid,Softmax	0.5	**98.54**
3	100	Relu,Sigmoid,Softmax	0.3	97.8
4	250	Sigmoid,Sigmoid,Sigmoid ,Softmax	0.5	99
4	1000	Sigmoid,Sigmoid,Sigmoid ,Softmax	0.3	98.5
4	100	Sigmoid,Sigmoid,Sigmoid ,Softmax	0.3	**99.3**
4	150	Sigmoid,Sigmoid,Sigmoid ,Softmax	0.3	97.08
4	150	Sigmoid,Softmax,Softmax ,Softmax	0.3	98.54
5	15	Sigmoid,…,softmax		93.3
5	15	Softmax ,…,softmax		94.3
5	20	Softmax,…,softmax		95.2
5	250	Sigmoid,…,softmax	0.25	96

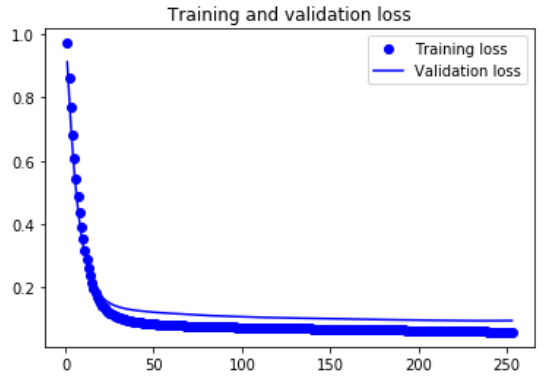


(a)

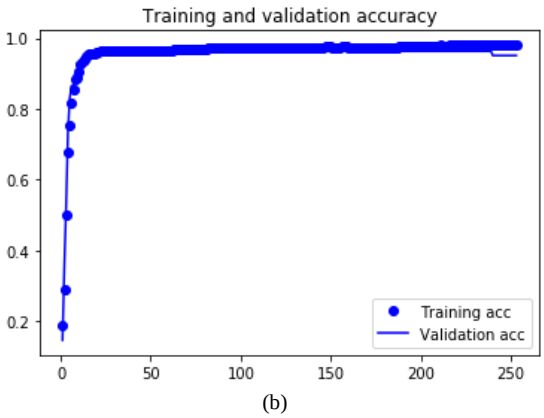


(b)

Fig. 11. (a) Training vs Validation Loss, (b) Training vs Validation Accuracy of 4 Hidden Layers Model with 100 Epochs.



(a)



(b)

Fig. 12. Two Hidden Layers Model with 2000 Epoch, (a) Training vs Validation Loss (b) Training vs Validation Accuracy.

The characteristics of the 2 hidden layers model are shown in Figure 12, where graph (a) explains the behavior of the training and validation loss and graph (b) presents the accuracy. The validation loss clearly reaches its minimum after 200 epochs and then halts, while after 250 epochs keep decreasing exponentially and reach the minimum value for nearly 0 then it is steady-state. Also, the training and validation accuracy increases linearly until 30 epochs, and then it reaches nearly 100%.

After applying feature selection using DA with deep learning, the results are shown in Table 3. As noticed in Table 3, it gives the best result is 97.907% when choosing 20 population,100 iteration, and 100 epochs which recommended five attributes as the most important (uniformity of cell size, uniformity of cell shape, bare nuclei, bland chromatin, and mitosis). These ML methods have been chosen because the results obtained from these methods have appeared to be more accurate than traditional classifiers. In addition to implementing these ML techniques for bigger data in the future will be at a faster rate. The main focus is to choose the most suitable classifier model for obtaining the highest accuracy and to find an improvement of similar previous works on the same database.

TABLE III. DEEP LEARNING MIXED WITH DA RESULTS

folds	population	iteration	epoch	features	Number of features	Activation function	accuracy
5	10	100	100	110011001	5	Relu,Sigmoid	96.52
5	20	100	100	011001010	4	Relu,Sigmoid	97.43
10	10	100	100	110011000	4	Relu,Sigmoid	96.69
10	20	100	100	111001000	4	Relu,Sigmoid ,Softmax	97.619
10	30	150	100	111011001	6	Relu,Sigmoid	97.62
10	20	100	1000	110001001	4	Relu,Sigmoid ,Sigmoid,Softmax	97.818
5	20	100	200	101001111	6	Sigmoid,Softmax ,Softmax,Softmax **Dropout=0.25**	97.256
10	20	100	100	101001011	5	Softmax,Softmax ,Softmax,Softmax	**97.907**
10	20	100	100	001001001	3	Sigmoid,Sigmoid ,Softmax **Dropout=0.5**	96.71
10	20	100	100	110101011	6	Relu,Sigmoid	96.89
10	20	100	1000	100111011	6	Relu,Sigmoid ,Sigmoid	97.49

VII. CONCLUSION

Breast cancer prediction is very significant in the area of Medicare and Biomedical. This study aims to enhance the accuracy of the diagnosis of breast cancer with the deep learning method. Analysis of WBCD with traditional classifiers such as NB, SVM, KNN, and RF achieved high accuracy. Proposed a model that predicts breast cancer based on a deep investigation in the performance of different deep

networks on this dataset. It has been implemented by Python to be the most effective in classifying the diagnostic data set into the two classes because of the seriousness of cancer; it's found that the accuracy of the proposed model ranges between 93.5% and 99.3%. In the case of the two hidden layers model, the highest outcomes result with 250 and 2000 epochs are 99% and 99.3% respectively. The same result might be obtained with four hidden layers models and 100 epochs. It is noticed that DL hybrid with DA as a feature selection model achieved an accuracy of 97.907%. Such comparative analysis of breast cancer classification would provide insights on the efficient approaches for the detection of cancer problems.

VIII. Future Work

The proposed model is applied to numerical data only. It would be interesting to see its behavior when it is applied to different types of data available in the medical field such as mammograms. In the future, the research may be carried out for a screening of features to diagnose breast cancer tumors.

References

[1] S. Chopra and E. L. Davies, "Breast cancer," Med. (United Kingdom), vol. 48, no. 2, pp. 113–118, 2020, doi: 10.1016/j.mpmed.2019.11.009.

[2] B. Sahu, S. Mohanty, and S. Rout, "A Hybrid Approach for Breast Cancer Classification and Diagnosis," ICST Trans. Scalable Inf. Syst., vol. 0, no. 0, p. 156086, 2018, doi: 10.4108/eai.19-12-2018.156086.

[3] M. Paredes, "Can Artificial Intelligence help reduce human medical errors? Two examples from ICUs in the US and Peru," vol. 2009, pp. 1–12, 2018, [Online]. Available: https://techpolicyinstitute.org/wp-content/uploads/2018/02/Paredes-Can-Artificial-Intelligence-help-reduce-human-medical-errors-DRAFT.pdf.

[4] Dr. WIlliam H. Wolberg, "UCI Machine Learning Repository: Breast Cancer Wisconsin (Original) Data Set." https://archive.ics.uci.edu/ml/datasets/Breast+Cancer+Wisconsin+%28Original%29 (accessed Dec. 16, 2020).

[5] S. Aruna, S. P. Rajagopalan, and L. V Nandakishore, "Knowledge Based Analysis of Various Statistical Tools in Detecting Breast Cancer," Comput. Sci. Inf. Technol., vol. 2, pp. 37–45, 2011, doi: 10.5121/csit.2011.1205.

[6] K. Polat and S. Güneş, "Breast cancer diagnosis using least square support vector machine," Digit. Signal Process. A Rev. J., vol. 17, no. 4, pp. 694–701, Jul. 2007, doi: 10.1016/j.dsp.2006.10.008.

[7] A. F. M. Agarap, "On breast cancer detection: An application of machine learning algorithms on the Wisconsin diagnostic dataset," ACM Int. Conf. Proceeding Ser., no. 1, pp. 5–9, 2018, doi: 10.1145/3184066.3184080.

[8] H. Asri, H. Mousannif, H. Al Moatassime, and T. Noel, "Using Machine Learning Algorithms for Breast Cancer Risk Prediction and Diagnosis," Procedia Comput. Sci., vol. 83, no. Fams, pp. 1064–1069, 2016, doi: 10.1016/j.procs.2016.04.224.

[9] P. S. Kohli and A. L. Regression, "2020 IEEE 5th International Conference on Computing Communication and Automation, ICCCA 2020," 2020 IEEE 5th Int. Conf. Comput. Commun. Autom. ICCCA 2020, pp. 1–4, 2020.

[10] K. B. Nahato, K. N. Harichandran, and K. Arputharaj, "Knowledge mining from clinical datasets using rough sets and backpropagation neural network," Comput. Math. Methods Med., vol. 2015, no. April, 2015, doi: 10.1155/2015/460189.

[11] S. AhmedMedjahed, T. Ait 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, doi: 10.5120/10041-4635.

[12] M. M. Islam, H. Iqbal, M. R. Haque, and M. K. Hasan, "Prediction of breast cancer using support vector machine and K-Nearest neighbors," 5th IEEE Reg. 10 Humanit. Technol. Conf. 2017, R10-HTC 2017, vol. 2018-Janua, no. February 2018, pp. 226–229, 2018, doi: 10.1109/R10-HTC.2017.8288944.

[13] H. Singh, Practical Machine Learning with AWS. 2021.

[14] J. D. Hunter, "Matplotlib: A 2D graphics environment," Comput. Sci. Eng., vol. 9, no. 3, pp. 90–95, 2007, doi: 10.1109/MCSE.2007.55.

[15] S. Van Der Walt, S. C. Colbert, and G. Varoquaux, "The NumPy array: A structure for efficient numerical computation," Comput. Sci. Eng., vol. 13, no. 2, pp. 22–30, 2011, doi: 10.1109/MCSE.2011.37.

[16] H. Li and D. Phung, "Journal of Machine Learning Research: Preface," J. Mach. Learn. Res., vol. 39, no. 2014, pp. i–ii, 2014.

[17] L. Vig, "Comparative Analysis of Different Classifiers for the Wisconsin Breast Cancer Dataset," OALib, vol. 01, no. 06, pp. 1–7, 2014, doi: 10.4236/oalib.1100660.

[18] Y. Qu, G. Ostrouchov, N. Samatova, and A. Geist, "Principal Component Analysis for Dimension Reduction in Massive Distributed Data Sets," Work. High Perform. Data Min. Second SIAM Int. Conf. Data Min., no. June 2014, pp. 4–9, 2002.

[19] N. Varghese, "A Survey Of Dimensionality Reduction And Classification Methods," Int. J. Comput. Sci. Eng. Surv., vol. 3, no. 3, pp. 45–54, 2012, doi: 10.5121/ijcses.2012.3304.

[20] V. Saravanan and R. Mallika, "An effective classification model for cancer diagnosis using micro array Gene expression data," Proc. - 2009 Int. Conf. Comput. Eng. Technol. ICCET 2009, vol. 1, pp. 137–141, 2009, doi: 10.1109/ICCET.2009.38.

[21] İ. Yıldız and A. T. Karadeniz, "Enhancement Of Breast Cancer Diagnosis Accuracy With Deep Learning," Eur. J. Sci. Technol., no. October, pp. 452–462, 2019, doi: 10.31590/ejosat.638428.

[22] Y. Bengio, Learning deep architectures for AI, vol. 2, no. 1. 2009.

[23] M. M. Mafarja, D. Eleyan, I. Jaber, A. Hammouri, and S. Mirjalili, "Binary Dragonfly Algorithm for Feature Selection," Proc. - 2017 Int. Conf. New Trends Comput. Sci. ICTCS 2017, vol. 2018-Janua, pp. 12–17, 2017, doi: 10.1109/ICTCS.2017.43.

[24] H. (National U. of S. Liu, H. (Osaka U. Motoda, R. Setiono, and Z. Zhao, "Feature Selection : An Ever Evolving Frontier in Data Mining," J. Mach. Learn. Res. Work. Conf. Proc. 10 Fourth Work. Featur. Sel. Data Min., pp. 4–13, 2010.

[25] C. S. Yang, L. Y. Chuang, Y. J. Chen, and C. H. Yang, "Feature selection using memetic algorithms," Proc. - 3rd Int. Conf. Converg. Hybrid Inf. Technol. ICCIT 2008, vol. 1, pp. 416–423, 2008, doi: 10.1109/ICCIT.2008.81.

[26] M. Mafarja, A. A. Heidari, H. Faris, S. Mirjalili, and I. Aljarah, Dragonfly algorithm: Theory, literature review, and application in feature selection, vol. 811. Springer International Publishing, 2020.

[27] Q. Song and M. Shepperd, "Missing data imputation techniques," Int. J. Bus. Intell. Data Min., vol. 2, no. 3, pp. 261–291, 2007, doi: 10.1504/IJBIDM.2007.015485.

[28] D. Berrar, "Cross-validation," Encycl. Bioinforma. Comput. Biol. ABC Bioinforma., vol. 1–3, no. April, pp. 542–545, 2018, doi: 10.1016/B978-0-12-809633-8.20349-X.

[29] F. S. Panchal and M. Panchal, "International Journal of Computer Science and Mobile Computing Review on Methods of Selecting Number of Hidden Nodes in Artificial Neural Network," Int. J. Comput. Sci. Mob. Comput., vol. 3, no. 11, pp. 455–464, 2014, [Online]. Available: www.ijcsmc.com.