IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received January 19, 2019, accepted January 28, 2019, date of publication February 1, 2019, date of current version February 20, 2019.

Digital Object Identifier 10.1109/ACCESS.2019.2897078

Analytics of Heterogeneous Breast Cancer Data Using Neuroevolution

BEIBIT ABDIKENOV[1], ZANGIR IKLASSOV[1], ASKHAT SHARIPOV[1], SHAHID HUSSAIN[2], AND PRASHANT K. JAMWAL[1], (Member, IEEE)

[1]Electrical and Computer Engineering Department, Nazarbayev University, 010000 Astana, Kazakhstan
[2]Human Centred Technology Research Centre, Faculty of Science and Technology, University of Canberra, Canberra, ACT 2617, Australia

Corresponding author: Prashant Jamwal (prashant.jamwal@nu.edu.kz)

This work was supported by the Faculty Development Competitive Research Grants, Nazarbayev University, under Grant 090118FD5322.

ABSTRACT Breast cancer prognostic modeling is difficult since it is governed by many diverse factors. Given the low median survival and large scale breast cancer data, which comes from high throughput technology, the accurate and reliable prognosis of breast cancer is becoming increasingly difficult. While accurate and timely prognosis may save many patients from going through painful and expensive treatments, it may also help oncologists in managing the disease more efficiently and effectively. Data analytics augmented by machine-learning algorithms have been proposed in past for breast cancer prognosis; and however, most of these could not perform well owing to the heterogeneous nature of available data and model interpretability related issues. A robust prognostic modeling approach is proposed here whereby a Pareto optimal set of deep neural networks (DNNs) exhibiting equally good performance metrics is obtained. The set of DNNs is initialized and their hyperparameters are optimized using the evolutionary algorithm, NSGAIII. The final DNN model is selected from the Pareto optimal set of many DNNs using a fuzzy inferencing approach. Contrary to using DNNs as the black box, the proposed scheme allows understanding how various performance metrics (such as accuracy, sensitivity, F1, and so on) change with changes in hyper-parameters. This enhanced interpretability can be further used to improve or modify the behavior of DNNs. The heterogeneous breast cancer database requires preprocessing for better interpretation of categorical variables in order to improve prognosis from classifiers. Furthermore, we propose to use a neural network-based entity-embedding method for categorical features with high cardinality. This approach can provide a vector representation of categorical features in multidimensional space with enhanced interpretability. It is shown with evidence that DNNs optimized using evolutionary algorithms exhibit improved performance over other classifiers mentioned in this paper.

INDEX TERMS Breast cancer prognostic modelling, entity embedding, deep learning networks, evolutionary algorithms, fuzzy inferencing.

I. INTRODUCTION

Breast cancer is predominantly diagnosed in women while it's a rare medical condition in males accounting for only 1% of all breast cancers [1]. It adversely affects the physiological as well as psychological health of subjects and can be fatal in some cases which is apparent from its high morbidity and mortality rates [2]. Breast cancer is also one of the predominant causes of cancer-related deaths worldwide and its rate of incidence is ever increasing. In 2012 women in US had 12.4% risk of being diagnosed with breast cancer in their life span while it was only 9.09%, in 1970's [2]. However, the increase in this proportion may be due to longer life expectancy, changes in the environment and lifestyle apart from increased awareness and scrutiny. It has been reported in the literature that while there were about 14.9 million new incidences of breast cancer in the world in 2012 the figure is estimated to reach up to 22 million in coming two decades [3]. It is important to note here that breast cancer incidences amount to 25% of all types of cancers which makes it the second most probable cancer [3]. Incidence rate of breast cancer also differs by places and ranges from a minimum of 19.4 over 100,000 people in East Africa to a maximum of 89.7 over 100,000 subjects in West Europe [4]. With formation of cancer cells in breast tissues, disease progresses and eventually the growth becomes out of control. Early stage of

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

TABLE 1. Databases for breast cancer.

Database	Sample size
SEER Research Data (1973 – 2015), USA	1 631 572
Wisconsin Breast Cancer Data, USA	569
Nottingham Primary Breast Cancer, UK	1076
Srinagarind Hospital, Thailand (1985 – 2006)	4312
Clinical Center of Kragujevac, Croatia	146
National Cancer Institute, Egypt	60

breast cancer can be detected by signs and symptoms such as swelling, skin irritation, breast pain, nipple pain, redness, thickening of nipple and a nipple discharge etc., [5]. Nevertheless, the disease can be prevented, treated or managed substantially on account of precise prognosis and diagnosis. Through prognostication it is possible to predict occurrence and severity of breast cancer besides further course of disease. Accurate and reliable prognosis can provide information on the type and intensity of required therapeutic intervention [6], [7]. This will save many patients from going through unnecessary painful and expensive treatment protocols and at the same time alert medical practitioners in deciding on intensity of treatment for some urgent cases [8].

Prognosis on the other hand requires rigorous analysis and synthesis of past available data on breast cancer. Over the time this database has increased enormously in length and breadth [9], [10]. There are number of breast cancer databases like Wisconsin breast cancer diagnosis(WBCD) [11], breast cancer data collected in the University of Nottingham [12], UCI Breast Cancer Database [13], Surveillance, Epidemiology, and End Results (SEER) data [14] and data from hospitals in Croatia, Egypt and Thailand [15]. Although there is good number of cancer databases available, most of these databases have undesirably small sample size. The SEER database, on the other hand, has records and statistics stored since 1973 till date which makes it one of the largest cancer databases. In the present research, we have accessed and used the SEER breast cancer database. Details of available breast cancer databases from literature are presented in Table 1. Apparently, traditional statistical tools, owing to their inherent limitations, may not comprehend and process the vast and diverse cancer databases and therefore machine learning tools have become leading tools in health informatics in recent years [16].

Rapid development in the machine learning field provides an opportunity to develop and train sophisticated models on larger datasets. To cite an example, deep convolutional neural networks model proposed by [17] and [18] achieved improved result on image classification task. Recently, the state of the art models show phenomenal performance on image and object recognition [19]. In addition, artificial neural networks (ANNs) drastically changed computer vision, speech recognition and natural language processing [20]–[22]. ANNs are looked upon as suitable candidates to replace traditional methods across the disciplines. As a matter of fact, ANNs can approximate continuous as well as arbitrary non-continuous information which predominantly is the case while dealing with cancer data.

The structured medical data comprises of heterogeneous variables, which are continuous and categorical in nature. The discrete categorical data can further be presented in two scales, which are nominal and ordinal. In case of ordinal scale, entities of categorical variable are assigned a numeric index. However, arbitrary numbers do not represent true distance measure between entities involved. For instance, we usually assign number grades from one to three to evaluate some event for expressions, such as, ''bad'', ''average'' and ''good''. These numbers only inform us about some order but fail to provide information or exact measures of distances between these entities. Categorical variables can also be represented by nominal scale providing appropriate labels. A suitable example to cite here could be the marital status in socio-economic or demographic analysis where entities are ''single'', ''married'', ''divorced'' and ''widowed''. Once again it is difficult to interpret as how the labels are related. Unfortunately, due to the non-continuous nature of data, majority of machine learning algorithms may not perform reasonably well on categorical data. A conventional approach to address this issue is to use one- hot encoding [23], but it also has two drawbacks. First, when categorical variables have high cardinality one-hot encoding outputs large size vector which influences computational complexity. Secondly, every entity of categorical variable treated independently without considering intrinsic relations between them and this affects prognosis precision to some extent apart from overfitting data. A possible solution to these drawbacks is an appropriate transformation of categorical data into numerical vector space. This is also termed as entity embedding of the categorical data [24]. Subsequently, wide range of methods adopted for continuous data can be applied on this transformed numerical vector.

Continuous vector representation, which is also called word embedding, was first introduced by Paccanaro and Hinton [25]. Word embedding made a significant breakthrough in various natural language processing challenges in last few years, which involves language modeling [26], machine translation [27] and text classification [28]. It has different implementations, but the prominent and robust ones are GloVe [29] and fastText [30]. The principle working of these models is to transform the semantic or syntactic attributes of a word into a low dimensional continuous vector representation. This further resolves issues encountered due to curse of dimensionality and similarity between words. Although, word embedding had been used in past for language modelling and text classification tasks, recently few research works have been proposed using entity embedding in the context of processing of categorical data. Entity embedding proposed by Zhang *et al.* also showed promising results [31]. In the present research, we propose to use entity embedding on heterogeneous breast cancer data and then develop a predictive model for prognosis.

Borrowing concepts of neuroevolution, deep neural networks (DNNs) have been employed which are optimized with evolutionary algorithms for binary classification of breast cancer survivability. Main contributions from the proposed research work include implementation of entity embedding on large breast cancer data and evolutionary optimization of DNNs for enhanced robustness and interpretability.

While most of the previous related works consider prediction accuracy as the sole objective, a many-objective optimization approach (optimizing many performance metrics simultaneously) using state of the art evolutionary algorithm is being proposed for the first time. While there exist a number of evolutionary algorithms [31]–[33], a recently proposed NSGAIII is implemented here which performs well while optimizing many objectives simultaneously [34].

The rest of the paper is structured as follows. Section II provides information about neuroevolution and its significance in many-objective optimization of DNNs. Discussion on the breast cancer database used in this research and the entity embedding of categorical data is provided in Section III. Section IV introduces state of the art evolutionary algorithm NSGA-III. Neuroevolution using NSGA-III is also discussed in Section IV along with apropos data preprocessing and experimental design. Section V presents performance metrics used for evaluation of the classifiers or the predictive models along with details of NSGAIII implementation. A fuzzy logic based approach for the selection of a single best DNN model from Pareto optimal set of DNN models is explained in Section VI. Details of experiments performed using other classifiers such as Logistic Regression, support vector machine (SVM), Random Forest and Gradient Boosting are provided in Section VII. Results from various experiments are illustrated and discussed in Section VIII whereas conclusion and future directions of the current research are given in Section IX.

II. NEUROEVOLUTION

Deep learning or sometimes referred as hierarchical learning paradigm applied to neural networks, is a computational generalization of the human biological information processing system. There exists a special class of neural networks, where learning is achieved through evolution process and hence they are termed as Evolutionary artificial neural networks [35], [36]. These neural networks are evolved using evolutionary algorithms, which are population based search methods inspired from Darwinian evolution [37].

Neuroevolution is about finding an alternative to the conventional neural network (NN) training algorithm called backpropagation which is a form of stochastic gradient descent approach. It is expected that owing to their inherent evolutionary mechanism, these algorithms may conquest the otherwise standard learning algorithms such as backpropagation and hybrid schemes. With its advent in 1980s, weights of the complex network were obtained using evolutionary approach keeping the architecture unchanged. This approach was called fixed-topology neuroevolution [38]. Although, by changing weights, the intensity or strength of neurons' knowledge was evolved but a new knowledge could not be evolved since the architecture was still the same. Further, randomly generated networks in the initial population lacking appropriate information also did not improve network performance. Later in 1990s, researchers experimented with changing network topology besides weights and it was termed as topology and weight evolving ANNs (TWEANNs) [39]. One of the popular algorithms called NEAT among Augmenting Topologies was widely used in neuroevolution [40]. In order to speed up the ANN training and reduce number of function evaluations, researchers from neuroevolution community proposed an alternative class of genetic encoding and named it as indirect encodings, which employs fewer numbers of genes than the number of connections and neurons in the network. This further facilitated evolving larger ANNs which were otherwise difficult to train using NEAT. Compositional pattern-producing networks (CPPNs) is one of such endeavors of indirect encodings [41]. One of the other successful researches worth mentioning here is the idea that in the successive evolutionary process, parents which are novel should be preferred over those which give better network accuracies. This new paradigm was known as novelty search and apparently networks were trained to be accurate and at the same time explore better alternative solutions [42]. However, there are other evolutionary approaches such as the one based on Fogel's evolutionary programming which emphasizes on the evolution of NN's behaviors through its architecture [43].

Until recently the focus of network training has been to enhance its performance or in other words reduce the network error in mapping outputs and inputs of a given system. Such approaches may not be robust and may results in overfitting the training data [44]. We propose that during training, the network performance should also be evaluated on the basis of other metrics such as precision, recall/sensitivity, and F1 score apart from the accuracy. While precision is the ratio of correctly predicted positive observations to the total predicted positive observations, recall reveals how many of positive observations were actually found. The weighted average of Precision and Recall is termed as F1 Score.

Evaluating a network on more than one performance metric transforms training into a multiple objective optimization problem. During selection stage of evolution, population of generated networks will now be evaluated not only for accuracy but for multiple objectives which are none other than the performance metrics. Recently, NSGAII (Non-dominated Sorting Genetic Algorithm), a popular evolutionary algorithm (EA), has been used to perform multi-objective optimization of neural networks minimizing the perceptron error and the network complexity [45]. However, applications of EAs to optimize neural networks have been limited to the variation in network hyper-parameters such as network architecture, number of neurons and parameters namely; connection weights [45]–[50].

To the best knowledge of authors, many-objective optimization of deep neural networks (DNN) considering various

TABLE 2. Entity embedding of categorical variables.

Names	Ethnicity
Dmitry	Russian
Abdul	Arab
Ramesh	Indian

Dmitry	Abdul	Ramesh	Ethnicity
1	0	0	Russian
0	1	0	Arab
0	0	1	Indian

network parameters and hyper-parameters simultaneously has not been done. In the present research, an advanced version of NSGAII [51]evolutionary algorithm which is called NSGAIII [34], [37], [52], has been used for many-objective optimization of deep neural networks. Hyper-parameters chosen for DNN many-objective optimization are learning rate, number of layers, number of neurons in each layer, choice of activation function, and number of iterations. The hyper-parameters of the network are optimized using evolutionary algorithm NSGA III, whereas the parameters such as connection weights between layers and biases are obtained using Levenberg-Marquardt optimization.

III. DATABASE AND ENTITY EMBEDDING OF CATEGORICAL DATA

Breast cancer data used in this study is obtained from the SEER database (http://seer.cancer.gov/), which is available in the public domain. The Surveillance, Epidemiology, and End Results (SEER) program of the National Cancer Institute, United States (US), is an authentic source of information on cancer incidence, prevalence, mortality, treatment, and other related information. This large database consists of comprehensive information from 28% of the US cancer affected population since 1973 [15]. Out of ten million cancer cases from 1973 up to 2015, about one million cases are of breast cancer [15]. Before using SEER data, appropriate procedure for accessing the database was followed during this research.

The SEER database discussed above is a high dimensional database consisting of both continuous as well as categorical variables. Analysis of such a heterogeneous data is difficult unless the categorical variables are converted into continuous domain or mapped into logical data form, a process which is also termed as entity embedding [53]. There are a number of data encoding schemes to handle categorical variables such as one-hot encoding, hash encoding, ordinal and target encoding etc. [23]. The aim of entity embedding is to map discrete values to a multi-dimensional embedding space where values with similar function output are placed close to each other. This is further explained using a Table 2 below. Entity embedding using one-hot encoding coverts ethnicity feature to three features through binarization meaning: ''is_Russian'', ''is_Arab'', ''is_Asian'' etc.

However, a neural network based entity embedding has been used in the present work owing to its capability of handling high cardinality categorical variables which is otherwise cumbersome using one-hot encoding [24]. Readers are encouraged to read a paper by Guo and Berkhahn [54] for further information on entity embedding of categorical variables. In the present work, the categorical variables are initially represented using widely used one hot encoding and then later these are multiplied with weights of an extra layer of linear neurons [55]. This extra layer is termed as the embedding layer. Following the standard practice, weights and other parameters of neural network of embedding layer are learned using Levenberg-Marquardt method [55]. Binary outputs obtained from the neural network (transformed categorical variables) are combined together with the continuous variables and given as inputs to the respective classifiers for further cancer prognosis [31].

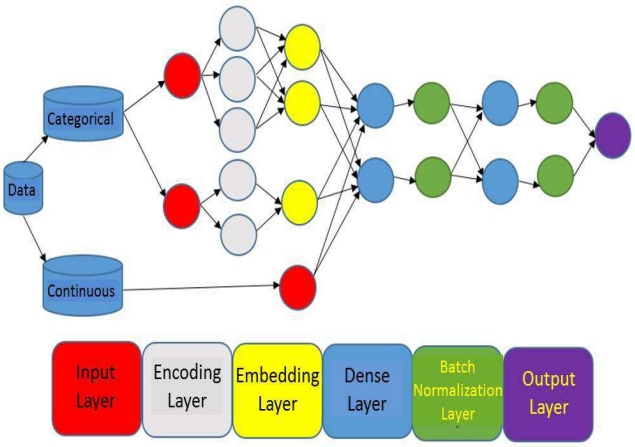

FIGURE 1. Architecture of NN model used for entity embedding.

The architecture of the entity embedding model used in the present work is shown in Figure 1. The entire system comprises of an input layer, encoding layer, embedding layer, dense layer, batch normalization layer and an output layer. The input layer comprises of both categorical and continuous variables. While continuous variables are connected directly to the dense layer, categorical variables pass through the transformation as discussed. Outputs of embedding layer are later connected to the dense layer where they are again grouped with the continuous data. Further, neurons between dense layers and batch normalization layers are connected unidirectionally i.e. in one-to-one manner whereas neurons in batch normalization layer and dense layer are connected completely similar to the batch normalization and the output layer.

Linear activation functions are used for entity embedding layer and the number of neurons in this layer are obtained based on grid search algorithm [31].

IV. NEUROEVOLUTION USING NSGAIII ALGORITHM

In the face of multiple objectives, the usual concept of minimization and maximization is replaced by obtaining a set of trade-off solutions which is also called as a Pareto optimal solution set. While using evolutionary algorithms for this purpose, a population of solutions (genotypes) is randomly generated and evolved using genetic operators such as selection, crossover, mutation etc. A population thus evolved in the next iteration of the algorithm is called an offspring. The competing solutions are compared using their individual

fitness indices which are evaluated based on their objective function values using non-domination criterion. This further means that the solutions, which have objective function values not dominated by other competing solutions, are selected. As mentioned in the previous Section, NSGA-III is an efficient extension to the former NSGA-II algorithm to address many objective optimization problems. The generic phases in evolving deep learning networks using NSGA-III are explained below [34]:

1. Initialize a population of N deep learning genotypes, each of which is encoded with chosen hyper-parameters. This is called the parent population (P_i). An offspring population (Q_i), is obtained as a result of the application of genetic operators such as cross over and mutation on (P_i).
2. Next the two populations are combined together to form a pool of $2N$ solutions ($P_i \cup Q_i$). The combined population of NN is evaluated on the training data and the three performance metrics are evaluated for each of the NN model. Later, a fitness index (non-domination index) F based on performance metrics is determined for every member of the population [56].
3. In order to select optimal N individuals from the combined population $R = P_i \cup Q_i$ (having $2N$ individuals) the population R is sorted according to their non-domination levels ($F_1, F_2, F_3 etc..$).
4. Thereafter, one individual is selected from each non-domination level to form a new population S_i, starting from F_1, until the size of S_i is equal to or bigger than N.
5. The offspring population S_i thus obtained is again combined with the preceding parent population and steps 2-5 are repeated for given number of generations (which is another hyper-parameter chosen in the beginning of algorithm) till the networks satisfy the set performance/termination.

One of the main advantages of using evolutionary algorithms for optimization of DNNs is that evolution can be combined with learning to provide a powerful synergy. It is known that, gradient-based learning algorithms are normally sensitive to the set of initial hyper-parameter values which affects the network performance considerably by providing sub-optimal values of parameters. On the contrary, evolutionary algorithms can be used to find suitable hyper-parameter values of networks. However, during every epoch of evolutionary algorithm of DNN, the parameters such as weights and biases of the network are obtained using conventional backpropagation scheme.

Preprocessing of the SEER database to select input variables for DNN and experimental design is explained in the subsequent sections before NSGA-III implementation.

V. EXPERIMENTS

A. DATA PREPROCESSING

In order to increase efficiency and accuracy of prognosis prediction from a classifier, the relevant database should be processed carefully. Data preprocessing essentially includes understanding data type and its distribution, applying suitable transformation, handling skewed and missing data, analyzing outliers, and reducing the dimensions. Available data is processed through several stages for dimensionality reduction without sacrificing its important features.

Prior to start working with the SEER database, SEER Stat software is used to access breast cancer data from 2004 to 2014. Later, number of variables are reduced iteratively referring to SEER record description documentation [57]. As a result, a set of 20 variables is obtained in text format for around 659802 cases. In the final dataset, 19 amongst 20 variables are considered as independent variables, whereas, ''survivability'' is taken as the dependent variable.

Further, out of 19 independent variables selected, 18 were categorical variables and the variable for the tumor size was the only continuous variable. The dependent variable, survival month, was transformed into binary format whereby values greater than sixty become one and values less than sixty were assigned to zero. Details of selected variables are provided in the Table 3.

It is important to mention here that the incidences of binary classified dependent variable are found to be almost balanced with 43% for negative and 57% for positive classes. All the input variables including the continuous and categorical ones are converted into binary numbers. Total input binary variables are found to be 89 after this conversion.

B. EXPERIMENTAL DESIGN

In order to perform experiments the data is randomly divided into train and test in a proportion of 90% and 10% respectively. Next, a ten-fold cross validation is performed on the train sample which is normally a practice adopted while working with cancer data [58]. During each epoch, nine subsets are selected to enter training whereas the remaining one is used later for validation purpose. Further, a generalization capability of model was evaluated on test sample. To prevent overfitting of classifiers, L2 regularization is also implemented. Finally, the DNN model is trained in a TensorFlow environment, which is an open source library used for machine learning applications. The simulations are implemented on a workstation with such configurations: Intel Xeon Platinum 2.10 GHz 48 Cores, NVIDIA Quadro P6000 24 GB, 512 GB DDR4.

C. PERFORMANCE METRICS

Enhancing performance by reducing network error may not be sufficient and robust rather it may results in over-fitting the training data. In order to test robustness of the prognosis prediction five important performance metrics have been chosen. The chosen metrics are accuracy, sensitivity, specificity, Area under the ROC curve (AUROC) and F1 score.

Sensitivity measures the ability of a test to detect the condition when the condition is present. On the other hand, specificity measures the ability of a test to correctly exclude

TABLE 3. Description of selected variables.

#	Variable name	Description	Data type
1	Age	Actual age of patient at the time of diagnosis	Factor
2	Race	White, black and others	Factor
3	Year of birth	Year of birth	Factor
4	Marital status	Marital status	Factor
5	State	State	Factor
6	Year of diagnosis	The year tumor was first diagnosed	Factor
7	Behavior code	In situ or malignant	Factor
8	Primary site	This data item identifies the site in which the primary tumor originated.	Factor
9	Histologic type	The data item Histologic Type describes the microscopic composition of cells and/or tissue for a specific primary.	Factor
10	Grade	Grade	Factor
11	Laterality	Laterality describes the side of a paired organ or side of the body on which the reportable tumor originated.	Factor
12	Diagnostic confirmation	This data item records the best method used to confirm the presence of the cancer being reported.	Factor
13	Reason no surgery	Reason no surgery	Factor
14	Tumor size	Tumor size	Numeric
15	Extension	Extension	Factor
16	Lymph nodes	Lymph nodes	Factor
17	Metastasis	Metastasis	Factor
18	Cause of death	Cause of death	Factor
19	Survival moth	Survival moth	Factor

the condition when the condition is absent. Usually high sensitivity tests have low specificity, which further means that these two objectives are conflicting. Further, precision is defined as the ratio of correctly predicted positive observations to the total predicted positive observations and recall reveals how many of positive observations were actually found. The weighted average of Precision and Recall is termed as F1 Score. However, to begin with, a confusion matrix is derived which is used to test the correctness and accuracy of a subject model. The metrics used in this research are defined below using common abbreviations such as TN (True Negatives) TP (True Positives) FN (False Negatives) and FP (False Positives).

$$\text{Accuracy} = (\text{TN} + \text{TP})/(\text{TP} + \text{FP} + \text{FN} + \text{TN})$$

$$\text{Sensitivity (Recall)} = \text{TP}/(\text{TP} + \text{TN})$$

$$\text{Specificity} = \text{TN}/(\text{TN} + \text{FP})$$

$$\text{F1 score} = (2 * \text{Precision} * \text{Recall})/(\text{Precision} + \text{Recall})$$

Area under the ROC curve (AUROC) measures an area under the receiver characteristic curve, which plots sensitivity against one minus specificity. In addition, the terms AUROC and AUC are used interchangeably in a literature [49].

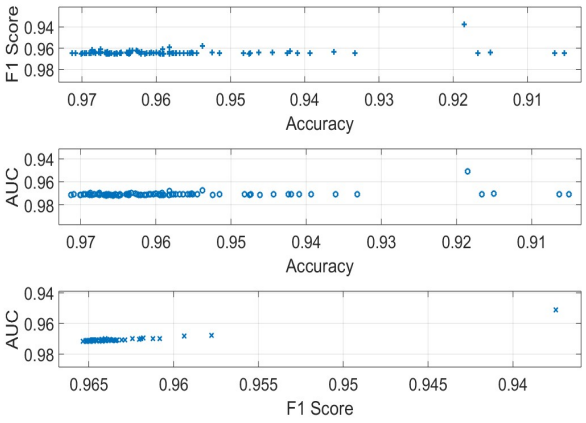

FIGURE 2. Performance metrics of the final Pareto optimal set of 100 DNNs.

D. NSGAIII IMPLEMENTATION

While implementing NSGAIII one thousand DNN models were initialized with varied hyper parameters. The ranges and types of hyper parameters used for the initial population are given below:

Number of neurons in hidden layers	1-1000
Number of layers	1-10
Learning rate	0.001-1
Types of activation functions used for the hidden layers:	Relu, Sigmoid, Linear, Elu, Selu

Initially, population of thousand solutions for DNN is randomly generated with four hyper parameters defined as variables with in their limiting ranges. Vital parameters of DNNs such as connection weights (initialized randomly), momentum index is kept constant as 0.9 and sigmoidal gain and threshold values are initialized as unity.

Activation function for the input layer of initialized models is chosen to be linear. Since the prediction of survivability is a binary classification problem, we have chosen sigmoidal activation function for the output layer.

During all these experiments, following simulation parameters were considered for NSGA III [37], [52].

Population size: 1000; Crossover prob.: 0.90; Real-parameter mutation prob.:0.1; Distribution index for crossover: 10; Distribution index for mutation: 50; Number of iterations: 20

Subsequent to the implementation of NSGAIII (as discussed in Section V) we obtain a Pareto optimal set of DNN models which are all non-dominated. Three performance metrics (Accuracy, AUC and F1 score) of the first front solutions of DNNs are plotted against each other and shown in Figure 2. While all the models on the first non-dominated front are equally good, we just need a singular DNN model for prognosis prediction. Selection of a single DNN model from the set of equally good DNNs is difficult and calls for some strategy from the realm of decision making

theories or processes. We propose a fuzzy logic based scheme in the next Section which can help in selecting the best DNN model from the better ones.

VI. SELECTION OF THE BEST DEEP LEARNING MODEL FROM PARETO OPTIMAL SET

For practical implementation, we need a singular best solution for DNN model or in other words a single Pareto point giving best compromised DNN model. This results in a big cognitive burden on the user and as such certain strategy is required to help the user in making this vital decision. Previously, some research has been proposed around this using min-max approach or fuzzy inference [52], [53]. In the present research, we propose a fuzzy inference based method which can provide us a single metric combining many performance metrics in a logical manner. Various steps used during implementation of this method are described below.

A. FUZZIFICATION

To begin with, the performance metrics are defined as fuzzy variables and their expected ranges are decided by the values obtained from the NSGAIII experiments (Section VI). The ranges for these fuzzy performance metrics are later normalized between 0 and 1. The metrics are defined as fuzzy variables using three fuzzy activation functions (AFs) namely; Low (L), Medium (M) and High (H). The shapes of AFs are chosen to be Gaussian (1-3). The typical practice in the design of fuzzy systems is to use triangular or trapezoidal activation functions, however, owing to the smooth transition between activation functions, a Gaussian distribution is selected in the present work. This step can be termed as *fuzzification* of the inputs with reference to the fuzzy selection systems.

$$L = ae^{-\left(\frac{f_i}{\sigma_i}\right)} \tag{1}$$

$$M = ae^{-\left(\frac{f_i-\frac{R_i}{2}}{\sigma_i}\right)} \tag{2}$$

$$H = ae^{-\left(\frac{f_i-R_i}{\sigma_i}\right)} \tag{3}$$

Here, f_i stands for the input performance metrics values from a competing DNN model, which are considered as objectives to optimize. The range of the performance metrics is R_i and is defined as $R_i = (\max(f_i) - \min(f_i))$. The constant $a = \frac{1}{\sigma\sqrt{2\pi}}$ and standard deviation of three fuzzy activation functions (L, M and H) is taken as $\sigma_i = R_i/5$ which is also constant.

B. FUZZY INFERENCE

Once the inputs (performance metrics) are defined as fuzzy variables, an inference mechanism is required in place to complete the design of this fuzzy system. Inferencing in fuzzy systems is realized through its rule-base. The rule-base essentially is a collection of *if*and *then* statements which maps the *antecedents or inputs* to the *consequents or the outputs*. A common structure of fuzzy rule-base is given below.

If f_1 *is* L *and*, ... *and* f_N *is* H *then* AS_i *is* y_i

Here $f_1 \ldots f_N$ are the performance metrics as inputs to the fuzzy system, AS_i is the Activation Score for i^{th} rule and its numerical value is y_i. Total number of rules can be derived from the number of AFs used for defining antecedent variables. For simplicity, we have considered only three (Accuracy, AUC and F1 score) out of five performance metrics while designing this fuzzy based system. Since each of the three performance metrics is defined using three AFs (L, M and H) the total number of fuzzy rules (N_r) shall be 3^3 i.e. 27. These rules have all possible combinations of AFs of the antecedents. The AFs are further assigned numerical values such that $L = 0$; $M = 1$; $H = 2$. The outputs or the consequents for all these rules are simply the sum of the activation scores of their component AFs as shown in (4).

$$y_i = 1 + \sum_{j=1}^{5} AS_{ij} \tag{4}$$

Here i represents the rule index, and AS_{ij} is the activation score for j^{th} objective function in i^{th} rule. For instance, if in a particular rule all the objectives have low (L) AFs then the output of the rule or the activation score of that rule shall be one else if all the AFs are medium (M), the output or the score shall be 6.

Later, we follow the conventional procedure of fuzzy inferencing to calculate output from fuzzy system. As per the practice, output for each rule is computed by considering the degrees of fulfillment of all the AFs for given set of input performance metrics. To compute degrees of fulfillment equations (1-3) are used by plugging input (f_i) values. Weights for individual rules are calculated using (5).

$$w_i = \prod_{j=1}^{5} (L_{ij} * M_{ij} * H_{ij}) \tag{5}$$

The overall activation score of a candidate DNN is essentially a numerical or crisp output of the fuzzy inference system. This output is the weighted average of all the individual rule consequents for a given set of input values. Therefore, the final overall activation score (OAS) can be computed using (6) as below.

$$Y = \frac{\sum_{i=1}^{N_r} (w_i y_i)}{\sum_{i=1}^{N_r} w_i} \tag{6}$$

Here N_r stands for number of fuzzy rules which are 27 in the present system. Performance metrics from all the non-dominated solutions for DNNs (obtained through NSGAIII) are given as inputs to the above described fuzzy selection system and the outputs obtained subsequently are recorded. Ten representative DNN Solutions from Pareto optimal DNNs with their performance metrics and their overall activation scores have been shown in Table 4, along with their respective input performance metric values and final overall activation scores. Apparently, a candidate design number 48 is finally selected for the proposed DNN design owing to its maximum fuzzy index value (6.732). It may be emphasized here that though all the solutions are non-dominated and should be equally good, their fuzzy indices or OAS are different. Further, interestingly the proposed fuzzy ranking and selection

TABLE 4. Ten instances from the Pareto optimal DNNs with their performance metrics and their overall activation scores.

DNN Solution Numbers	Test Accuracy	Test F1 Score	Test AUC	Overall Activation Score (OAS)
45	0.96230	0.96419	0.97143	6.7213
46	0.95915	0.96495	0.97151	6.7183
47	0.95896	0.96500	0.97128	6.7179
48	**0.97130**	**0.96489**	**0.97211**	**6.7320**
49	0.96155	0.96508	0.97150	6.7212
50	0.96474	0.96398	0.97102	6.7235
51	0.93932	0.96483	0.97105	6.6917
52	0.91501	0.96401	0.97010	6.6513
53	0.96207	0.96495	0.97112	6.7213
54	0.95451	0.96476	0.97086	6.7118

TABLE 5. Parameters of the finally selected DNN Model.

Activation function for input layer	Linear
Number of neurons in input layer	89
Number of hidden layers	4
Number of neurons in hidden layers	8
Activation function for hidden layer	RELU
Learning rate	0.3425
Regularization method	L2
Weights initialization method	Uniform
Mini-batch size	128
Training epoch	20
Normalization	Batch
Activation function for output layer	Sigmoid

method is able to provide better discrimination among candidate solutions. Eventually, a solution with maximum fuzzy index is found to be better than rest of the DNN models.

Parameters of the finally selected DNN Model, amongst Pareto optimal models, are displayed in Table 5.

VII. EXPERIMENTS WITH OTHER CLASSIFIERS

Subsequent to obtaining hyper-parameters for the best DNN model using NSGA III, its performance is compared with other classifiers. In the present work we have implemented widely used classification algorithms such as Logistic Regression, Support Vector Machines, Random Forest and Gradient Boosting. Various parameters used for these classifiers are given in Table 6. The breast cancer data is sampled into train and test in a ratio of 90% to 10%. Further, a ten-fold cross validation is performed on train sample to find optimal model parameters.

All the classifiers are regularized in order to avoid overfitting. Identical test samples are used to measure generalization capabilities of classifiers. Logistic Regression, SVM, Random Forest were implemented using scikit-learn package and Gradient Boosting was developed using xgboost package.

VIII. RESULTS AND DISCUSSIONS

After successful optimization of DNNs using NSGAIII, a Pareto optimal set of DNNs is obtained. Some of the representative DNN solutions with their performance metrics are given in Table 4. The fuzzy inference based method devised to select the best DNN solution from the Pareto optimal solutions has been successful. As can be seen from Table 4, 48th DNN solution has highest fuzzy overall activation score and is considered to be the best among all. Glancing at the values displayed in Table 4, the selected solution (48th) fares well in accuracy & AUC and has comparable

TABLE 6. Parameters of other classifiers.

Coefficient of regularization	0.8
Coefficient of regularization	0.8
Number of trees	200
Maximum depth of tree	10
Minimum number of samples to split an internal node	02
Coefficient of regularization	0.4
Maximum depth of tree	05
Minimum sum of instance weight needed in a child	04
Subsample ratio of the training instances	0.8
Subsample ratio of columns when constructing each tree	0.3

TABLE 7. Comparison of classifiers with entity embedding.

		DNN	Logistic Reg.	SVM	Random Forest	Gradient Boosting
Accuracy (Acc.)	Train	97.13	97.01	97.04	97.13	97.10
	Test	97.09	96.99	97.02	97.07	97.08
Sensitivity (Sn)	Train	0.9669	0.9718	0.9675	0.9790	0.9769
	Test	0.9663	0.9719	0.9677	0.9787	0.9771
Specificity (Sp)	Train	0.9703	0.9689	0.9725	0.9656	0.9667
	Test	0.9689	0.9685	0.9720	0.9647	0.9661
F1 Score	Train	0.9649	0.9615	0.9653	0.9667	0.9663
	Test	0.9646	0.9651	0.9652	0.9662	0.9662
AUC	Train	97.21	97.04	97.00	97.23	97.18
	Test	97.15	97.02	96.98	97.17	97.16

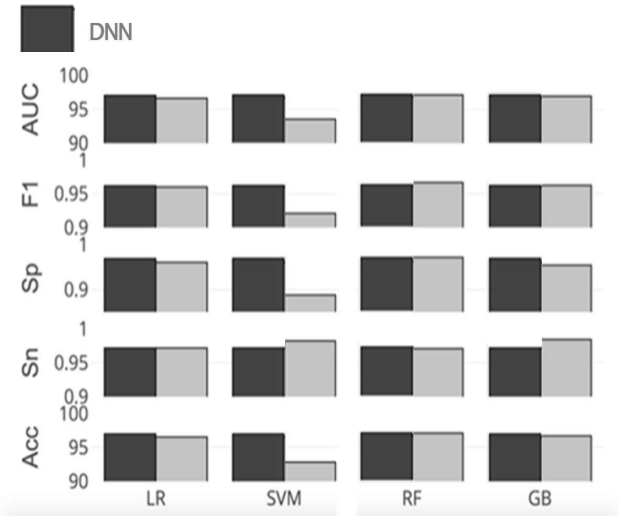

FIGURE 3. Comparison of classifiers with DNN for train samples.

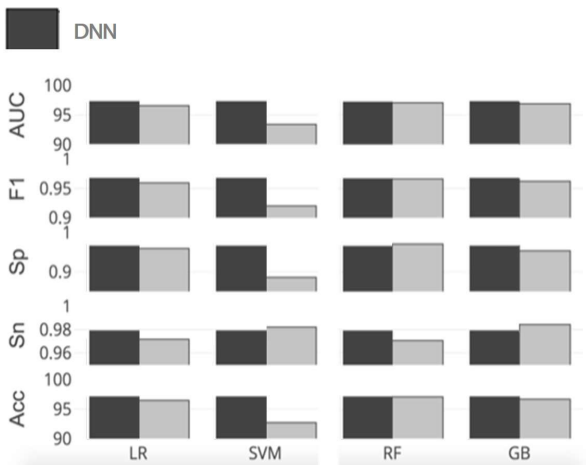

FIGURE 4. Comparison of classifiers with DNN for test samples.

F1 score. The set of hyper-parameters for the finally selected DNN is provided in Table 5. Further analysis showed that the OAS is sensitive to the activation function used at the hidden layers. RELU and Sigmoid functions provided better OAS values compared to other functions. Similarly, better OAS can be obtained with less number of neurons in the hidden layers and number of layers should not be more than four in the present case. Other classifiers such as Logistic Regression, Support Vector Machines etc. are also implemented and trained using categorical variables along with continuous variables from SEER database. These classifiers are compared with the finally selected DNN Model using five performance metrics. Results from the train and test experiments while using classifiers and final DNN model are displayed in Table 7. The resulting performance metrics are also illustrated in Figures 3&4 for train and test experiments.

IX. CONCLUSION AND FUTURE WORK

Breast cancer prognostic modelling requires synthesis of large SEER database which has many discrete features apart from continuous ones. In this research, we have proposed a neural network based entity embedding approach to obtain continuous vector representations of categorical variables. Later, these transformed categorical variables are used along with other continuous variables for prognostic modelling of breast cancer data. In order to achieve enhanced accuracy as well as interpretability we have proposed a neuroevolution approach whereby NSGA III is used to optimize and provide hyper parameters for DNNs. As a result of this optimization a set of Pareto optimal DNN models is obtained which gives us further insight in the behavior of DNNs. A novel method of selecting final DNN model from the set of Pareto optimal solutions is also demonstrated successfully in this manuscript. Intuitively, looking at the results one can find a relation between hyper parameters (building blocks of DNN) and various performance metrics. It is possible therefore to achieve a specific performance metric by modifying DNN hyper parameters. Further analysis of the Pareto optimal DNN solutions in the light of their hyper parameters and performance metrics can provide more information about the behavior of DNN. Increased transparency and interpretability of DNN models may help in performing training experiments more efficiently and effectively. Similarly, enhanced transparency in DNN models will give rise to its acceptability among medical practitioners as well.

During the experiments it was found that the evolutionary algorithm (NSGAIII) used in the present research may not handle many performance criteria at the same time. At many instances during experiments the algorithm converged prematurely providing a false or pseudo Pareto optimal front. In future, authors would like to modify the selection operator of NSGAIII in order to address this issue. Further, analogies shall be established between hyper-parameters and model performance metrics as a future direction of this research.

REFERENCES

[1] C. Falato *et al.*, "Prognosis in patients diagnosed with loco-regional failure of breast cancer: 34 years longitudinal data from the Stockholm–Gotland cancer registry," *Breast Cancer Res. Treatment*, vol. 172, pp. 703–712, Dec. 2018.

[2] E. Hortal, D. Planelles, F. Resquin, J. M. Climent, J. M. Azorín, and J. L. Pons, "Using a brain-machine interface to control a hybrid upper limb exoskeleton during rehabilitation of patients with neurological conditions," *J. NeuroEng. Rehabil.*, vol. 12, p. 92, Oct. 2015.

[3] J. Ferlay *et al.*, "Cancer incidence and mortality worldwide: Sources, methods and major patterns in GLOBOCAN 2012," *Int. J. Cancer*, vol. 136, no. 5, pp. E359–E386, 2015.

[4] D. Laudisio, G. Muscogiuri, L. Barrea, S. Savastano, and A. Colao, "Obesity and breast cancer in premenopausal women: Current evidence and future perspectives," *Eur. J. Obstetrics Gynecol. Reproductive Biol.*, vol. 230, pp. 217–221, Nov. 2018.

[5] A. Borah and B. Nath, "Identifying risk factors for adverse diseases using dynamic rare association rule mining," *Expert Syst. Appl.*, vol. 113, pp. 233–263, Dec. 2018.

[6] S. M. Telloni, "Tumor staging and grading: A primer," in *Methods In Molecular Biology*, vol. 1606. Clifton, NJ, USA: Springer, 2017, pp. 1–17.

[7] M. Vahdaninia, S. Omidvari, and A. Montazeri, "What do predict anxiety and depression in breast cancer patients? A follow-up study," *Social Psychiatry Psychiatric Epidemiol.*, vol. 45, no. 3, pp. 355–361, 2010.

[8] S. L. Kozachik and K. Bandeen-Roche, "Predictors of patterns of pain, fatigue and insomnia during the first year following a cancer diagnosis in the elderly," *Cancer Nursing*, vol. 31, no. 5, pp. 334–344, 2008.

[9] D. Kalaitzopoulos, "The potential of precision medicine," *New Horizons Transl. Med.*, vol. 3, pp. 63–65, Mar. 2016.

[10] J. Yoon, C. Davtyan, and M. van der Schaar, "Discovery and clinical decision support for personalized healthcare," *IEEE J. Biomed. Health Informat.*, vol. 21, no. 4, pp. 1133–1145, Jul. 2017.

[11] C. Liangjun, P. Honeine, Q. Hua, Z. Jihong, and S. Xia, "Correntropy-based robust multilayer extreme learning machines," *Pattern Recognit.*, vol. 84, pp. 357–370, Dec. 2018.

[12] S. M. Beecher, D. P. O'Leary, R. McLaughlin, K. J. Sweeney, and M. J. Kerin, "Influence of complications following immediate breast reconstruction on breast cancer recurrence rates," *Brit. J. Surgery*, vol. 103, pp. 391–398, Mar. 2016.

[13] L. Dora, S. Agrawal, R. Panda, and A. Abraham, "Optimal breast cancer classification using Gauss–Newton representation based algorithm," *Expert Syst. Appl.*, vol. 85, pp. 134–145, Nov. 2017.

[14] P. K. Jamwal, B. Abdikenov, and S. Hussain, "Evolutionary optimization using equitable fuzzy sorting genetic algorithm (EFSGA)," *IEEE Access*, vol. 7, pp. 8111–8126, 2019. doi: 10.1109/ACCESS.2018.2890274.

[15] B. R. A. Cirkovic, A. M. Cvetkovic, S. M. Ninkovic, and N. D. Filipovic, "Prediction models for estimation of survival rate and relapse for breast cancer patients," in *Proc. IEEE 15th Int. Conf. Bioinf. Bioeng. (BIBE)*, Nov. 2015, pp. 1–6.

[16] I. N. Yulita, M. I. Fanany, and A. M. Arymurthy, "Fast convolutional method for automatic sleep stage classification," *Healthcare Informat. Res.*, vol. 24, no. 3, pp. 170–178, 2018.

[17] F. Gao *et al.*, "SD-CNN: A shallow-deep CNN for improved breast cancer diagnosis," *Comput. Med. Imag. Graph.*, vol. 70, pp. 53–62, Dec. 2018.

[18] J. Liu *et al.*, "An end-to-end deep learning histochemical scoring system for breast cancer TMA," *IEEE Trans. Med. Imag.*, vol. 38, no. 2, pp. 617–628, Feb. 2019.

[19] J. Ahmad, K. Muhammad, J. Lloret, and S. W. Baik, "Efficient conversion of deep features to compact binary codes using Fourier decomposition for multimedia big data," *IEEE Trans. Ind. Informat.*, vol. 14, no. 7, pp. 3205–3215, Jul. 2018.

[20] N. Agarwal, V. N. Balasubramanian, and C. V. Jawahar, "Improving multiclass classification by deep networks using DAGSVM and Triplet Loss," *Pattern Recognit. Lett.*, vol. 112, pp. 184–190, Sep. 2018.

[21] J. Gu *et al.*, "Recent advances in convolutional neural networks," *Pattern Recognit.*, vol. 77, pp. 354–377, May 2018.

[22] S. Purushotham, C. Meng, Z. Che, and Y. Liu, "Benchmarking deep learning models on large healthcare datasets," *J. Biomed. Informat.*, vol. 83, pp. 112–134, Jul. 2018.

[23] P. Cerda, G. Varoquaux, and B. Kégl, "Similarity encoding for learning with dirty categorical variables," *Mach. Learn.*, vol. 107, pp. 1477–1494, Sep. 2018.

[24] P. Frasconi, M. Gori, and A. Sperduti, "A general framework for adaptive processing of data structures," *IEEE Trans. Neural Netw.*, vol. 9, no. 5, pp. 768–786, Sep. 1998.

[25] A. Paccanaro and G. E. Hinton, "Learning distributed representations of concepts using linear relational embedding," *IEEE Trans. Knowl. Data Eng.*, vol. 13, no. 2, pp. 232–244, Mar. 2001.

[26] Y. Kim, Y. Jernite, D. Sontag, and A. M. Rush, "Character-aware neural language models," in *Proc. 30th AAAI Conf. Artif. Intell. (AAAI)*, 2016, pp. 2741–2749.

[27] R. Sennrich, B. Haddow, and A. Birch, "Neural machine translation of rare words with subword units," in *Proc. 54th Annu. Meeting Assoc. Comput. Linguistics (ACL)*, 2016, pp. 1715–1725.

[28] S. Zhang, X. Zhang, and J. Chan, "A word-character convolutional neural network for language-agnostic Twitter sentiment analysis," in *Proc. ACM Int. Conf. 22nd Australas. Document Comput. Symp.*, 2017, p. 12.

[29] Y. Wang *et al.*, "A comparison of word embeddings for the biomedical natural language processing," *J. Biomed. Informat.*, vol. 87, pp. 12–20, Nov. 2018.

[30] P. Bojanowski, E. Grave, A. Joulin, and T. Mikolov, "Enriching word vectors with subword information," *Trans. Assoc. Comput. Linguistics*, vol. 5, pp. 135–146, Dec. 2017.

[31] H. Li and Q. Zhang, "Multiobjective optimization problems with complicated Pareto sets, MOEA/D and NSGA-II," *IEEE Trans. Evol. Comput.*, vol. 13, no. 2, pp. 284–302, Apr. 2009.

[32] Q. Zhang and H. Li, "MOEA/D: A multiobjective evolutionary algorithm based on decomposition," *IEEE Trans. Evol. Comput.*, vol. 11, no. 6, pp. 712–731, Dec. 2007.

[33] J. Bader and E. Zitzler, "HypE: An algorithm for fast hypervolume-based many-objective optimizationn," *Evol. Comput.*, vol. 19, no. 1, pp. 45–76, Mar. 2008.

[34] J.-H. Yi, S. Deb, J. Dong, A. H. Alavi, and G.-G. Wang, "An improved NSGA-III algorithm with adaptive mutation operator for big data optimization problems," *Future Gener. Comput. Syst.*, vol. 88, pp. 571–585, Nov. 2018.

[35] T. Praczyk, "Cooperative co-evolutionary neural networks," *J. Intell. Fuzzy Syst.*, vol. 30, no. 5, pp. 2843–2858, 2016.

[36] B. Shabash and K. C. Wiese, "EvoNN: A customizable evolutionary neural network with heterogenous activation functions," in *Proc. Genetic Evol. Comput. Conf. Companion (GECCO)*, 2018, pp. 1449–1456.

[37] K. Deb and H. Jain, "An evolutionary many-objective optimization algorithm using reference-point-based nondominated sorting approach, part I: Solving problems with box constraints," *IEEE Trans. Evol. Comput.*, vol. 18, no. 4, pp. 577–601, Apr. 2013.

[38] J. M. Kontoleon, "Optimum link allocation of fixed topology networks," *IEEE Trans. Rel.*, vol. R-28, no. 2, pp. 145–147, Jun. 1979.

[39] X. Yao, "Evolving artificial neural networks," *Proc. IEEE*, vol. 87, no. 9, pp. 1423–1447, Sep. 1999.

[40] K. O. Stanley and R. Miikkulainen, "Evolving neural networks through augmenting topologies," *Evol. Comput.*, vol. 10, no. 2, pp. 99–127, 2002.

[41] J. E. Auerbach and J. C. Bongard, "Evolving complete robots with CPPN-NEAT: The utility of recurrent connections," in *Proc. Genetic Evol. Comput. Conf. (GECCO)*, 2011, pp. 1475–1482.

[42] A. Cully and J.-B. Mouret, "Evolving a behavioral repertoire for a walking robot," *Evol. Comput.*, vol. 24, no. 1, pp. 59–88, 2016.

[43] X. Yao and Y. Liu, "A new evolutionary system for evolving artificial neural networks," *IEEE Trans. Neural Netw.*, vol. 8, no. 3, pp. 694–713, May 1997.

[44] P. C. Pendharkar, "An empirical study of design and testing of hybrid evolutionary–neural approach for classification," *Omega*, vol. 29, pp. 361–374, Aug. 2001.

[45] K. Senhaji, H. Ramchoun, and M. Ettaouil, "Multilayer perceptron: NSGA II for a new multi-objective learning method for training and model complexity," *Adv. Intell. Syst. Comput.*, vol. 756, pp. 154–167, May 2018.

[46] R. Denysiuk, A. Gaspar-Cunha, and A. C. B. Delbem, "Neuroevolution for solving multiobjective knapsack problems," *Expert Syst. Appl.*, vol. 116, pp. 65–77, Feb. 2019.

[47] Z. Chen, C. K. Yeo, B. S. Lee, C. T. Lau, and Y. Jin, "Evolutionary multiobjective optimization based ensemble autoencoders for image outlier detection," *Neurocomputing*, vol. 309, pp. 192–200, Oct. 2018.

[48] A. M. Hernández-Díaz, A. Bueno-Crespo, J. Pérez-Aracil, and J. M. Cecilia, "Multi-objective optimal design of submerged arches using extreme learning machine and evolutionary algorithms," *Appl. Soft Comput.*, vol. 71, pp. 826–834, Oct. 2018.

[49] K. Mason, J. Duggan, and E. Howley, "A multi-objective neural network trained with differential evolution for dynamic economic emission dispatch," *Int. J. Elect. Power Energy Syst.*, vol. 100, pp. 201–221, Sep. 2018.

[50] P. Melin and D. Sánchez, "Multi-objective optimization for modular granular neural networks applied to pattern recognition," *Inf. Sci.*, vols. 460–461, pp. 594–610, Sep. 2018.

[51] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan, "A fast and elitist multiobjective genetic algorithm: NSGA-II," *IEEE Trans. Evol. Comput.*, vol. 6, no. 2, pp. 182–197, Apr. 2002.

[52] H. Jain and K. Deb, "An evolutionary many-objective optimization algorithm using reference-point based nondominated sorting approach, Part II: Handling constraints and extending to an adaptive approach," *IEEE Trans. Evol. Comput.*, vol. 18, no. 4, pp. 602–622, Aug. 2014.

[53] F. Cao *et al.*, "An algorithm for clustering categorical data with set-valued features," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 29, no. 10, pp. 4593–4606, Oct. 2018.

[54] C. Guo and F. Berkhahn. (2016). "Entity embeddings of categorical variables." [Online]. Available: https://arxiv.org/abs/1604.06737

[55] Y. Cai, Y. Wang, H. Xu, S. Sun, C. Wang, and L. Sun, "Research on rotor position model for switched reluctance motor using neural network," *IEEE/ASME Trans. Mechatronics*, vol. 23, no. 6, pp. 2762–2773, Dec. 2018.

[56] K. Deb, *Optimization for Engineering Design: Algorithms and Examples*. New Delhi, India: Prentice-Hall, 2005.

[57] M.-T. Chen *et al.*, "Comparison of patterns and prognosis among distant metastatic breast cancer patients by age groups: A SEER population-based analysis," *Sci. Rep.*, vol. 7, Aug. 2017, Art. no. 9254.

[58] K.-H. Yu *et al.*, "Predicting non-small cell lung cancer prognosis by fully automated microscopic pathology image features," *Nature Commun.*, vol. 7, Aug. 2016, Art. no. 12474.

BEIBIT ABDIKENOV received the bachelor's degree in economics from Suleiman Demirel University, Kazakhstan, in 2007, and the master's degree in information technology from Eurasian National University, Kazakhstan, in 2014. He is currently pursuing the Ph.D. from Nazarbayev University, Astana, Kazakhstan.

ZANGIR IKLASSOV received the bachelor's degree in economics from Nazarbayev University, Astana, Kazakhstan, where he is currently pursuing the M.Sc. degree. He has experience in machine learning and data analysis, and he is currently a Research Assistant with Nazarbayev University.

ASKHAT SHARIPOV received the bachelor's degree in electrical and computer engineering from Nazarbayev University, Astana, Kazakhstan, where he is currently a Research Assistant.

SHAHID HUSSAIN received the B.Sc. degree (Hons.) in mechatronics and control engineering from the University of Engineering and Technology at Lahore, Lahore, Pakistan, in 2007, and the M.E. and Ph.D. degrees in mechanical engineering from The University of Auckland, Auckland, New Zealand, in 2009 and 2013, respectively. He is currently an Assistant Professor with The University of Canberra, Canberra, Australia. His research interests include robot-assisted rehabilitation, compliant actuation, the optimization of robots, human–robot interaction, the biomechanical modeling of musculoskeletal systems, and the nonlinear control of dynamic systems.

PRASHANT K. JAMWAL (M'15) received the master's degree from IIT Roorkee, India, and the Ph.D. degree from The University of Auckland, Auckland, New Zealand. He is currently a Faculty Member of Nazarbayev University, Astana, Kazakhstan. His research interests include artificial intelligence, fuzzy mathematics and its applications, smart sensors and actuators, biomechatronics, biomedical robotics, evolutionary algorithms, and multi-objective optimization. He has over 20 years of teaching and research experience in mechatronics, medical robotics, and advanced manufacturing technologies. He is currently an Associate Editor of the *International Journal of Biomechatronics and Robotics*, and he is acting as a Reviewer of many international journals and conferences.