PERFORMANCE OPTIMIZATION OF NEUROEVOLUTION FOR IMPROVED PROGNOSIS OF THE BREAST CANCER

Beibit Abdikenov

A thesis submitted in partial fulfilment of the requirement of
Nazarbayev University
for the degree of Doctor of Philosophy

May 2020

Declaration

I declare that the research contained in this thesis, unless otherwise formally indicated within the text, is the original work of the author. The thesis has not been previously submitted to this or any other university for a degree, and does not incorporate any material already submitted for a degree.

Signed: Beibit Abdikenov

Dated: 23/05/2020

Abstract

Cancer is the second largest cause of mortality, responsible for one in every six deaths globally. Cancer has a significant socio-economic impact and its global cost is estimated to be close to $150 billion. Breast cancer is the most common female cancer and its high incidence places it among Kazakhstan's most challenging public health problems. Advances in computing and sensing technologies and increased storage availability means that vast quantities of data are now available. While the data is sure to help practitioners understand what causes breast cancer and the best treatment approaches, the number of oncologists understanding its use is limited. Accurate and reliable prognoses are increasingly difficult because of the enormous amounts of data about breast cancer and the low survival rates. The available data's heterogeneity adds to the challenges for data analytics posed by sheer data volume. Moreover, categorical variables in the heterogeneous dataset require accurate pre-processing if enhanced interpretation is to make progress towards prognosis possible. An advanced research in estimating the missing values in databases is also introduced in this thesis work.

Rigorous research efforts have brought about the development of a novel entity embedding scheme based on neural networks capable of addressing effectively the encoding of categorical variables with high cardinality during the presented research. Employing our proposed scheme, it is now possible to represent the categorical variables as real values in high-dimensional space capable of greatly improved interpretation.

Neuroevolution, which is a Meta heuristic approach, has been suggested through our work as a robust way of modelling prognosis from the breast cancer database. Neuroevolution also results in multiple equitable solutions of DNNs (Deep Neural Networks) thereby providing

users with many options to choose from. Neuroevolution performance has been optimized using the EAs (Evolutionary Algorithms), namely, MOEA/D, NSGAIII, and SPEA2, but this research revealed a number of limitations in existing EAs and so this thesis proposes an improved EA: FIEA (Fuzzy Inspired Evolutionary Algorithm) which uses a fuzzy analytical approach to perform multi-criteria optimization and is also instrumental in selecting a final DNN model from the Pareto optimal set. This approach also provides insight into how the hyper-parameters control accuracy, sensitivity, F1 and other performance metrics. This is a change from traditional approaches which apply DNNs as a black box. The interpretability improved in this way can be used to advance or adjust DNNs' behaviour and there is evidence that FIEA-optimized DNNs perform better than other algorithms described in the literature.

Acknowledgments

First and foremost, I thank the Almighty for giving me the strength and patience to work through all these years.

My deepest gratitude goes to my main supervisor and mentor Associate Professor Prashant Kumar Jamwal. I am thankful for professional recommendations and invaluable support he has provided during my Ph.D. research. It was a great honour and pleasure to conduct research under his supervision.

My thankfulness and admiration also goes to my co-supervisors – Assistant Professor Amin Zollanvari for his timely advices in machine learning models and Professor Bharat Jasani who helped me with the understanding of breast cancer and analyzing the oncology databases.

I would also like to thank Autonomous University of Madrid, where I spend brief period of my Ph.D. research. I would like to put on record my thanks to Professor Ana Maria Gonzalez for her guidance during research conducted at Madrid.

With sense of gratitude, I would like to acknowledge the kind support and encouragement provided by the Director of Ph.D. Studies Associate Professor Luis R. Rojas-SOLÓRZANO.

I am enormously thankful to my mother for her love, prayers and support. I would also like to thank and appreciate all my Ph.D. colleagues and friends for their active support.

Table of Contents

List of Figures

List of Tables

List of Abbreviations

ANFIS – Adaptive Neuro-Fuzzy Inference System

ANNs – Artificial Neural Networks

ASMiGA – Archive-based steady-state micro genetic algorithm

AUROC – Area under the Receiver Operating Characteristics

CFS – Correlation based Feature Selection

CPPNs – Compositional pattern-producing networks

DMOEA – Dynamic multi-objective evolutionary algorithm

DNNs – Deep Neural Networks

EA – Evolutionary Algorithm

EANNs – Evolutionary Artificial Neural Networks

EENNs – Entity Embedding Neural Networks

FAMD – Factorial Analysis of Mixed Data

FIS – Fuzzy Inference System

GA – Genetic Algorithm

GMD – Generalized Mahalanobis Distance

HMOEA – Hybrid multi-objective evolutionary algorithm

HVDM – Heterogeneous Value Difference Metric

IARC – International Agency for Research on Cancer

IVDM – Interpolated Value Difference Metric

KKT – Karush-Kuhn-Tucker

Lasso – Least absolute shrinkage and selection operator

MCA – Multiple Correspondence Analysis

MDR – Mutual domination rate

MF – Membership function

MFS – Mixture Feature Selection

ML – Machine learning

MOEA – Multi-objective evolutionary algorithm

MOEA/D – Multi-objective evolutionary algorithm based on decomposition

MOP – Multi-objective problem

MS – Membership score

NEAT – NeurEvolution of Augmenting Topologies

NNs – Neural Networks

NSGA – Non-dominated Sorting Genetic Algorithm

OMS – Overall membership score

PCA – Principal Component Analysis

PF – Pareto Front

PLS – Partial Least Squares

RBF – Radial Basis Function

RECIST – Response Evaluation Criteria in Solid Tumors

SBS – Segment-based search

SPEA – Strength Pareto Evolutionary Algorithm

SSL – Semi-supervised learning

SVMs – Support Vector Machines

TWEANNs – Topology and weight evolving Artificial Neural Networks

VDM – Value Difference Metric

WHO – World Health Organization

WVDM – Windowed Value Difference Metric

1 Introduction

1.1 Background and Motivation

The World Health Organization (WHO) describes cancer as a term commonly used to characterize a broad category of diseases which affects various parts of the human body and growths abnormally. The term malignant neoplasm is also widely used in the literature. There are over hundred categories of cancer ailments which are the main sources of morbidity and mortality around the world. Cancer incidence and subsequent loss of life are both growing rapidly globally. Key factors in an increase in incidence of cancer may include unhealthy diet, physical inactivity, tobacco, and lower rates of childbearing as well as socioeconomic development and the ageing of the population [1]. WHO data indicates that cancer is a main cause of mortality in both emerging and developed countries (Figure 1.2) [1]. 18.1 million incidences and 9.6 million deaths were registered globally in 2018 [1].

1.1.1 Cancer and breast cancer

As reported by the International Agency for Research on Cancer (IARC) the highest number of cancer morbidity registered in Eastern Asia (5.6 million) in 2018, next Northern America (1.9) and South-Central Asia (1.7 million). Eastern Asia in the same year also produced the highest mortality rate (3.4 million), but in this case South-Central Asia was second with 1.2 million and North America came next with 693,000 [2]. The three most spread malignant tumor types in terms of new cases are lung, breast and colorectal (Figure 1.2). Moreover, they are among leading malignant tumor types in terms of mortality and cover up to 35 percent of total mortality (Figure 1.3) [1], [2].

Breast cancer is among the prevalent types of cancer. It develops in breast tissue when cell growth is uncontrolled. It is found almost entirely in women, with only 1% of breast

cancers affecting men accounting [3]. It has a negative influence on a patient's physiological, mental and emotional condition and shows high rates of both incidence and mortality.

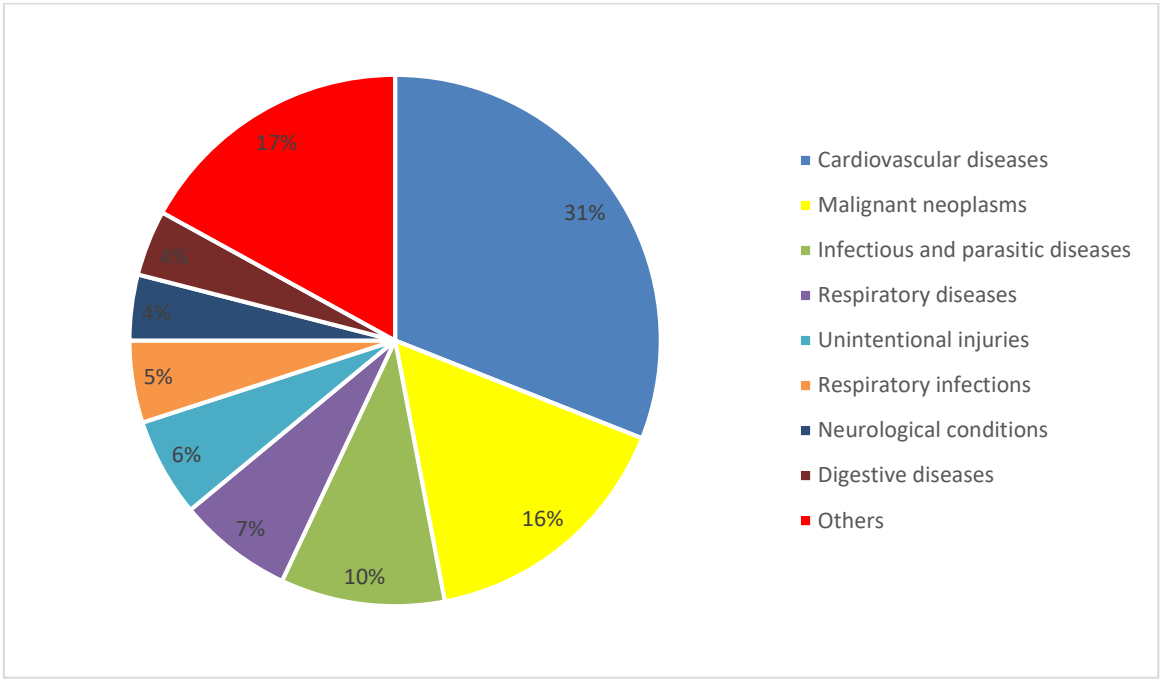


Figure 1.1. Causes of Death Worldwide, 2016 (Million) [1].

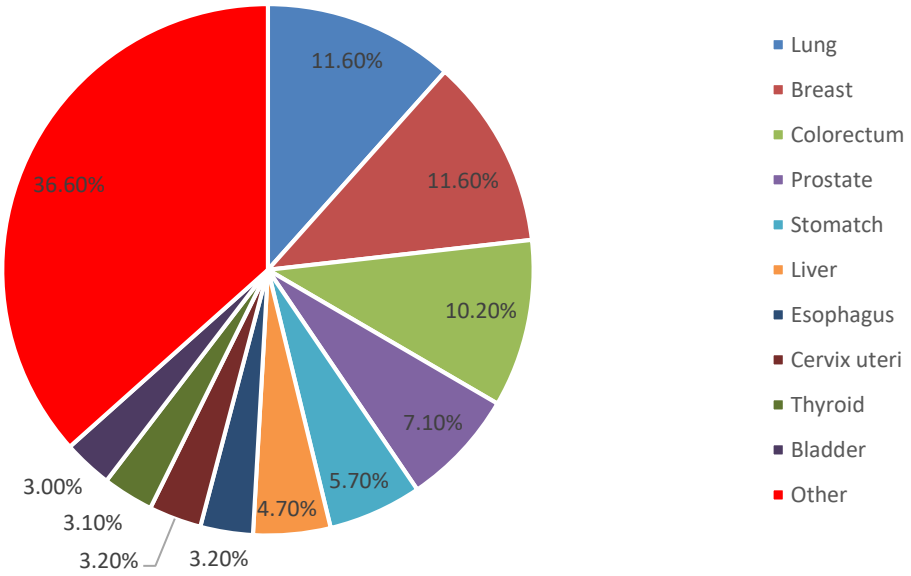


Figure 1.2. Cancer incidence, 2018 [1].

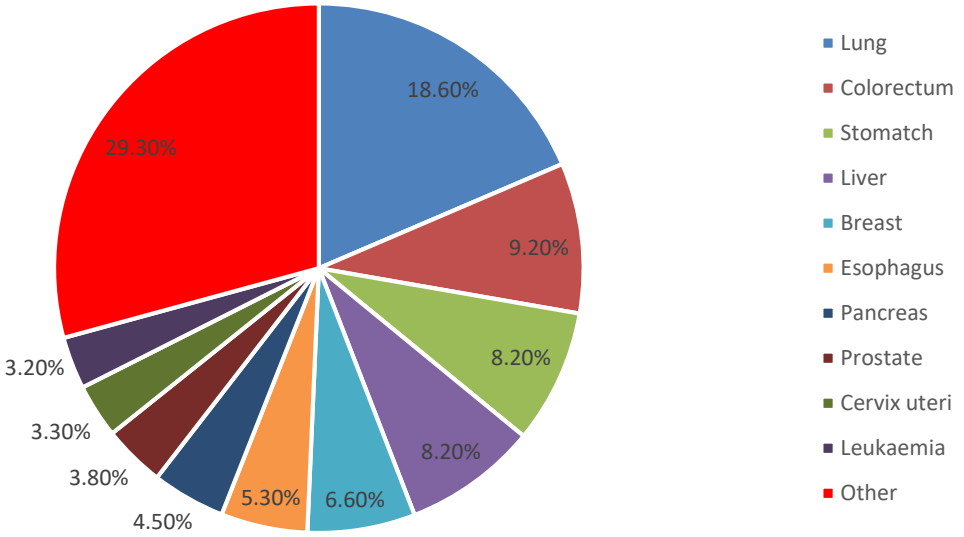


Figure 1.3. Cancer mortality, 2018 [1].

Between the 1970s and 2012, the number of women in the USA diagnosed with breast cancer rose from 9.09% to 12.4% [4], but this increase may be a result of increased life expectancy and environmental and lifestyle changes in addition to enhanced awareness and control. Around 2.1 million new cases and 626,000 deaths were recorded in 2018 (Figure 1.4) [1]. It is vital to indicate here that the breast cancer morbidity covers up to 25 percent of all variations of female cancers which demonstrates a highest incidence rate among all cancers within women [1]. Breast cancer is also highest for mortality, comprising 15 percent of all female cancer deaths (Figure 1.4). Breast cancer is the disease most often registered in many parts of the world, with highest rates in Australia/New Zealand, Northern Europe, North America and most of Sub-Saharan Africa (Figure 1.5). Death rates are also high in Melanesia and Polynesia, and in North Africa.

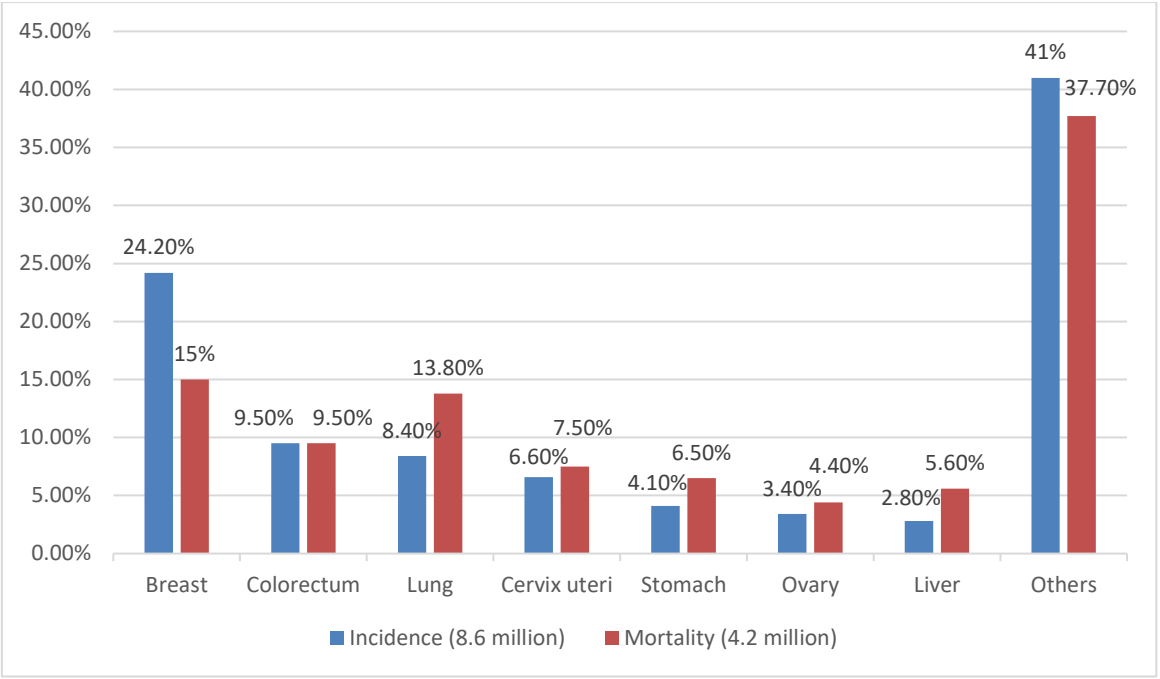


Figure 1.4. Cancer incidence and mortality for females, 2018 [1].

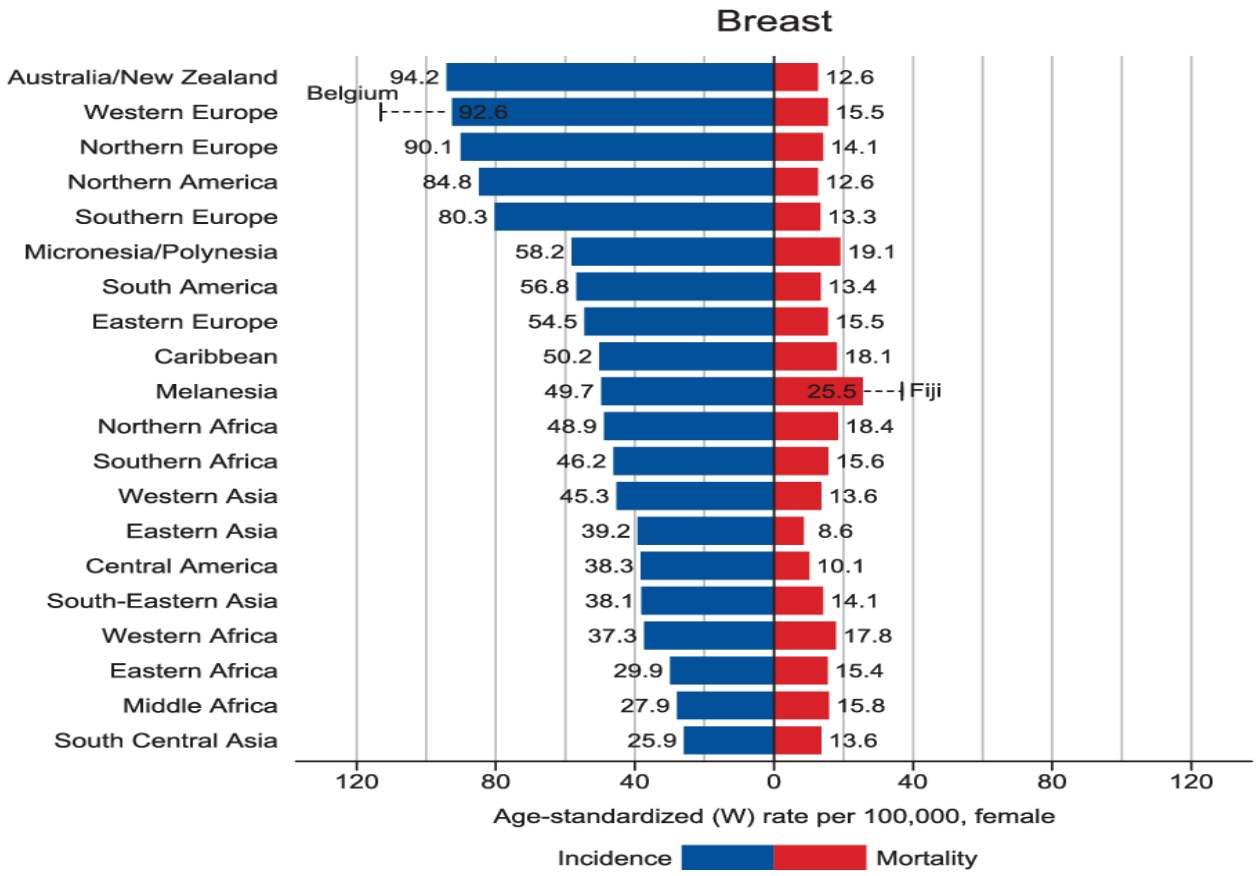


Figure 1.5. Region-specific incidence and mortality rates for breast cancer, 2018 [1].

A tumor evolves and in the long run the development goes uncontrollable in consequence of **formation of cancer cells in breast tissues**. The tumor further becomes malignant by invading into nearby tissues and the process is termed as a metastasis. Symptoms enabling early identification of breast cancer include **glandule, breast pain, nipple pain, skin irritation, reddening, thickness of nipple and a discharge from the nipple** [5]**.** Though no precise cause of the disease has yet been identified, it would seem that environmental and genetic factors are involved.

A recent rise in the rate of cancer occurrence and diagnosis has been accompanied by greater understanding and knowledge of how to control and prevent it. The evidence shows that cancer can to a great extent be prevented by evidence-based actions and, if not prevented, there is a high probability of cure for many types of cancer if they are detected and treated at an early stage. Accurate diagnosis and prognosis are vital if treatment is to succeed because every type of cancer requires a specific treatment regimen comprising surgery and/or radiotherapy and/or chemotherapy. Accurate diagnosis makes possible good incidence prediction and identification of the stage breast cancer has reached as well as prognosis of the disease's future course. Precise and reliable prognosis can provide information on the form and amount of intervention required [6], [7]. This will rescue numerous patients from experiencing needless painful and costly treatment protocols and simultaneously warn physicians in determining intensity of cure for some immediate cases [8]. The main objective is to treat cancer or to greatly extend life. Improving the survivor's and sufferer's quality of life is also a principal objective.

1.1.2 Prognosis and survivability

Prognosis is greatly dependent on evaluation of the treatment, sequela, receptivity, relapse and survival in patients or groups of patients after a treatment. It is the primary factor

in defining the treatment directly after the identification of the disease. Through prognosis it is possible to predict further course of a disease in order to guide the character and intensity of the therapeutic intervention, which needs handling vast amount of patient data. The three central aspects of prognosis are: susceptibility, recurrence, and survivability [9]. This research focuses on the survivability prediction and it can be described as the patient remaining alive for a certain time after the diagnosis of the tumor. Ordinary techniques of monitoring and predicting the tumor depend on detecting the existence of specific signal characteristics by a physician, while prognosis is a matter of rigorously analysing and synthesising past cancer data. This database has grown greatly in both length and breadth [10], [11].

1.1.3 Data Analytics

Health technologies have been enhancing medical services and clinical research, one of them is Big Data. The amount of healthcare data is extremely increasing and has huge potential. One of the examples is the field of oncology, where physician has to deal with enormous and diverse amount of patient data (medical reports, medical imaging, etc.) to monitor and predict an outcome of a treatment, which cannot be entirely mastered by a single physician [12]. Moreover, the contemporary fast-paced medical research generates such high volume of journal publications that it is virtually impossible to cope with all this information and characterize them in terms of clinical applicability. The physician is also confronted with lots of patient data (medical reports, medical imaging exams, etc.) to interpret when posing a diagnostic. It becomes even more critical in the context of cancer treatment where the patient life is at stake. Indeed, the volume of information dramatically increases in the course of the therapy because the patient has regular medical examinations to monitor treatment efficiency. For example, the Response Evaluation Criteria in Solid Tumours (RECIST) criterion uses the tumour size measured in medical images for assessment purposes [13]. If the cancerous tumour

stops growing, this is an indication that the patient may be a good responder to the treatment. However, one criterion is not sufficient to evaluate the therapeutic efficacy of a given treatment. Other criteria such as patient condition, medical history and genomic profile must be taken into account. Furthermore, the current patient data has to be compared with other patient data having similar characteristics. If the oncologist is able to successfully analyse available database in a holistic manner, an efficacious personalized cancer treatment can be offered to the patient. However, analysis of such enormous volume of data is difficult, cumbersome and time consuming in a clinical environment. As a result it might cause subjectivity which leads to incorrect diagnosis and prognosis. The US Department of Health and Human Services Office announced 180 000 death cases due to medical error during investigating the medical documentations of patients in 2008 [14]. Therefore, medical intelligence systems evolved as one of the primary research areas in the domain of medical diagnosis and prognosis. The goal of these works is to develop an intelligence system which can predict and recommend the optimal cancer treatment based on reliable models and analysis of large clinical archives, namely medical big data analytics. Development of such a system would constitute a breakthrough in the prevailing oncological practice.

1.1.4 Machine Learning and AI

Since 1990, prognosis has been improved by the use of a variety of approaches using statistical and machine learning algorithms. They include in no particular order Neural Networks (NNs), Logistic Regression, Genetic Algorithms (GAs), Decision Trees, Bayesian Networks, Support Vector Machines (SVMs), Fuzzy Inference System (FIS) and Ensemble Learning [15], [16]. Most of these algorithms are inspired by nature and provide high precision and accuracy in prognosis. Developments in machine learning make it possible to use larger data to train models. Deep Neural Network (DNN) model, for example, achieved remarkable

results when classifying images [17], [18] and current state of the art models perform extremely well in image and object recognition [19]. Moreover, DNNs brought dramatic changes in computer vision, speech recognition and natural language processing, all areas in which they are replacing traditional methods [20]–[22].

1.1.5 Dimensionality reduction

Apparently, processing medical data is computationally expensive and usually becomes erroneous due to redundancy and irrelevance. Therefore, many different dimensionality reduction methods are proposed and some of these are widely used [23]–[28]. The role of dimensionality reduction is to improve algorithms by avoiding overfitting errors and providing faster and cost-effective models [29]–[31]. Dimensionality reduction can be further separated into two categories, namely, feature selection and feature extraction [32], [33]. Feature selection techniques select smaller subset of original variables without alteration and can be further classified into three subcategories, i.e. filter, wrapper and embedded [26]–[29], [32]–[34]. On the contrary, feature extraction methods reduce dimensions by transforming the original variables into a new subspace by choosing variables with highest variance [32], [34].

1.1.6 Heterogeneous data

The challenges we mostly encounter in nature are often continuous. Moreover, they are unstructured and intrinsic. Unlike unstructured data, we also frequently face continuous and categorical structured data, which are termed as heterogeneous data. The categorical data are presented in many databases and usually are lack of continuity. They present on both ordinal and nominal scales. In an ordinal scale, categorical data entities are indexed numerically, but the numbers are arbitrary and not representative of actual distance between entities. For example, usually to evaluate some event we assign numbers from one to three for expressions “bad”, “average” and “good”, but it is just ordering and we do not know exact measures

between these entities. An example of a nominal scale in demographic or socio-economic analysis would be marital status, where assigned values might be “single”, “married”, “divorced” and “widowed”. The data’s non-continuous nature limits many machine learning algorithms to perform properly on categorical data. An ordinary method to resolve this issue is to use one- hot encoding, but it has number of limitations. The first is that categorical variables with high cardinality lead to a large vector being output by one-hot encoding, making computation more complex. The second is that each categorical variable entity is treated independently with no consideration of their intrinsic interrelationships. These disadvantages can be resolved by transforming categorical data into numerical vector representation, which are entity embedding of the categorical data.

Most of the existing dimensionality reduction methods can process data composed of either continuous or categorical variables. However, in many practical applications the available data has heterogeneous variables [35]–[38]. For example, cancer data contains continuous variable as tumour size and categorical variable as tumour type [39]. Also, population, socio-economic and marketing databases include mix variables such as marital status, education, age and number of children [40]. Another example is aggregated quantitative neuroimaging data and qualitative genetic data to study brain structure [36]. There are number of approaches to process heterogeneous variables. Firstly, the fundamental approach for many methods based on applying appropriate distance function [41]. However, most of them rely on Euclidean Distance function, which works with continuous variables. Other group of distance functions such as Value Difference Metric (VDM) is suitable for categorical variables. Therefore, applying distance function that works on mixed variables is fundamental and proper direction. There are three heterogeneous distance functions such as Heterogeneous Value Difference Metric (HVDM), Interpolated Value Difference Metric (IVDM) and Windowed Value Difference Metric (WVDM), but they do not involve dimensionality reduction.

Secondly, attributes can be adapted to same format before the learning. One way is to transfer categorical variables into discrete numerical values, but this technique is not suited since range between samples would then have no meaning. Another way is to discretize continuous variables, which results in loss of information [40], [42], [43]. Thirdly, we can apply the state-of-the-art method "Entity Embedding", which is a mapping of categorical variables into Euclidean Space [44]. The transformation is learned by Deep Neural Networks. Finally, we can transform categorical variables using one hot encoding method, where we normalize variables and combine them with standardized continuous variables, afterwards appropriate grouping or projection based method is applied [45].

1.1.7 Word embedding

Hinton introduced continuous vector representation, also called a word embedding [46], which was a notable breakthrough in natural language processing involving language modelling [47], machine translation [48] and text classification [49]. Among various implementations, Word2Vec [50], GloVe [51] and fastText [52] are both well-known and robust. These models work on the principle of transforming a word's semantic or syntactic attributes into a low dimensional continuous vector representation, resolving dimensionality and word similarity problems. The words King and Man or Queen and Woman are close to each other in Euclidean Space where dimensions are predetermined [50]. They can be represented algebraically in the form of the equation Queen = (King – Man) + Women. While word embedding's primary use is for language modelling and text classification, it has also been used to process categorical data which is referred as an entity embedding. Researchers using this method have produced promising results, with increased accuracy for most classifiers [44], [53]. The goal of the entity embedding in neural networks is to map an each entity of categorical variable to some continuous vector representation of defined dimensions. This is

equivalent to representing categorical variable using one hot encoding then multiplying with weights of an extra layer of linear neurons, which is embedding layer. The weights of embedding layer are learned as other parameters of neural network. Next, transformed categorical variables together with continuous variables act as an input layer for other layers. The entire neural network model is trained using backpropagation algorithm. The mathematical and other details of the entity embedding is clearly described in this manuscript [44].

1.1.8 Neuroevolution

Deep Neural Network or occasionally mentioned as hierarchical learning paradigm is a mathematical generalization of the human biological information processing system [54]. A special class of Neural Networks achieve learning through a process of evolution, hence these networks are termed as Evolutionary Artificial Neural Networks (EANNs) [55], [56]. They gradually advanced using Evolutionary Algorithms (EAs), which are search methods and draws inspirations from Darwinian's evolution theory [57]. The object of neuroevolution is to find a different training approach from a standard Neural Networks referred as backpropagations, which is also called as a stochastic gradient descent. Because of the evolutionary mechanism inherent in them, these algorithms are expected to outperform such standard learning algorithms as backpropagation and hybrid schemes. Beginning in the 1980s, weights of the complex networks were obtained through an evolutionary approach which did not alter the architecture and was therefore known as Fixed-Topology Neuroevolution [58]. Changing weights resulted in evolution of the intensity or strength of neurons' knowledge, but there could be lack of generation or evolution of a new knowledge since the network configuration was unchanged. Moreover, initial population of networks generated randomly without incorporating some prior knowledge also did not help to improve network

performance. Changing network configuration together with weights and biases was researched in the late 1990s and it was known as topology and weight evolving ANNs (TWEANNs) [59].

1.1.9 Optimization of hyper parameters

A hybrid algorithm comprising a backpropagation and a gradient descent algorithm made it possible to obtain network parameters while using DNNs. However, such hyper-parameters as the number of deep layers and neurons, must be chosen by the developer, who is also responsible for selecting activation functions, learning rate and other parameters. This hyper-parameter choice is therefore heavily dependent on the developer's insight and experience and is significantly subjective. One of the objectives of this research is to obtain optimal hyper-parameters which will enhance performance of DNN.

Assessment of DNN performance is based on various performance indices, which include accuracy, F1 score, sensitivity and specificity among others. The objective of obtaining optimal hyper-parameters becomes a multi-objective optimization problem, because there are many performance indices which require optimization. A simple approach would be aggregation of all performance indices into a single objective function which could be then optimized using existing single-objective optimization methods, but the nature of these performance indices and their interpretations makes it difficult to add or multiply together to form a single objective function.

In the light of above, a multi-objective optimization is required to be performed in order to optimize DNN and obtain optimal hyper-parameters. A number of optimization methods exist, but EAs are the best choice since they can simultaneously optimize several performance indices and return a Pareto optimal set of DNNs. Such a solution is described as Pareto optimal because all of the resulting DNNs will be equally good. This further means that a user will have many choices to select from and depending on user's priorities for performance indices

different DNNs can be made available. The next sub-chapter briefly discusses existing evolutionary algorithms as well as some challenges which researchers are trying to solve.

1.1.10 Evolutionary algorithms

The meaning of optimization differs from its regular scope of maximization or minimization while solving multi-objective problems. Problems from actual world establish challenges which require a best acceptable compromised solution. Faced with multiple conflicting goals that are matchless and incommensurate, the optimal solution is usually takes the form of a non-singular equitable solution set [60]. Owing to the varied perception of human end user, it is desired to obtain a wider set of equitable solutions comprising combinations of trade-offs established between objectives. Deriving such set of solutions is cumbersome process when conventional optimization methods are applied. Therefore, evolutionary algorithms have emerged as a reasonable option to the conventional approaches within past twenty years. Few simulations are needed for EAs to provide equitable solutions during performing synchronous optimization of multitude of objectives [61]. Unlike conventional optimization methods, EAs work with population of solutions; thanks to their inherent evolution mechanism derived from natural evolution processes, they make it possible to explore improved Pareto solutions. Moreover, EAs do not need derivatives of objective functions and have robust operators like reproduction and regeneration to avoid convergence to local optima. Also, they have been applied and considerably advantaged in various areas such as engineering design, groundwater monitoring, autonomous vehicle navigation, polymer extrusion and city planning [62]. Obviously, the concept of Pareto dominance is conceptualized during optimization of mutually conflicting objectives [63] , hence EAs have been favoured in majority of the real world multi-objective problems (MOPs) [64], [65]. Primarily among

evolutionary algorithms majority of the researchers preferred the idea of Non-dominated sorting Genetic Algorithm (NSGA) in their research [61].

Regardless of productive application of EAs in solving multi-objective problems (MOPs) within last twenty years, there are specific problems which need instant attention. As stated previously, the concept of non-dominance is primarily used in EAs while selecting/sorting better solutions during the optimization process. However, it loses its importance while applying on multi–objective optimization problems [61]. Consequently, majority of solutions become non-dominated and populate the Pareto front giving a pseudo Pareto Front (PF), which leads to premature algorithm convergence and failure to reach the optimal solution. Hence, EAs are inefficacious in optimizing multi-objective optimization problems [60], [66]–[68].

A significant advantage of using EAs to optimize DNNs is the ability to generate powerful synergy by combining evolution with learning. Gradient-based learning algorithms have a known sensitivity to the initial hyper-parameter values and network performance is significantly degraded when sub-optimal parameter values are obtained. This can be counteracted by the use of EAs to find suitable hyper-parameter network values for subsequent training with Levenberg-Marquardt optimization.

1.2 Research objectives

Reading through the extant literature on breast cancer data analytics and evolutionary algorithms, several research opportunities are identified and consolidated in the form of research objectives as below:

1. Access a SEER breast cancer database and perform data pre-processing in order to obtain a concise data (dimension reduction) of critical or vital parameters without loss of important information concerning the prognosis of survivability;

2. Develop a method to convert high cardinality categorical variables into continuous data for better interpretation and use with Neural Networks;
3. Design of a neuroevolution system for prognosis of breast cancer survivability;
4. Multi-objective optimization of DNNs to obtain optimal set of hyper parameters;
5. Develop an improved alternate evolutionary algorithm addressing the challenges encountered with the existing EAs;
6. Compare and benchmark performance of the proposed evolutionary algorithm with existing algorithms.

1.3 Research contributions

Some notable contributions were made during this research.

- ✓ The SEER data which in fact is a big data is pre-processed using various methods of dimensionality reduction. The resulting database consists of critical parameters ensuring that important information is not lost in this process;
- ✓ A factorial analysis of large and mixed data was performed for the first time during this research;
- ✓ Neural networks based entity embedding was performed on categorical data to be used with continuous data for breast cancer data analytics. The entity embedding again is proposed for the first time in this thesis and is a novel contribution;
- ✓ A neuroevolution method that optimizes Deep Learning Networks using Evolutionary Algorithms has been developed for prognostic modeling of breast cancer data for survivability prediction;
- ✓ A new evolutionary algorithm called, "Fuzzy Inspired Evolutionary Algorithm (FIEA)" was developed during the course of present research and it is later used to obtain optimal

set of hyper-parameters of DNN through multi-criteria optimization of their performance metrics;

- ✓ The newly developed evolutionary algorithm was compared with the existing state of the art EAs and it was found that proposed method (FIEA) outperformed the existing EAs.

2 published journal articles:

[1] B. Abdikenov, Z. Iklassov, A. Sharipov, S. Hussain and P. Jamwal, "Analytics of Heterogeneous Breast Cancer Data Using Neuroevolution", IEEE Access, Vol 7, 2019;

[2] P. Jamwal, B. Abdikenov, S. Hussain and M. Ghayesh, "Evolutionary Optimization using Equitable Fuzzy Sorting Genetic Algorithm (EFSGA)", IEEE Access, Vol 7, 2019.

2 Literature review

2.1 Breast cancer prognosis

During last two decades, rapid escalation in the research has been observed on various types of machine learning (ML) and statistical techniques applied to the diagnosis and prognosis of critical ailments.

An early research was done by Burke and others [69]–[72], where they conducted seminal research on the application of NNs for cancer prognosis and compared it with other statistical models. Later Delen et al. [73] applied three popular algorithms such as Logistic Regression, Decision Trees and NNs to predict survivability of breast cancer diagnosis. During these experiments SEER database was used after reducing to 17 variables and 202,932 records subsequent to data pre-processing. The decision tree achieved 93.6% accuracy, which was the best result among other approaches.

In [16], the authors carried out an exhaustive survey of various machine learning techniques and presented increasing dependency on protein biomarkers and microarray data. Use of ANNs is also recommended over other methods. Further, reasonable efforts devoted to data quality and validation issues as well. This review gives complete and sufficient information on the usage of ML techniques in cancer prognosis. In a more recent review [15], different supervised ML techniques and various databases have been presented. It is apparent from the work presented that SVM technique should be a prevalent choice owing to relatively high accuracy. Authors conclude that heterogeneous data and hybrid model can also provide accurate results.

Gevaert et al. [72] combined clinical and microarray data and using Bayesian networks they were able to achieve Area under the Receiver Operating Characteristics (AUROC) curve

of 0.845. Other group of researchers combined clinical data with genetic markers and used I-RELIEF algorithm [74] to achieve improved AUROC. In [75], the authors developed a breast cancer prognosis prediction system based on clinical and microarray data. They compared three ANNs and four SVMs, among them, conjugate gradient ANNs and linear SVMs showed the best precision which are 100% and 99%.

In [9] an analysis of different SVMs based classifiers has been presented. Authors tested the potential of classifiers in conjunction with three kernel methods which are Radial Basis Function (RBF), Polynomial and Sigmoid. It is shown that C-SVM classifier combined with RBF kernel achieved higher accuracy (83.4%) than other classifiers. In [9], researchers investigated robust predictive model by comparing ANNs, SVMs and semi-supervised learning (SSL) techniques on SEER data. As a result, it is suggested that SSL is a reliable model that would allow oncologists to run precise survival analysis. Recent work on application of machine learning on medical data was conducted by [76], where they applied 13 state-of-the-art methods on 165 open data classification problems. Researchers presented Gradient Tree Boosting as best method and gave valuable advices on parameter tuning.

Many of the contemporary ML techniques are mature and provide high accuracy on precise datasets, but are not usually suitable in dealing with human knowledge and interpretation involving vagueness. In contrast, fuzzy systems has been used for modelling a human-like learning and decision making processes [77]. Fuzzy sets and fuzzy logic have been applied in various domains, including medicine, and shown to be an alternative method to most ML and statistical methods from the first days they were represented by Zadeh. Fuzzy expert systems by employing ML techniques and artificial intelligence together with fuzzy logic can help modelling uncertainties and imprecision. Fuzzy inference systems apply a number of "if-then" rules to conduct non-linear transformations from input space to output space. These "if-then rules" are the core of a fuzzy inference system. However, it has some vulnerable points in

its learning algorithm which are: scarcity of standard algorithms for mapping human knowledge or experience into the rule base and database of fuzzy inference system, and necessity in efficacious algorithms for optimizing the membership function. Many algorithms have been implemented to optimize fuzzy models in order to achieve best prediction performance.

The Adaptive Neuro-Fuzzy Inference System (ANFIS) introduced by J.S. R. Jang is an alternate algorithm which aggregates the excellence of both fuzzy rule based system and neural networks. Many other researchers utilized this method in their studies for obtaining better performance. In research conducted by Ubeyli precision of 99.08% was obtained [78].

In [79], the researchers developed ten ANFIS models and trained models using two various algorithms. The neural network backpropagation and the mixture of least squares and backpropagation gradient descent method. As expected, hybrid method showed better performance. In another research, authors compared partial logistic ANNs against ANFIS on Nottingham Breast Cancer data set, where they got similar precision [80].

In [81], discussed issue of variability during decision making. They applied concept of "non-stationary fuzzy set", which allows change of membership function over time. A set of 1310 cases and 6 variables used to implement ANFIS model. The test outcomes show that the standard fuzzy inference systems achieves overall 84.6%, while model proposed by authors increase performance to around 88.1%. Moreover, paper describes input variables and methods in detail, also provides some recommendations on handling variability. In [82], authors showed that prognosis results can be more accurate when using clinic pathologic and genomic data. Firstly, they choose different number of input parameters based on five feature selection methods (Genetic algorithm (GA), Person's correlation coefficient, Relief-F, CC and GA, Relief-F and GA). Next, data with appropriate selected parameters are fed into four

classification models (ANNs, SVMs, Linear Regression and ANFIS). As a result, the hybrid model Relief-F-GA-ANFIS showed the best accuracy of 93.81%.

2.2 Dimensionality reduction

Feature selection and feature extraction methods have been successfully utilized in developing robust systems in healthcare and other areas. One of the initial studies were conducted by Kira and Rendall [83], where authors efficiently resolved classification tasks involving highly correlated variables and algorithm Relief emerged. Later, other group of researchers [83] extended the algorithm by solving noise, multiclass and regression problems. Tibshirani [84] proposed a method called "Lasso", which selects good features by shrinking some coefficients and setting others to zero. In [85], authors resolved the limitations of the "Lasso" method and presented a new algorithm "Elastic net".

Guyon and Elisseeff presented a comprehensive study which covered a broad aspects of various feature selection methods and demonstrated increasing reliance on text categorization and microarray data. Moreover, they recommend using a correlation coefficient or mutual information with a variable ranking method and performing forward or backward selection with a nested subset selection.

In [86], authors proposed a technique named Supervised Principal Components and applied on DNA microarray data. It is analogous to regular Principal Components Analysis (PCA) except that it uses a subset of the variables correlated with the output variable. PCA reduces the number of variables and these reduced variables are called principal components. In addition, they retain the most possible variance and are linear combinations of the original variables. Also, in the recent study [87] sparsity and non-linearity problems are solved using supervised principal component analysis. Other groups of researchers [88], [89] used Partial Least Squares (PLS) technique on genomic data for classification and regression problems.

These techniques belong to the group of projection based methods, the differences are in sparse, linear, nonlinear, supervised and unsupervised nature of data.

2.3 Heterogeneous data

In [23], Hall proposed a correlation-based filter algorithm which is suitable for both continuous and discrete problems. The result of study outperformed the extended Relief (ReliefF) algorithm. Tang and Mao proposed a novel algorithm called a mixture feature selection (MFS) algorithm [42]. It includes feature evaluation and search algorithm for heterogonous variables. The MFS was compared with Correlation based Feature Selection (CFS), generalized Mahalanobis distance (GMD) and Relief algorithm. According to results the MFS performed better compare to other algorithms. However, it can be applied only to data with medium number of variables. Number of studies [90], [91] were conducted on application of rough set on mixed variables and presented promising results. In recent publications [39], [92] better results were obtained by combining classical and fuzzy rough set methods. In another study [40] Doquire and Verleysen implemented mutual information based embedded technique for mixed data, which combined filter and wrapper approaches. The concept is to separately evaluate continuous and categorical variables, subsequently merge them. It was tested on number of artificial and real data. Moreover, the technique had better results compare to the CFS algorithm when nonlinearity exists between variables. Summarizing the literature, we can infer that Logistic regression, Neural Networks and SVMs are prevalent classifiers for cancer prognosis. In addition, we can observe trend in microarray data and reliance on SEER database. However, conventional methods in dimensionality reduction were applied only to homogenous data; there are recent studies where application of a rough set on heterogeneous data is presented. Moreover, there are diverse variations of projection based methods in the literature, but still these approaches are not adopted to mixed data.

2.4 Neuroevolution

The algorithm NeuroEvolution of Augmenting Topologies (NEAT) was an augmenting topology broadly applied in neuroevolution [93]. To accelerate ANNs training and decrease the number of function evaluations, researchers from neuroevolution society suggested an alternative genetic encoding class which they named indirect encodings; it makes use of less genes than the amount of connections and neurons in the network and made possible the evolution of larger ANNs that were complicated to train using NEAT. Compositional pattern-producing networks (CPPNs) are an example of indirect encodings [94]. Another successful research is the idea of giving preference in the consecutive evolutionary process to novel parents over those giving better network accuracy. This new concept is named a novelty search and networks are trained to be precise and concurrently investigate better alternative solutions [95]. Nevertheless, there are other group of evolutionary methods that depend on Fogel's evolutionary programming which emphasizes on the evolution of NNs behaviours through its architecture [96].

2.5 Evolutionary algorithms

Evolutionary algorithms such as NSGA-II and Strength Pareto Evolutionary Algorithm II (SPEA-II) have been successfully implemented in optimization of two objectives, but they presented inefficacious results in cases of three or more objectives [67], [68]. Because of its weak capabilities the widespread non-domination method may not perform well on optimization problems involving many objectives [97]–[99]. The algorithm's advance means that many solutions become non-dominated and populate the Pareto Front, leading to the algorithm's quick convergence to a pseudo non-dominated front. This has led to many proposed solutions of which modifying the PF has been the most investigated. Segment-based search (SBS) has been suggested as a variation operator to improve EA performance [100]. A

different approach [101] reduced the number of non-dominated solutions as a way of improving PF quality, while the PF has also been shown to be improved by different non-dominated solution ranking [102]. *Substitute distance assignment schemes* have also been suggested to improve PF quality [103], enhancing evolutionary algorithm performance by replacing the crowding distance operator. Koppen and Yoshida [103] put forward a number of indicators apart from non-dominance ranking to establish whether one solution was better than others. Next, different metrics, to establish and measure quality of solutions, have been provided in [104] to compare potential solutions and select the most competitive one. Unfortunately, methods depend on set quality measure using some indicators cannot be used during optimization due to their enormous computational expenses.

A dynamical multi-objective evolutionary algorithm (DMOEA) was proposed in [68] and compared with other algorithms, but this approach has two limitations. The first is that it considers all objectives equally important. The second is that it requires the extreme values of all objectives to be known before optimization and this may not be possible. Differential evolution has been extended for use in multi-objective optimization by means of a grid-based adaptive multi-objective differential evolution algorithm [105]. A solution's rank is assigned by means of three indices such as grid fitness, grid density, and grid-objective-wise standard deviation, but the computationally overhead is high in return for no advantage over a number of existing approaches. Greater efficiency and search performance has been claimed for an approximate non-dominated sorting algorithm suggested for use where optimization problems involve more than three objectives [106] in which dominance between two solutions is decided by comparison of up to three objectives with one the objectives. A suggested archive-based steady-state micro genetic algorithm (ASMiGA) [107] maintains in the archive a set of non-dominated solutions to a minimum allowable size, with improved mating and selection schemes for better performance. A hybrid algorithm (FP–NSGA-II) [108] combining the NSGA-II with a front

prediction algorithm is claimed to give better PF approximation. A hybrid multi-objective evolutionary algorithm (HMOEA) for real-valued MOPs suggested in [109] would have a non-dominated archive of personal best during evolution maintained by each solution in the population but greater computational complexity means that this may not be the best method. NSGA III [110], [111] has also been suggested as a possible solution for generic constrained optimization problems involving multiple objectives.

As well as an improved sorting mechanism, EAs need a clear termination criterion. An optimization algorithm should terminate either on reaching the global optima or on obtaining the values of objectives in their acceptable ranges. Most algorithms have no termination scheme because they have no mechanism to ascertain their final solutions' global optimality. A research [112] suggested using an optimization convergence curve to terminate optimization, while standard deviation of the maximum crowding distance criterion as well as other metrics have been proposed by [112] and [113] to bring the algorithm's stagnation to the researcher's attention. Another approach was end optimization when the rate at which the solutions are improving falls below a pre-established threshold value [114], [115]. However, establishing that value could be problematic in the case of higher dimension optimization problems. Fernandez et al. [116] proposed the Kolmogorov–Smirnov statistical test in which performance metrics are used to evaluate convergence; suitable metrics would include generational distance, hyper-volume and spread of the competing solutions in succeeding generations. Liu et al. [117] explored this approach further in the search for an online convergence detection criterion according to which optimization can be stopped at the point when similarity between competing solutions' mean and variance, and therefore convergence, is indicated by such statistical tests as t-test and 2-variance test . Deb et al. [118] suggested using the Karush-Kuhn-Tucker (KKT) proximity measure as the criterion to terminate an evolutionary multi-objective algorithm, while [119] suggested a hybrid EA framework to address uncertainty concerning

termination. MGBM, a global stopping criterion combining a mutual domination rate (MDR) indicator with a simplified Kalman filter for evidence-gathering purposes, has also been proposed [120] , but the assumption underlying all the above approaches is that EAs improve solutions in the initial stage of evolution and it may be that this is not always correct.

An important issue concerning EAs is choice of a final solution from the assortment of Pareto optimal solutions. While all PF final solutions are non-dominated, the end user or decision-maker looks for a single solution that can be seen as best of the best. This has not been heavily researched, though fuzzy inference has been one of the methods used to select a final solution from the Pareto solutions [114], [116], [117], [121]. After simultaneous optimization, another way to ease the decision-maker's task may be to reduce the number of objectives [122]. The literature mentions other visualization techniques [123] in which objectives are mapped into a low dimensional space for better visualisation. [115] classified various approaches to find the final solution.

It is apparent from this discussion that, though EAs solve a number of objective optimization problems, they suffer from low discrimination between solutions, have no clear termination criterion and need a way of selecting the best compromise solution after optimization. Researchers have tended to treat some of these difficulties as isolated research issues [108], [111], [116], [118], [124], but there is no single approach capable of effectively addressing all the above areas where EA needs to be improved. A multi-objective evolutionary algorithm based on decomposition (MOEA/D) had been proposed as a way of improving existing EAs [125]–[128] by decomposing the multi-objective problem into a number of single objective optimization problems which are then solved concurrently to produce a PF, but using MOEA/D will not solve problems with a non-convex solution space because exploring the entire objective space would require an infinite number of iterations of objective function aggregations with different weight vectors.

3 Heterogeneous data

3.1 Database

Prognosis needs thorough analysis and synthesis of historical data on breast cancer [11]. Different databases, that have grown enormously, over time, in both length and breadth [129], [130] can either be publicly accessed (e.g., online) or may be acquired through collaboration between institutions and research teams. Available and popular breast cancer databases include Wisconsin breast cancer diagnosis [131], the University of Nottingham's breast cancer data [77], [132] and SEER data [133] as well as data held by hospitals in Croatia [134] and Thailand [135]. A comprehensive literature review on breast cancer databases was performed and a list of investigated databases is provided in Table 3.1. Despite the fact that sufficient number of cancer databases are accessible, majority of these databases have unacceptable small sample size and small number of parameters for constructing a robust prognosis model. Moreover, some databases have short or unknown data collection period. All these pitfalls adversely influence prediction performance of breast cancer prognosis. In this research we used the Surveillance, Epidemiology, and End Results (SEER) Program data of the National Cancer Institute, which is a fount of comprehensive data on cancer incidence and survival in the United States including about 34.6 % of the population [133]. The latest release has almost eleven million cancer cases from 1975 to 2017. The SEER Program collects and arranges cancer data into nine separate groups. The eight general groups are: breast, colon and rectum, other digestive, female genital, lymphoma and leukemia, male genital, respiratory, and urinary. But when the amount of overall cases are small for some types, they are arranged to constitute a group of "other" cancers, i.e. the ninth group. There are 149 variables for each cancer case and cases are collected from nine registries at different geographical locations.

Table 3.1. Databases for breast cancer.

#	Name	Size	Period	Number of variables
1.	Surveillance, Epidemiology, and End Results (SEER) Program data of the National Cancer Institute, USA [133].	1 631 572	1973-2015	149
2.	Danish Breast Cancer Cooperative Group (DBCG) Database, Denmark [136].	110 000	1977-2014	6
3.	Korean Breast Cancer Society Database, South Korea [137].	16 398	1996-2010	18
4.	Uppsala/Orebro Regional Breast Cancer Register, Sweden [138].	12 163	1992-2002	9
5.	Lynn Sage Database (Lynn Sage Comprehensive Breast Center at Northwestern Memorial Hospital), USA [139].	6 726	1990-2015	23
6.	The University of Texas MD Anderson Cancer Center (Houston, TX), USA [140].	3728	1997-2006	10
7.	Chung Shan Medical University Hospital, Jen-Ai Hospital, Far Eastern Memorial Hospital Tumor Registry, Taiwan [141].	2964	2010-2016	20
8.	Geneva University Hospital, Switzerland [142].	2481	1998-2019	22
9.	Srinagarind Hospital, Thailand [135].	2462	1990-2001	26
10.	Iranian Center for Breast Cancer, Iran [143].	1 189	1998-2008	22
11.	Nottingham Tenovus Primary Breast Cancer Series, UK [77], [132].	1 076	1986-1998	9
12.	National Institute of Oncology of Rabat, Morocco [144].	716	2009	22
13.	Wisconsin Breast Cancer Data, USA [131].	569	-	32
14.	Policlinico Tor Vergata Biospecimen Cancer Repository, Italy [145].	454	2007-2018	13
15.	Institute Portuquese of Oncology of Porto, Portugal [146].	399	-	16
16.	Clinical Center of Kragujevac, Croatia [134].	146	-	58
17.	Grimsby Hospital, UK [147].	92	1998-2007	19
18.	National Cancer Institute, Egypt [148].	60	-	11

The SEER database has been extensively used in a diverse research projects, which can be tracked in PubMed and Scopus indexing platform. In 2018, Hegselmann *et al.* [149] had

performed exhaustive search on prognosis of breast using SEER database and as a result, after cleaning and pre-processing 34 research articles were identified. Similarly, in our present research 12 research articles were identified (Table 3.2) after comprehensive pre-processing. Most of the publications are identical to the list in Hegselmann's work, but few new publications were added excluding publications with small sizes. From the literature it is clear that current best performance metrics are achieved by Lotfnezhad *et al.* [150]. In their research they have achieved accuracy (0.9670), sensitivity (0.9770) and specificity (0.9560) using SVM and improved the results of Delen *et al.* [73] (accuracy as 0.9360; sensitivity as 0.9602; and specificity as 0.9066). However, both the works did not consider F1 Score and AUC. Similarly, Hegselmann *et al.* [149] achieved highest results for accuracy (0.9134), F1 (0.9517), AUC (0.9062). Later, in other chapters results of these three research works will be used as a benchmark (Table 3.5, Table 3.10, and Table 4.6).

Table 3.2. List of publications from literature on breast cancer prediction using SEER database.

#	Title	Output variable	Number of input variables	Performance of models					Sample size
1.	Predicting overall survivability in comorbidity of cancers: A data mining approach. Zolbanin,*et al.* (2015) [151].	Survivability	33		Accuracy	Sensitivity	Specificity		17 907
				RF	0.7525	0.6563	0.7562		
				NN	0.7088	0.6229	0.7618		
				DT	0.7288	0.5728	0.8250		
				LR	0.7134	0.6339	0.7500		
2.	Breast cancer data analysis for survivability studies and prediction. Shukla *et al.* (2018) [152].	Survivability	26			Accuracy			85 189
				SOM-DBSCAN + MLP		0.9048			
3.	Robust predictive model for evaluating breast cancer survivability. Park, *et al.* (2013) [9].	Survivability	16		Accuracy	Sensitivity	Specificity	AUC	162 500
				ANN	0.6500	0.7300	0.5800	0.7000	
				SVM	0.5100	0.6500	0.5200	0.8000	
				SSL 0.7100	0.7100	0.7600	0.6500	0.7800	
4.	A dynamic gradient boosting machine using genetic optimizer for practical breast cancer prognosis. .Lu, *et al.* (2019) [153].	Survivability	14		Accuracy	Sensitivity	Specificity	AUC	82 707
				GAOGB	0.7503	0.8136	0.6877	0.7507	
5.	Optimal data mining method for predicting breast cancer survivability. Wang, *et al.* (2013) [154].	Survivability	14		Accuracy	Sensitivity	Specificity	AUC	215 375
				DT (J48)	0.9107	0.9961	0.8180	0.7178	
				LR	0.9123	0.9893	0.1646	0.8999	
				Rule based classifier	0.8772	0.5632	0.9178	-	
6.	Cancer Survivability with Logistic	Survivability	17		Accuracy	Precision	Recall	F1 Score	338 596

	Regression. Bozorgi, *et al.* (2018) [155].			LR	0.8970	0.9000	0.9900	0.9400	
7.	Predicting Breast Cancer Survivability Using Data Mining Techniques. Bellaachia, *et al*. (2006) [156].	Survivability	16		Accuracy	Precision	Recall	-	151 886
				NB	0.8450	0.8800	0.9300	-	
				ANN	0.8650	0.8700	0.9700	-	
				C4.5	0.8670	0.8800	0.9600	-	
8.	Prediction of Breast Cancer Survival Through Knowledge Discovery in Databases. Lotfnezhad, *et. al* (2015) [150].	Survivability	18		Accuracy	Sensitivity	Specificity	-	22 763
				SVM	**0.9670**	**0.9770**	**0.9560**	-	
				BN	0.8390	0.8180	0.8610	-	
				CHAID	0.8240	0.8220	0.8270	-	
9.	Reproducible Survival Prediction with SEER Cancer Data. Hegselmann, *et. al* (2018) [149].	Survivability	57		Accuracy	F1 Score	AUC	-	248 751
				BR	0.8724	0.9318	0.5000	-	
				LR	0.9066	0.9481	0.8808	-	
				LR 1-N ENC	0.9136	0.9518	0.9023	-	
				MLP	0.9131	0.9516	0.9002	-	
				MLP 1-N ENC	0.9130	0.9517	0.9039	-	
				MLPEmb 1-N ENC	**0.9134**	**0.9517**	**0.9062**	**-**	
10.	A support vector machine-based ensemble algorithm for breast cancer diagnosis. Wang, *et. al* (2018) [157].	Tumour classification	14		Accuracy	Sensitivity	Specificity	AUC	82 707
				WAUCE	0.7642	0.7280	0.8002	0.7641	
11.	An improved survivability prognosis of breast cancer by using sampling and feature selection technique to solve imbalanced patient classification data. Wang, *et. al* (2013) [158].	Survivability	9		Accuracy	Sensitivity	Specificity	AUC	215 221
				SMOTE DT	0.7910	0.4750	0.8230	0.7000	
				SMOTE LR	0.7590	0.6450	0.7710	0.7830	
				CSC DT	0.7720	0.6690	0.7920	0.7580	
				CSC LR	0.7520	0.7520	0.7520	0.8290	
				DT	0.9120	0.1400	0.9910	0.7720	
				LR	0.9130	0.1560	0.9900	0.8290	
12.	Predicting breast cancer survivability: a comparison of three data mining methods. Delen, *et. al* (2005) [73].	Survivability			Accuracy	Sensitivity	Specificity	-	202 932
				NN	0.9121	0.9437	0.8748	-	
				DT (C5)	**0.9362**	**0.9602**	**0.9066**	-	
				LR	0.8920	0.9017	0.8786	-	

Random Forest (RF); Neural Network (NN); Decision Tree (DT); Logistic Regression (LR); Semi-supervised learning (SSL); Support Vector Machine (SVM); Naïve Bayes (NB); Bayes Net (BN); Base Rate (BR).

3.2 Data pre-processing

Careful and precise pre-processing is an essential part of data analysis, requiring an understanding of data type and distribution, the application of suitable transformations, ways of handling skewed and missing data, outlier analysis, and dimensional reduction to reduce the computation time and increase the accuracy of prognosis.

In this research 1973-2015 SEER database release was used following the correct access procedures. Data passes through several stages until a reduced set of data without sacrificing its characteristics is obtained. Firstly, using SEER Stat software, all breast cancer cases from 2004 to 2014 were selected. Secondly, referring to SEER record description documentation and analysing other literature number of variables [9], [73], [159]–[162] were iteratively decreased. As a result, a set of 21 variables was obtained in text format for around 809397 cases. Here 20 variables were independent, and “survivability” was a dependent variable. Next, text file was loaded into R environment and appropriate script was written to remove “male” cases and preserve cases with “Alive” OR “Breast” for cause of death variable. Then, for the tumour size variable, following assignments were recoded "989 = 1000; 990 = 0.9; 991 = 5; 992 = 15; 993 = 25; 994 = 35; 995 = 45; 996 = NA; 997 = NA; 998 = NA; 999 = NA; 888 = NA". Later, 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. Finally, we removed missing values and convert variables into appropriate type. As a result, we finally obtained 659802 samples with 19 variables which are presented in a Table 3.3. The dependent variable was almost balanced with 43% negative and 57% positive classes.

Table 3.3. 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 tumour was first diagnosed	Factor
7	Behaviour code	In situ or malignant	Factor
8	Primary site	This data item identifies the site in which the primary tumour 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 tumour 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	Tumour size	Tumour 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 month	Survival moth	Factor

3.3 Factorial Analysis of Mixed Data

3.3.1 Methodology

In this research we present Factorial Analysis of Mixed Data (FAMD) method, which is adapted for heterogeneous variables and is combination of Principal Component Analysis (PCA) and Multiple Correspondence Analysis (MCA) methods. These methods are briefly described here.

Among all projection-based methods PCA is well known and widely used by practitioners. It is a multivariate method implemented to reduce number of quantitative variables [163], [164]. Reduced variables are called principal components, which hold the most possible variance and are linear combinations of original variables [165]. Also, principal components are not correlated. First principal component captures highest variance and others follow in decreasing sequential order with remaining variance. Moreover, they are orthogonal with each other. However, PCA is suitable only for numeric values and in case of nominal and ordinal values are not appropriate.

A Multiple Correspondence analysis is a multivariate extension of Correspondence Analysis designed for qualitative variables. Before finding linear combinations that summarize data, categorical variables are converted into complete disjunctive table, where each category is coded as a binary variable [38], [166]. Moreover, MCA can also be considered as a generalization of PCA when the variables to be analysed are categorical instead of quantitative. Therefore, it is important to know that two methods are distinct, but belong to the same group.

The simultaneous processing of continuous and categorical data is common in factorial analysis. One method that handles this issue is Factor Analysis of Mixed Data (FAMD), which aggregates PCA and MCA to process mixed data [38]. More specifically, the quantitative variables are scaled to unit variance. Then, qualitative variables are encoded into a binary table and after scaled using the particular scaling of MCA. This specific standardization balances both continuous and categorical variables and ensures equal contribution in the analysis. The novelty of this method is in appliance of appropriate standardization process to categorical variables and combination of PCA and MCA methods.

After projecting data into a new space and choosing appropriate number of principal components, reduced data is provided to the classifiers as input. Two popular classifiers have been chosen to be used in the present study owing to the concept of no free lunch theorem. This

theorem ensures that there is no best single method and all methods perform similar on average [167].

3.3.2 Evaluation metrics

Single metric is not always appropriate; hence we are evaluating predictability of methods using five performance metrics, namely, accuracy, sensitivity, specificity, precision and F1 score. The metrics used are defined as shown below:

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

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

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

$$\text{Precision} = TP / (TP + FP)$$

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

3.3.3 Results and discussion

The reduced data was obtained through applying the FAMD algorithm on pre-processed data. The key factor in implementing this algorithm is to choose the number of principal components which holds maximum variance. A Kaiser Rule procedure is used to define the number of principal components [168]. It works on the basis of observation of the correlation matrix eigenvalues. Kaiser advices that only eigenvalues at least equal to one are kept [168], [169]. The results of our model are shown in Figure 3.1 and Table 3.4. The number of principal component was chosen as 159, which is based on Kaiser's Rule. In addition this number holds 55% of the total variance. Next, the reduced data were randomly divided into train and test in a proportion of 80% and 20% respectively. The train sample is fitted into penalized logistic regression and neural network. The classifiers were trained in R environment using "caret", "nnet" and "plr" packages. The best model for each classifier was chosen based on optimal hyper-parameters, which are L2 penalty, complexity parameter for penalized logistic

regression and number of hidden units, decay for neural networks. The values were obtained through 10 fold cross validation method. Further, a generalization capability of models was evaluated on test samples. Each model was implemented 10 times and the average results are shown in Table 3.5. Accuracy of 0.9323 for logistic regression and 0.9530 for neural network are promising and comparable to the results stated in the literature.

The accuracy (0.9530) and sensitivity (0.9444) of the Neural Network model, are slightly less than the accuracy (0.9670) and sensitivity (0.9770) of the benchmark model given by H. Lotfnezhad *et al.* [150] after implementing the SVM model. However, it is important to note that the database used had a sample size of 22 768, which is middle sized data. On the contrary, the Neural Network model was trained on a much larger database with sample size 659 802. In addition, the specificity of the Neural Network model is found better compared to the benchmark model. It is important to note here that the SVM models do not perform well on large scale data [170], [171].

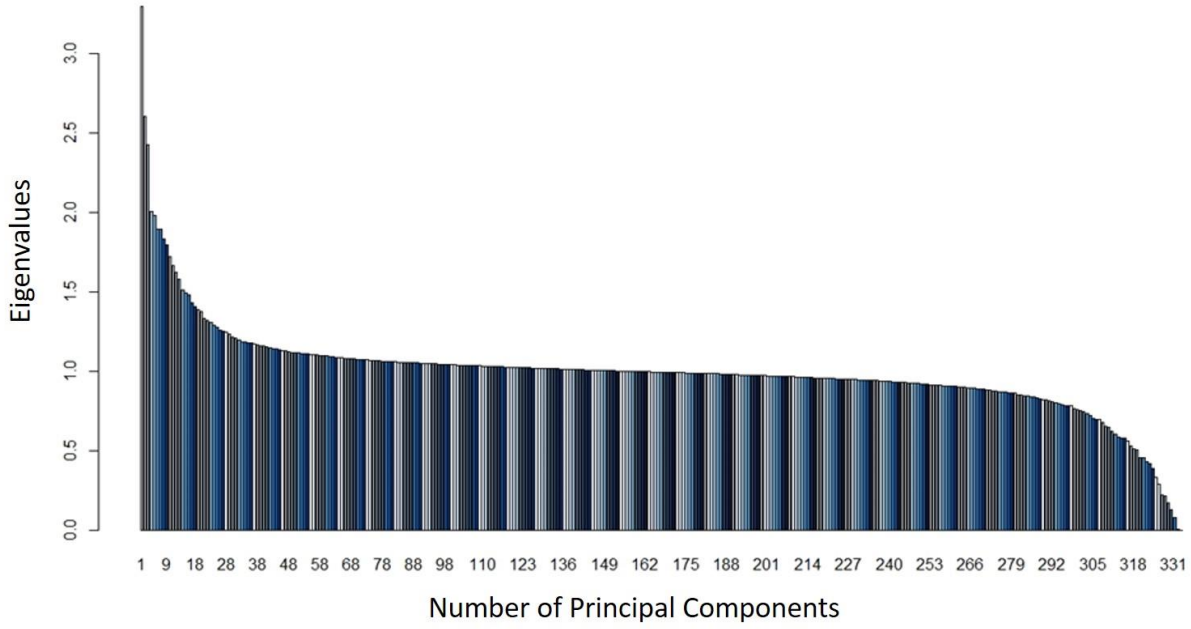


Figure 3.1. Eigenvalues and principal components.

Nevertheless, the Neural Network based model achieved higher accuracy (0.9530) and specificity (0.9592) compared to the second benchmark model, the Decision Tree model (accuracy as 0.9362 and specificity as 0.9066) proposed by Delen *et al.* [73]. The performance metrics of second benchmark model was achieved on big data with sample size 202 932 and therefore its results are more suitable to compare with results of Neural Network model in this research.

Table 3.4. Eigenvalues and percentage of variance.

Principal Component	**Eigenvalue**	**Percentage of variance**	**Cumulative percentage**
1	3.2935	0.9890	0.9890
2	2.6014	0.7812	1.7702
3	2.4259	0.7285	2.4987
4	2.0059	0.6023	3.1011
5	1.9810	0.5948	3.6960
6	1.8969	0.5696	4.2657
7	1.8922	0.5682	4.8339
8	1.8342	0.5508	5.3847
9	1.7949	0.5390	5.9238
10	1.7247	0.5179	6.4417
158	1.0008	0.3005	55.5822
159	**1.0005**	**0.3004**	**55.8827**
160	0.9998	0.3002	56.1830
331	0.0820	0.0246	99.9977
332	0.0064	0.0019	99.9997
333	0.0009	0.0002	100.0000

Table 3.5. Results of classifiers and their comparison with the benchmark results from the literature.

	Logistic Regression	**Neural Network**	**SVM** (Lotfnezhad, *et al.* (2015) [150])	**Decision Tree** (Delen, *et al.* (2005) [73])
Accuracy	0.9323	0.9530	0.9670	0.9362
Sensitivity	0.9418	0.9444	0.9770	0.9602
Specificity	0.9254	0.9592	0.9560	0.9066
Precision	0.9013	0.9438	-	-
F1 score	0.9211	0.9439	-	-

3.3.4 Summary

We have applied a projection-based method named FAMD on heterogeneous data. As a result we obtained reduced data with maximum variance. Next, the reduced data is fitted to classifiers. The Neural Network model with dimension reduction achieved the accuracy of 0.9530 that is comparable to the recent methods in literature. This paper clearly indicates that the results exhibit potential of the application of FAMD algorithm on breast cancer survivability prediction. The contribution of study is an application of the FAMD algorithm as a dimensionality reduction method for breast cancer survivability on large scale data, which is not done before.

3.4 Entity Embedding

3.4.1 Methodology

The goal of the entity embedding in neural networks is to map an each entity of categorical variable to some continuous vector representation of defined dimensions. This is equivalent to representing categorical variable using one hot encoding then multiplying with weights of an extra layer of linear neurons, which is embedding layer. The weights of embedding layer are learned as other parameters of neural network. Next, transformed categorical variables together with continuous variables act as an input layer for other layers.

The entire neural network model is trained using backpropagation algorithm. The mathematical and other details of entity embedding is clearly described in this manuscript [44].

In this research we developed an Entity Embedding Neural Networks (EENNs) model to predict breast cancer survivability. Figure 3.2 shows the model's architecture. It comprises an input layer, an encoding layer, an embedding layer, a dense layer, a batch normalisation layer and an output layer. The input layer is made up of categorical and continuous variables, where continuous variables have a full and direct connection to the dense layer but categorical variables are first transformed as already described and the embedding layer's outputs are fully connected to the dense layer. The connection of neurons between dense layers and batch normalization layers is unidirectional, and there is full connection between the batch normalization layer and the dense layer and between the batch normalization and output layers.

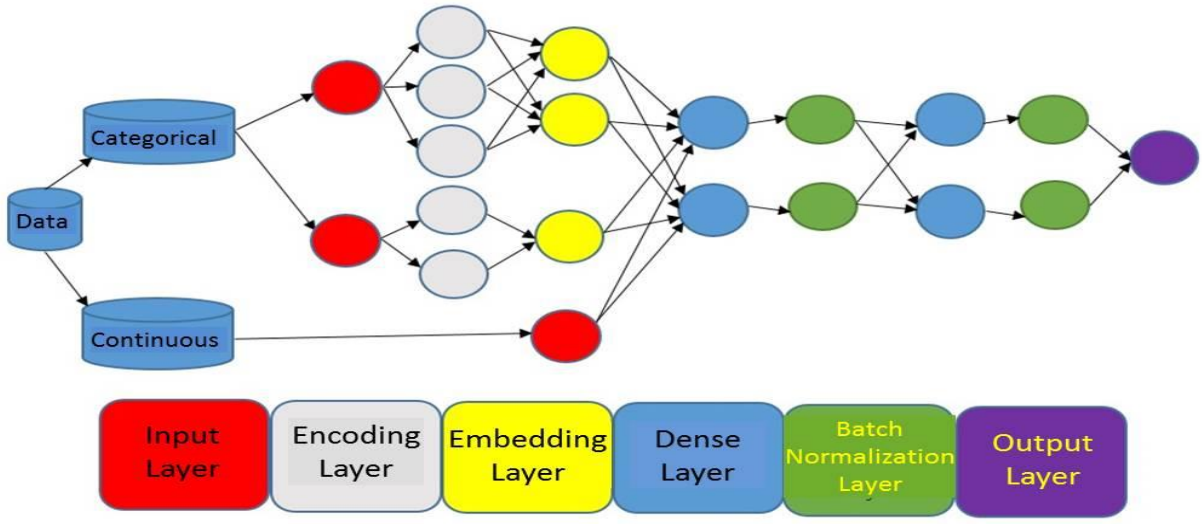


Figure 3.2. Architecture of EENNs model [172].

Table 3.6. Parameters of EENNs model.

Activation function for embedding layer	Linear
Number of neurons in embedding layer	89
Number of dense layers	2
Number of neurons in each hidden layer	832
Activation function for dense layer	ELU

Learning rate	0.3277
Regularization method	L2
Weights initialization method	Uniform
Mini-batch size	128
Training epoch	20
Normalization	Batch
Activation function for output layer	Sigmoid

Four parameters such as activation function for dense layer, number of layers in dense layer, number of neurons in dense and learning rate were chosen for optimization. These parameters were optimized using a grid search algorithm. The chosen four parameters and other specifications of our model are presented in Table 3.6. The prediction of survivability in our case is binary classification problem; therefore we have chosen sigmoid activation function for output layer.

3.4.2 Evaluation metrics

In order to test robustness of the prognosis prediction apart from interpretability and accuracy five vital performance metrics have been chosen. The chosen metrics are accuracy, sensitivity, specificity, AUROC curve and F1 score. The metrics used are defined as shown below:

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

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

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

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

AUROC curve plots the sensitivity against one minus specificity. In addition, the terms AUROC and AUC are used interchangeably in a literature.

3.4.3 Comparison with other classifiers

In this work we compared our entity embedding neural network classifier with widely used classification algorithms such as Logistic Regression, Support Vector Machines, Random Forest and Gradient Boosting. The 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 parameters, which are presented in Table 3.7.

Table 3.7. Parameters of other Classifiers [172].

Logistic Regression	
Coefficient of regularization	0.8
SVMs (Linear Kernel)	
Coefficient of regularization	0.8
Random Forest	
The number of trees	200
The maximum depth of tree	10
The minimum number of samples required to split an internal node	2
Gradient Boosting	
Coefficient of regularization	0.4
The maximum depth of tree	5
The minimum sum of instance weight needed in a child	4
Subsample ratio of the training instances	0.8
Subsample ratio of columns when constructing each tree	0.3

The classifiers are trained in two different ways. In first case, we fitted data in original form without mapping into multidimensional vector space. However, in second case the numerical representations of categorical variables extracted using our approach and continuous variables are further used as an input for classifiers. In addition, all classifiers are regularized in order to avoid overfitting. Identical test samples are used to measure generalization

capabilities of classifiers. Logistic Regression, SVMs, Random Forest were implemented using scikit-learn package and Gradient Boosting was developed using xgboost package.

3.4.4 Results and discussions

The proposed approach as a classifier has been compared with other classification algorithms. One of the main contribution of the work was to show power of entity embedding approach for mapping categorical variables and the results presented in Table 3.8 and Table 3.9 are promising. Prognosis prediction results from various classifiers are also illustrated in Figure 3.3 and Figure 3.4. Performance metrics namely, accuracy (Acc.), Sensitivity (Sn),, specificity (Sp), F1 score and AUC values have been plotted using bar charts. Comparison between classifiers on train data samples and test data samples have been plotted and the results are shown in Figure 3.3 and Figure 3.4. As shown in tables' and figures' accuracy for all classifiers are better in case of entity embedding approach. In addition, it is important to note that in overall all models performance metrics on test results with entity embedding are better than benchmark models from literature Table 3.10. Moreover, results with embedding from this research outperformed the results from S. Hegselmann and others work in terms of accuracy and AUC, where they also implemented the idea of embedding.

Table 3.8. Comparison of classifiers without Entity Embedding.

		Logistic Regression	SVM	NN	Random Forest	Gradient Boosting
Accuracy (Acc.)	Train	0.9654	0.9282	0.9528	0.9713	0.9671
	Test	0.9647	0.9270	0.9523	0.9706	0.9667
Sensitivity (Sn)	Train	0.9718	0.9823	0.9756	0.9709	0.9844
	Test	0.9717	0.9820	0.9758	0.9706	0.9842
Specificity (Sp)	Train	0.9606	0.8883	0.9649	0.9716	0.9543
	Test	0.9594	0.8860	0.9639	0.9706	0.9535
F1 Score	Train	0.9598	0.9210	0.9646	0.9664	0.9622
	Test	0.9592	0.9201	0.9641	0.9658	0.9619
AUC	Train	0.9662	0.9353	0.9703	0.9712	0.9693
	Test	0.9656	0.9340	0.9698	0.9706	0.9689

Table 3.9. Comparison of classifiers with Entity Embedding [172].

		Logistic Regression	SVM	NN	Random Forest	Gradient Boosting
Accuracy (Acc.)	Train	0.9701	0.9704	0.9645	0.9713	0.9710
	Test	0.9699	0.9702	0.9640	0.9707	0.9708
Sensitivity (Sn)	Train	0.9718	0.9675	0.9730	0.9790	0.9769
	Test	0.9719	0.9677	0.9734	0.9787	0.9771
Specificity (Sp)	Train	0.9689	0.9725	0.9698	0.9656	0.9667
	Test	0.9685	0.9720	0.9692	0.9647	0.9661
F1 Score	Train	0.9615	0.9653	0.9663	0.9667	0.9663
	Test	0.9651	0.9652	0.9663	0.9662	0.9662
AUC	Train	0.9704	0.9700	0.9714	0.9723	0.9718
	Test	0.9702	0.9698	0.9713	0.9717	0.9716

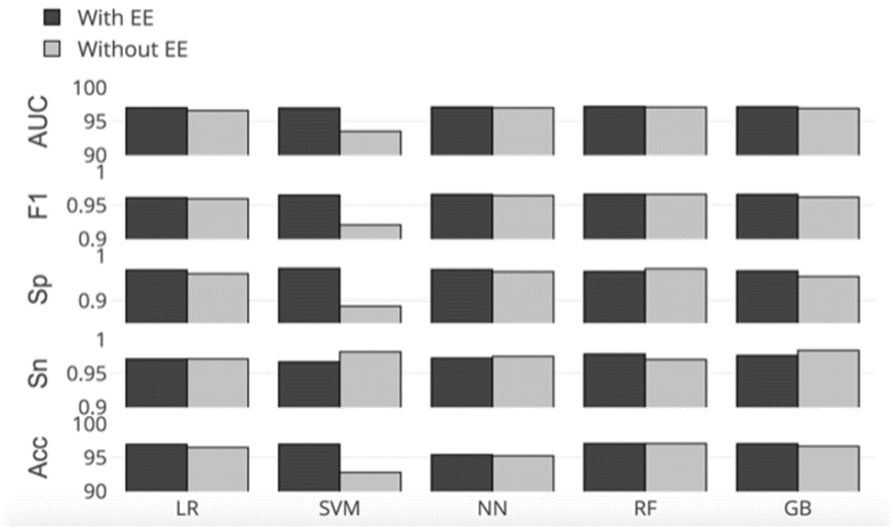


Figure 3.3. Comparison of classifiers for train sample with entity embedding (EE) and without entity embedding.

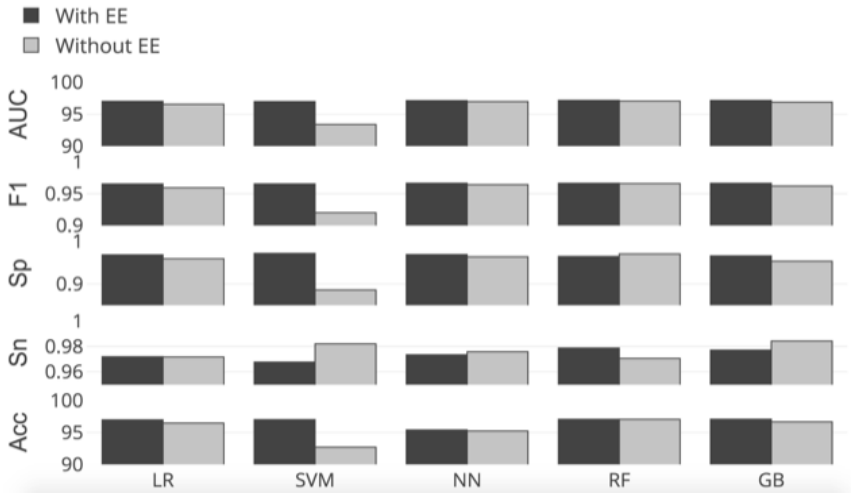


Figure 3.4. Comparison of classifiers for test sample with entity embedding (EE) and without entity embedding.

Table 3.10. Results of classifiers and their comparison with the benchmark results from the literature.

	Logistic Regression (EE)	**SVM (EE)**	**NN (EE)**	**Random Forest (EE)**	**Gradient Boosting (EE)**	**MLPEmb 1-N ENC** [149]	**SVM** [150]	**Decision Tree** [73]
Accuracy	0.9699	0.9702	0.9640	0.9701	**0.9708**	0.9134	0.9670	0.9362
Sensitivity	0.9719	0.9677	0.9734	0.9787	**0.9771**	-	0.9770	0.9602
Specificity	0.9685	**0.9720**	0.9692	0.9647	0.9661	-	0.9560	0.9066
F1 Score	0.9651	0.9652	**0.9663**	0.9662	0.9662	0.9517	-	-
AUC	0.9702	0.9698	0.9713	**0.9717**	0.9716	0.9062	-	-

3.4.5 Summary

We have transformed heterogeneous variables into numerical representations using an entity embedding neural network model approach. Next, numerical representations are fitted into different methods as an input for classification of breast cancer survivability. The majority of the classifier achieved better results with the entity embedding and it is better than recent results in literature. This paper clearly demonstrates that the results exhibit potential of the application of the entity embedding approach on breast cancer survivability prediction. The contributions of study are application of the entity embedding as a dimensionality reduction method for the breast cancer survivability.

4 Neuroevolution

Until recently a network training has focused on improving its performance by reducing network errors in mapping a given system's outputs and inputs. This focus may lack robustness as well as result in overfitting of the training data [173]. A network performance in training should also be evaluated with regard to such metrics as precision, recall/sensitivity, and F1 score. When a network is evaluated against more than two metrics for performance, training becomes a multi-objective optimization problem. During the evolution-by-selection stage, the generated network population is evaluated for the performance metrics such as accuracy, etc. Neuroevolution is an approach based on artificial intelligence which uses evolutionary algorithms to optimize the neural networks (NNs) to give optimized network parameters and configuration. Multi-objective neural network optimization in recent years has been performed by the popular evolutionary algorithm NSGA-II to minimize perceptron errors and network complexity [174], but use of EAs in neural networks optimization has been limited to network parameter variations such as network architecture, number of neurons and the connection weights parameters [174]–[178]. Investigations from literature have not found any multi-objective optimization of DNNs with simultaneous consideration of numerous network parameters and hyper-parameters. In this research, the advanced version of NSGA-II, [179] NSGA-III [110], [111], [180], has been used for multi-objective optimization of DNNs with selected hyper-parameters being learning rate, number of layers, number of neurons in each layer, choice of activation function, and number of iterations. The network's hyper-parameters are optimized using the evolutionary algorithm NSGA III, with Levenberg-Marquardt optimization used to obtain parameters such as weights and biases between layers.

4.1 Neuroevolution using NSGA-III algorithm

When encountered with multiple objectives, the concept of minimization and maximization is replaced by the search for a set of trade-off solutions known as a Pareto optimal solution set. In the course of using evolutionary algorithms to this end, genotypes in the form of population of solutions are generated randomly and evolved by means of genetic operators such as selection, crossover, mutation etc. "Offspring" is the term used to describe a population which is evolved during the simulations of the algorithm. Competing solutions are compared by means of the fitness values evaluated based on their performance metric values using a non-domination criterion. In this way the solutions, with performance metric values not dominated by other solutions in the population, are selected. NSGA-III is an extension to the earlier NSGA-II algorithm and addresses *multi-objective optimization problems.* Generic phases in which NSGA-III evolves deep neural networks are given below [180]:

I. Initialize a parent population (Pi) of N deep neural networks genotypes encoded with chosen hyper-parameters. Obtain an offspring population (Qi) by applying genetic operators such as mutation and crossover to (Pi);

II. Combine the two populations (Pi & Qi) to form a mating-pool of 2N solutions and evaluate using the training data on the basis of three performance metrics. Later, a non-domination fitness index F is determined for every member of the population of solutions based on performance metrics;

III. To select an optimal set of N solutions from the combined population, sort the population R according to non-domination levels (F1, F2, F3 etc.);

IV. Thereafter, create a new population Si by selecting one individual from each non-domination level starting from F1 until the size of Si is equal to or bigger than N;

V. Combine Si with the preceding parent population and repeat steps 2-5 for a given number of generations (this is another hyper-parameter chosen at the start of the algorithm) till the set performance/termination is satisfied by the networks.

A major advantage of DNNs optimization by evolutionary algorithms is the ability to generate powerful synergy by combining evolution with learning. Gradient-based learning algorithms are known to be normally sensitive to the initial hyper-parameter value set which significantly affects the network performance by providing sub-optimal parameter values. Evolutionary algorithms through optimization are able to identify set of hyper-parameters, but during each DNNs evolutionary algorithm epoch such parameters as network weights and biases are obtained using a regular backpropagation scheme.

4.2 Design of Experimental and implementation

Sub-chapters 3.1 and 3.2 describe the databases and selected variables for data analytics. To conduct experiments, the data is divided randomly into train (90%) and test (10%) after which the train sample is subjected to a ten-fold cross validation, a normal practice when working with cancer data [181].Nine subsets are selected for training during each epoch with the remaining one held back for validation later. The test sample is also used to test the model's generalization capability. To prevent classifiers being overfitted, L2 regularisation is also implemented and then the DNN models are optimized in a TensorFlow environment, which is an open source library of machine learning. The simulations took place on a workstation with the configuration as: Intel Xeon Platinum 2.10 GHz 48 Cores, NVIDIA Quadro P6000 24 GB, 512 GB DDR4.

A total of 1,000 DNN models with varied hyper parameters of neuroevolution were initialised and optimized using NSGAIII. Table 4.1 lists the hyper parameters used for the initial population.

Table 4.1. Initial hyper parameters of DNNs [172].

Description	Value
Number of neurons in each hidden layer	1-1000
Number of layers	1-10
Learning rate	0.001-1
Types of activation functions used for hidden layers	ReLU, Sigmoid, Linear, ELU, SELU

To begin with, a set of 1,000 DNN models is generated using random network parameters with 4 numbers of predefined hyper parameters within the given ranges (Table 4.1). Network weights are initialised randomly, momentum index is chosen as 0.9 and sigmoidal gain and threshold values are initialised as one.

Linear activation is chosen for the initialised models' input layer. Since survivability prediction is a classification problem of binary nature, we chose sigmoidal transfer function for the output layer. During simulation experiments, the parameters given in Table 4.2 were implemented for NSGA III [57], [111]:

Table 4.2. Simulation parameters for NSGA-III [172].

Description	Value
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

After implementing NSGA-III, a Pareto optimal set of non-dominated DNN models were obtained. Three metrics, chosen to evaluate performances, are shown in Figure 4.1 for the first front solutions of DNNs, the metrics are namely, F1 score, Accuracy, and AUC. Although the DNN models on the first non-dominated front are equally better, users need only one DNN model to finally select and perform prognosis prediction. Selecting a final DNN model from a set of Pareto DNNs requires a strategy drawn from the realms of decision theory.

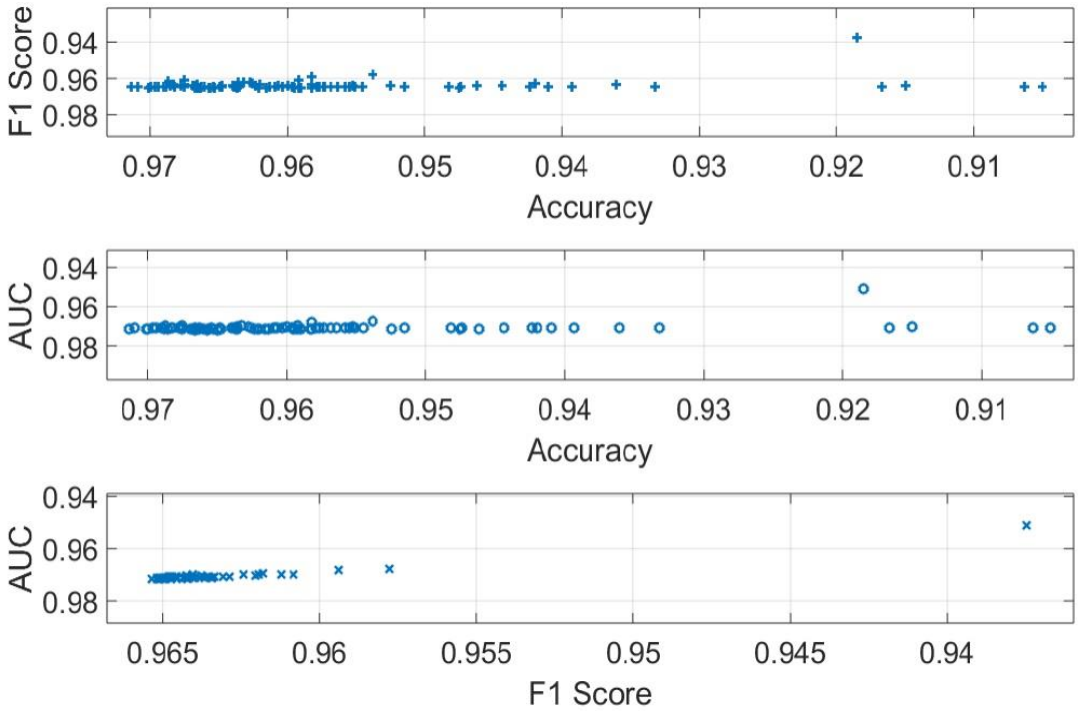


Figure 4.1. Final set of Pareto optimal solutions and their performance metrics [172].

Decreasing error from the network alone may not enhance performance sufficiently and may not be robust enough but may result in over-fitting of the training data. Five important performance metrics were selected to test the prognosis prediction's robustness: *F1 score, AUROC curve, accuracy, sensitivity, and specificity*. Sensitivity measures a test's ability to detect a condition when it is present, while specificity measures a test's ability to exclude the condition correctly when it is not present. High sensitivity tests usually have low specificity, and therefore these two objectives conflict. Precision is the ratio of correctly predicted positive observations to the total of predicted positive observations and indicates how many positive observations were actually found. The *F1 Score* is the weighted average of *Precision* and *Recall*. The process begins, however, with a confusion matrix to test a model's accuracy and correctness using the metrics: *TP (True Positives), TN (True Negatives), FP (False Positives), and FN (False Negatives).*

$$Accuracy = (TN + TP) / (TP + FP + FN + TN)$$

$$Sensitivity\ (Recall) = TP / (TP + TN)$$

Specificity = TN / (TN + FP)

*F1 score = (2 * Precision * Recall) / (Precision + Recall)*

The *AUROC curve*, is defined as the area under the ROC curve, which is plotted between sensitivity and one minus specificity. *AUROC* and *AUC* are the synonymous terms used in the literature [177].

4.3 Fuzzification and fuzzy inference

A practical implementation requires a single best DNN model solution or single Pareto point giving the best compromise DNN model. This places on the user a heavy cognitive burden and a strategy is needed to help make this vital decision. Past research has suggested the use of the min-max approach [110], [182], but the method proposed in this research is fuzzy inference-based and can provide a single metric that logically combines a number of performance metrics. The steps to implement this method are described below.

4.3.1 Fuzzy metrics

The metrics used to evaluate performance are first converted into fuzzy variables using the values obtained from the NSGA-III experiments to decide for their expected ranges which are normalised between 0 and 1. Three fuzzy membership functions (MFs) are used in defining metrics as fuzzy variables: *L for Low*; *M for Medium*; and *H for High*. Geometries of MF are chosen be similar to Gaussian curve (4.1-4.3), even though in fuzzy systems they are usually trapezoidal or triangular. Gaussian distribution was chosen for this work because of the gradual change between membership functions. This step is termed as *fuzzification* of inputs.

$$L = ae^{-(f_i/\sigma_i)} \tag{4.1}$$

$$M = ae^{-\left(\frac{f_i - \frac{R_i}{2}}{\sigma_i}\right)} \tag{4.2}$$

$$H = ae^{-\left(\frac{f_i - R_i}{\sigma_i}\right)} \tag{4.3}$$

where f_i is the input performance metric values from a competing DNN model, treated as objectives to be optimized. The performance metrics range is R_i where R_i = *(max (fi)-min (fi))*. The values for constants "a" and the standard deviation "σ_i" are taken as $1/\sigma\sqrt{2\pi}$ and R_i */ 5* respectively for the three fuzzy membership functions (L, M, and H).

4.3.2 Fuzzy inference

Having defined the performance metric inputs as fuzzy variables, the fuzzy system's design is completed by devising an inference mechanism. A fuzzy system's inferencing is realized through the system's rule-base which is in essence a collection of *if* and *then* statements mapping *antecedents* or *inputs* to consequents or outputs. A common fuzzy rule-base structure is:

If f_i is L and, … … … and f_N is H then AS_i is y_i

In this case, $f_1 \ldots f_N$ are the performance metrics acting as inputs to the fuzzy system, MS_i is the Membership Score for i^{th} rule and has the numerical value y_i. The total number of rules can be derived from the number of MFs used in defining antecedent variables. For simplicity, we designed this fuzzy based system with only three (Accuracy, AUC and F1 score) of the five performance metrics. Since each of the three metrics uses three MFs (L, M and H), the total number of fuzzy rules (N_r) is 3^3 or 27, which encompasses all possible combinations of the antecedents' MFs. Numerical values assigned to the MFs are: L = 0; M = 1; H= 2. As shown in (4.4), the consequents or outputs for these rules are the sum of their component MFs' membership scores.

$$y_i = 1 + \sum_{j=1}^{3} MS_{ij} \tag{4.4}$$

Here i is the rule index (there are 27 rules) and MS_{ij} the membership score (0 for L, 1 for & 2 for H) for the j^{th} objective function in the i^{th} rule. For instance, if all objectives in a

particular rule have low (L) MFs then the output or membership score of that rule is one, while if all MFs are medium (M), the output or score is 4.

Then we follow the conventional fuzzy inferencing procedure to calculate output from a fuzzy system, whereby each rule's output is computed from the degree of fulfilment of all MFs for a given set of input performance metrics. Degrees of fulfilment are computed by plugging input (f_i) values into equations (4.1-4.3). Weights for each individual rule are calculated with equation (4.5).

$$w_i = \prod_{j=1}^{3}(L_{ij} * M_{ij} * H_{ij}) \tag{4.5}$$

A candidate DNN's overall membership score is in essence is a numerical or crisp output from the fuzzy system and is computed taking the weighted average of outputs from all the rules for given input values. The final overall membership score (OMS) can be therefore computed using (4.6):

$$Y = \frac{\sum_{i=1}^{Nr}(w_i y_j)}{\sum_{i=1}^{Nr} w_i} \tag{4.6}$$

where N_r is the count of fuzzy rules (here these are twenty seven). Inputs to the fuzzy system are the performance metric values from all the non-dominated DNN solutions obtained through NSGAIII. The outputs from fuzzy system are all recorded and analyzed. Table 4.3 shows ten representative DNN Solutions, their performance metrics and overall membership scores from Pareto optimal DNNs have been shown in, together with their performance metric values used as inputs and their final overall membership scores. Candidate design number 48 is in the end selected for the proposed DNN design thanks to its maximum fuzzy index value of ***6.7320***. Although all solutions are non-dominated and should be as good as each other, they have different fuzzy indices or OMS. It should also be noted that the proposed method of fuzzy ranking discriminates better among candidate solutions and a solution with maximum fuzzy

index is eventually identified as better than the other DNN models. Table 4.4 shows the selected DNN model's parameters.

Table 4.3. Representative DNN solutions from the Pareto optimal set shown with their overall membership scores calculated from performance metric values [172].

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	6.7176
48	**0.97130**	**0.96489**	**0.97211**	**6.7320**
49	0.96155	0.96508	0.97150	6.7312
50	0.96474	0.96398	0.97102	6.7235
51	0.93932	0.96483	0.97105	6.7217
52	0.91501	0.96401	0.97010	6.7213
53	0.96207	0.96495	0.97112	6.7213
54	0.95451	0.96476	0.97086	6.7118

Table 4.4. Final DNN model and its parameters [172].

Activation function for input layers	Linear
Number of neurons in input layer	89
Number of hidden layers	4
Number of neurons in each hidden layer	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

4.4 Simulations with existing classifiers

After acquiring hyper-parameters from the finally selected DNN model (employing NSGA-III), it is compared with that of other classifiers using performance indices. This work has used popular classification algorithms such as Gradient Boosting Logistic Regression, Random Forest, and Support Vector Machines, and Table 4.5 lists the parameters used for these classifiers. Breast cancer data is sampled into train (90%) and test (10%) and a ten-fold cross validation is performed on the train sample to find the optimal model parameters.

All classifiers are regularized to avoid overfitting, and their capacity for generalization is measured using test samples. Logistic Regression, SVMs, Random Forest were implemented from the scikit-learn package and Gradient Boosting was developed using the XGBoost package.

Table 4.5. Values of parameters used for other classifiers [172].

Coefficients of regularization	0.8
Coefficients of regularization	0.8
Number of trees	200
Maximum depth of tree	10
Minimum number of samples to split an internal node	2
Coefficient of regularization	0.4
Maximum depth of tree	5
Minimum sum of instances weight needed in a child	4
Subsample ratio of the training instances	0.8
Subsample ratio of columns when constructing each tree	0.3

4.5 Results and discussions

Using NSGA-III to successfully optimize DNN generates a Pareto optimal set of DNNs. Table 4.3 lists representative DNN solutions together with their performance metrics. The fuzzy inference method used to choose the best from the Pareto optimal DNNs solutions which is shown in Table 4.3 to be the 48th solution since that has the highest fuzzy overall membership score and scores are high for accuracy and AUC as well as having a comparable F1 score. Table 4.4 shows the hyper-parameter set for this DNNs. Analysis shows the OMS to be sensitive to the hidden layers' membership function. The OMS values returned by Sigmoid and RELU functions were better than those of other functions. In the same way, better OMS can be obtained with fewer neurons in the hidden layers and for this case there should be no more than four layers. Other implemented algorithms included Logistic Regression and Support Vector Machines which were trained using both categorical and continuous variables from the SEER database. Three performance metrics were used to compare results from these

classifiers with the final DNN model selected, and Figures 4.2 and 4.3 show the train and test results. Moreover, final DNN model outperformed the benchmark model from the literature Table 4.6.

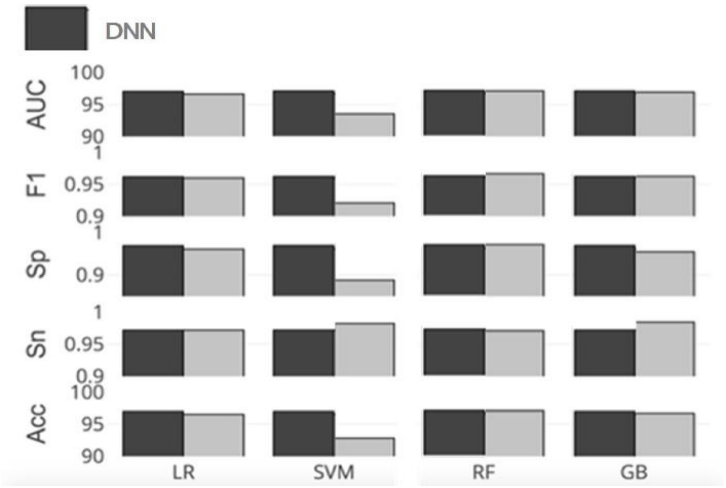


Figure 4.2. Proposed DNN compared with other classifiers on train samples [172].

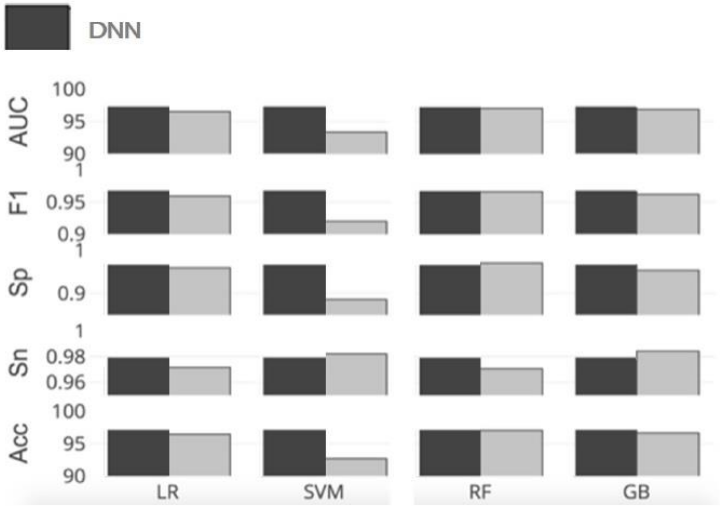


Figure 4.3. Proposed DNN compared with other classifiers on test samples [172].

Table 4.6. Comparison of final DNN solution with the benchmark model.

	Final DNN (NSGA III)	**MLPEmb 1-N ENC (**S. Hegselmann, et al. (2018) [149]**)**
Accuracy	0.9713	0.9134
F1 Score	0.9649	0.9517
AUC	0.9721	0.9062

4.6 Summary

Prognostic modelling of breast cancer must be both very accurate and easy to interpret. Neuroevolution is proposed with NSGA III for optimization of DNNs and to provide hyper-parameters of neuroevolution. This optimization resulted in a set of Pareto optimal DNN models, all of which were equally good breast cancer prognosis candidate solutions, but it remains necessary to select one from the Pareto set; in this case, selection was done using a novel fuzzy inference approach. Hyper parameters are DNN's building blocks and scanning the results makes possible an intuitive grasp of the relationship between them and a number of performance metrics, making it possible to achieve specific performance metric by modification of the hyper parameters. Analysing the Pareto optimal DNN solutions with reference to their hyper parameters and correlating with the performance metrics, it is possible to extract more information about the mechanism or working of DNN models. This is expected to enhance DNNs models' transparency and interpretability and may make it possible to perform training experiments more efficiently and effectively, while greater transparency in DNN models will also make them more acceptable to medical practitioners.

During the experiments, the ability of the evolutionary algorithm (NSGAIII) to handle many performance criteria simultaneously was found to be questionable, with the algorithm frequently converging prematurely with production of a false or pseudo Pareto optimal front. It is therefore suggested that NSGAIII's selection operator be modified. Analogies between hyper-parameters and the model's performance metrics should also be established as a future direction for this research.

5 Fuzzy Inspired Evolutionary Algorithm

An analytical sorting method based on fuzzy dominance as an advance on extant multi-objective evolutionary algorithms (MOEAs) is presented in this Chapter. The design of their sorting schemes is such that these EAs may not clearly discriminate between solutions while solving optimization problems with many objectives. These algorithms have been also criticized for ambiguous termination criteria and inability in selecting a singular solution from the Pareto optimal solution set. **Fuzzy Inspired Evolutionary Algorithm (FIEA)**, is proposed as an alternative to address these issues. In this algorithm, performance metrics are defined as qualitative metrics in the form of fuzzy functions and the candidate solutions are given an **overall membership score (OMS)** on the basis of their respective fuzzy metrics values, after which the OMS is used to rank these solutions to improve the sorting process. Benchmark optimization case studies are optimized by means of the proposed algorithm and three other commonly used methods, generating performance indices to allow various aspects of the proposed algorithm to be evaluated and compared with existing methods. This proposed algorithm can also help selection of the final solution from the Pareto optimal solution set.

5.1 Fuzzy systems and evolutionary algorithms

Lotfi Zadeh's seminal paper [183] described fuzzy set theory, in which conventional mathematical operators can treat qualitative numbers. Fuzzy logic is now popular as a metaheuristic approach in learning non-linear, undefined and ambiguous systems [184]. NSGA-II and other evolutionary optimization methods have been used in optimizing fuzzy systems by providing optimized rule-base and optimal placement of fuzzy membership functions (MFs) [185]–[188]. Previous uses of fuzzy systems included improving evolutionary optimization methods' overall performance [114], [116], [117], [189]–[191]. However, fuzzy

logic has only been used in EAs either to select an appropriate solution from the PF solution set or to guide EA convergence in line with user preference. Dominance criterion using fuzzy logic has been used recently together with MOEA/D (MOEA/DFD in the literature) [192]. Fuzzy Pareto dominance is used to discriminate between two solutions; should any solution fail to dominate; scalar decomposition is used. Offspring diversity is assured by the use of uniformly distributed weight vectors. Fuzzy logic's ability to handle both qualitative and quantitative data effectively has led to its use in solving a number of real world problems [193]–[202].

The proposed approach is implemented as follows. The performance criteria or the objectives are first converted to as variables which are fuzzy in nature and this process is termed as *Objective fuzzification*. Subsequent steps are: *consolidation of dominant solution fronts*; *fuzzy inspired sorting;* and *realization of operators* from EAs which are selection operator, crossover and mutation operators. These steps are discussed in more detail in the following sub-chapters.

5.1.1 Fuzzy Performance metrics

As are normal with EAs, a set of solutions are initialised randomly, and values of performance metrics corresponding to each of the solution in the set are calculated. Later, these metrics are converted to fuzzy values using standard procedure. Figure 5.1 shows this process in action whereby two performance metrics, ranging between 0 and 1.0 are redefined as fuzzy performance metrics represented by means of four Gaussian membership functions (MFs), though triangular, trapezoidal or other shapes can be used if preferred.

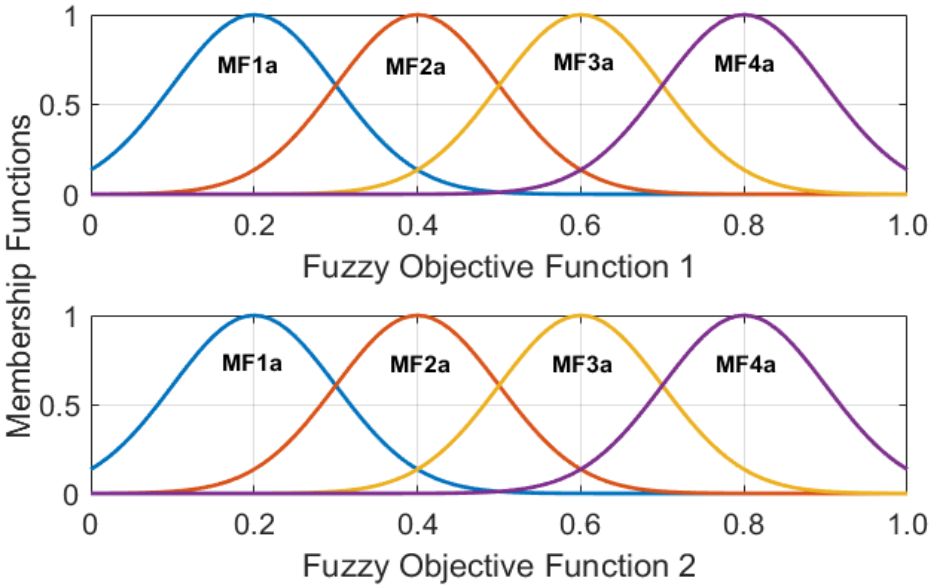


Figure 5.1. Fuzzy Membership functions and their placement for two example problem involving objectives functions.

The number of MFs, their shapes and their positions are subsequently decided on. There can be two or more MFs associated with each objective function, depending on the accuracy required in the results. Other parameters namely, centres of the membership functions (the MF centre points shown as A, B, C and D in Figure 5.2) and standard deviations (σ) are computed during the course of the algorithm using eq. (5.1-5.3). The ranges and the extreme values of the performance metrics are found from the population of solutions and are used to calculate MF parameters for equations (5.1-5.3).

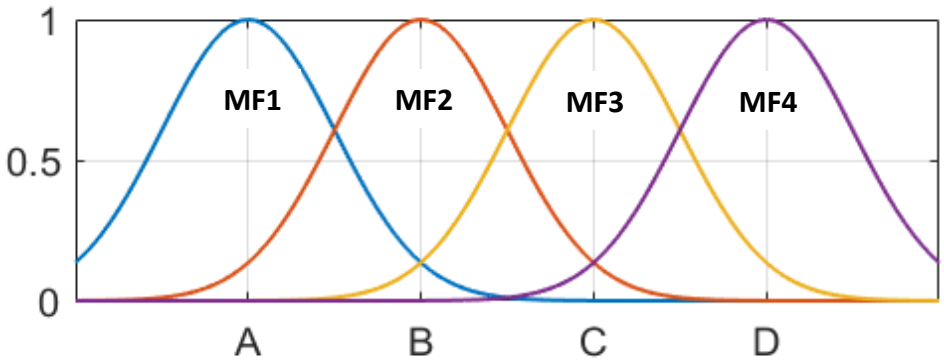


Figure 5.2. Automatic generation of MF parameters.

$$P_i = 3/4 * \min(f_i)\,;\; B_i = A_i + \frac{R_i}{(M_i - 1)}; C_i = B_i + \frac{R_i}{(M_i - 1)}; D_i = 1.25 * \max(f_i) \quad (5.1)$$

$$\sigma_i = R_i/(2M_i - 1) \quad (5.2)$$

$$R_i = (1.25 * \max(f_i) - 0.75 * \min(f_i)) \quad (5.3)$$

where R_i is the range of performance metrics values (f_i), and the standard deviation (σ_i) is defined such that the MFs encompass the whole objective function value ranges. The total number of MFs for the i^{th} performance metric is given by M_i. Each iteration dynamically updates the objectives' universe of discourse (or range) with corresponding changes to the fuzzy parameters.

5.1.2 Dominance and non-dominance criterion

The current and prevailing sorting approach based on non-dominance criterion allows selection of competing solutions that provide 'better' or 'not worse' performance metric during the selection stage. Now "worse" and "good" are linguistic variables, and so deciding their extent – that is, how much "worse" or "good" – may be impossible using quantitative numbers. Comparing objectives numerically have a negative effect on an algorithm's discrimination capabilities, resulting in an irresolute state. As this Chapter will explain, the idea or the concept of a solution's dominance over other competing ones requires qualitative and not quantitative analysis. Fuzzy inspired sorting, therefore, should rightly substitute the current idea of sorting based on the non-dominance criterion.

At the root of this approach is the fact that, if accuracy of the solution cannot be determined (and in real- life situations this is always the case), comparing the objective function values qualitatively will be more reliable than comparing quantitatively.

The non-dominance definition is ambiguous, as we are looking not for the best solutions but for solutions not dominated by others. Because of this, many solutions are accepted as non-dominating ones specifically in cases where there are many objectives. As the next sub-chapter

explains, the proposed dominance criterion which is inspired by fuzzy logic, unlike non-dominance, discriminates solutions better and places in their dominant solution fronts.

5.1.3 Dominant solution fronts

After performance metrics have been fuzzified, dominant solution fronts are created in the solution space by giving metrics' MFs membership scores (MS), which for the m^{th} membership function is:

$$MS(m) = m - 1 \tag{5.4}$$

Thus, membership scores assigned for MF1, MF2, MF3, and MF4 (Figure 5.2), are: 0, 1, 2, and 3 (See Table 5.1).

Table 5.1. Membership scores for membership functions.

	MF1	MF2	MF3	MF4
Membership Score (*MS*)	0	1	2	3

The whole solution space is then split into a *known number* of *dominant solution fronts* (Figure 5.3, [203]) calculated from the number of performance metrics and their MFs. An expression (5.5) is used which gives the number of dominant solution fronts (F_f) for a MOP with 'N' metrics and 'M_j' MFs for j^{th} performance metric.

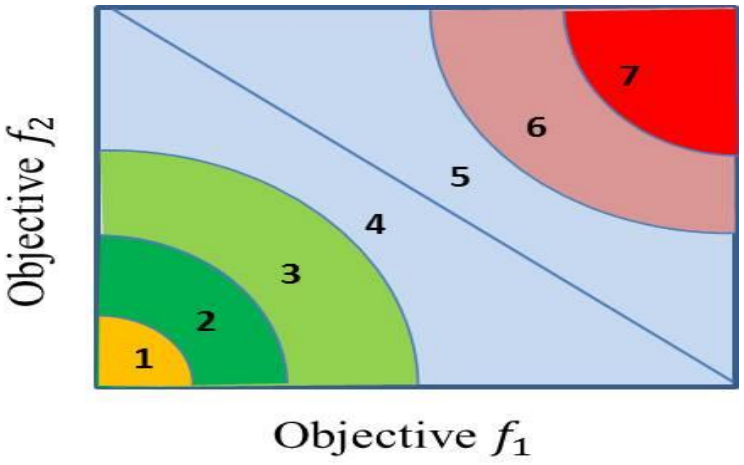


Figure 5.3. Placement of FIEA dominant fronts in the solution space.

$$F_f = 1 + \sum_{j=1}^{N}(M_j - 1) \quad (5.5)$$

Table 5.2 clarifies computation of the number of dominant solution fronts. It considers a two objective MOP in which both functions are converted to fuzzy functions using two MFs and later by three MFs (Table 5.2 Column 1). Two objective functions, each with 2 MFs, means 3 dominant solution fronts, while a problem involving two objectives each described by 3 MFs will have 5 dominant solution fronts. It follows that increasing the number of MFs increases the number of dominant fronts sharing the solution space.

Table 5.2. Collective Membership scores of MFs.

Number of MFs	Membership Functions	Membership Scores for MFs	Possible combinations of fuzzy objective values for two objectives (or fuzzy rule-base)	Combined Membership Scores (MS)	Fuzzy dominant front index (MS+1)
2	$MF1_a$ & $MF2_a$ $MF1_b$ & $MF2_b$	0 & 1 0 & 1	$MF1_a$ & $MF1_b$	0	1
			$MF1_a$ & $MF2_b$	1	2
			$MF2_a$ & $MF1_b$	1	2
			$MF2_a$ & $MF2_b$	2	3
3	$MF1_a$, $MF2_a$, $MF3_a$	0, 1 & 2	$MF1_a$ & $MF1_b$	0	1
	$MF1_b$, $MF2_b$, $MF3_b$	0, 1 & 2	$MF1_a$ & $MF2_b$	1	2
			$MF1_a$ & $MF3_b$	2	3
			$MF2_a$ & $MF1_b$	1	2
			$MF2_a$ & $MF2_b$	2	3
			$MF2_a$ & $MF3_b$	3	4
			$MF3_a$ & $MF1_b$	2	3
			$MF3_a$ & $MF2_b$	3	4
			$MF3_a$ & $MF3_b$	4	5

Fuzzy front formation is also explained in illustrations 5.3 & 5.4. In a two objective MOP, each objective has been converted to fuzzy function using four MFs. According to (5.5), the total number of dominant fronts for this problem will be 7. Figure 5.3 demonstrates the placement of these seven dominant fronts, while Figure 5.4 gives a detailed illustration of the dominant fronts how they will look from the top or plan view [203]. Solutions at the central regions belong to the corresponding front, while solutions close to and around the centre may

share two or more dominant fronts. It is explained later in this chapter that an operator (*floor*) can be used to place solutions in their closest central region while deciding their dominant front numbers.

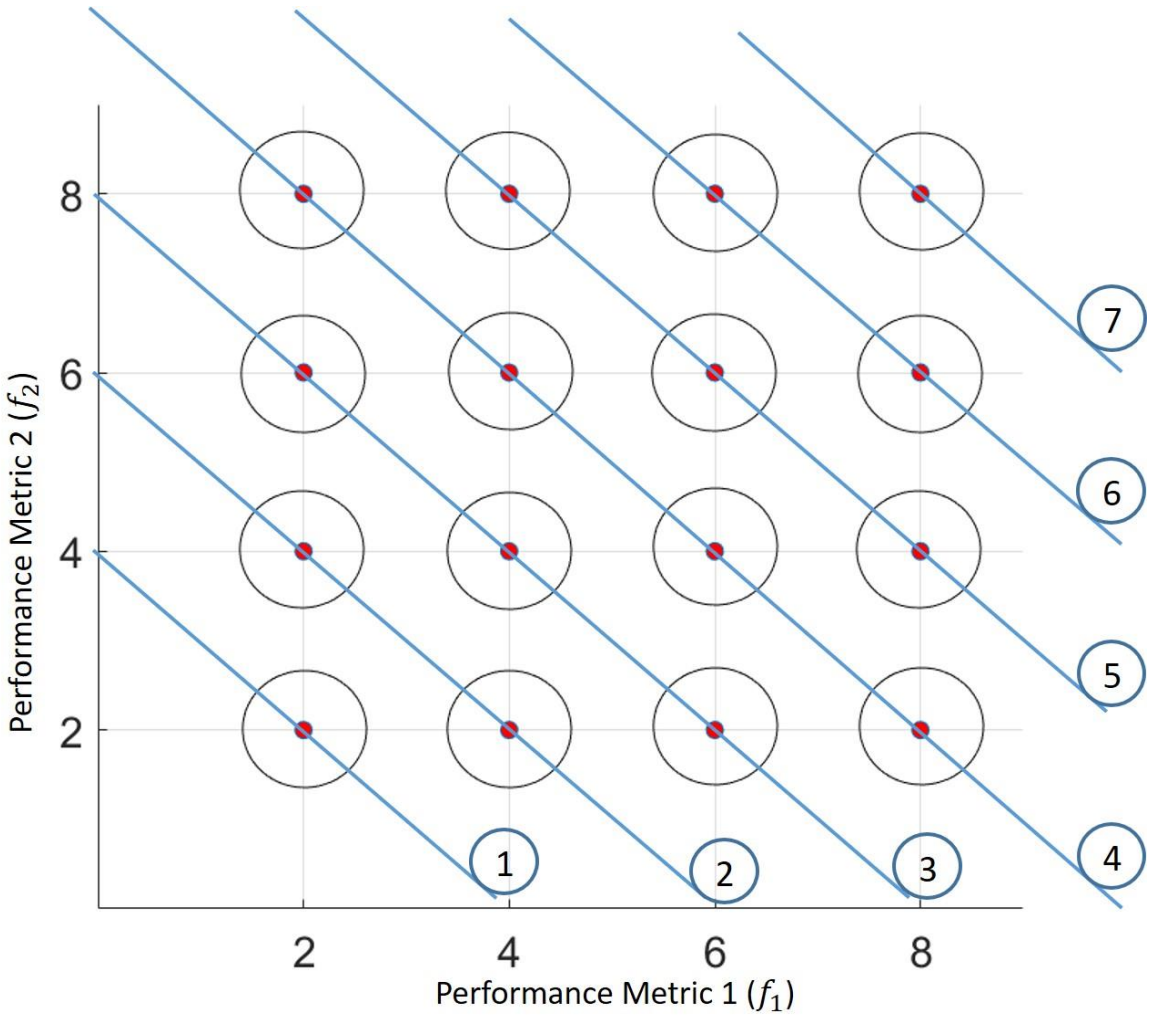


Figure 5.4. Placement of Membership functions in the solution space showing how the dominant fronts are formed.

As in this proposed approach, the dominant fronts in the solution space for a MOP with known number of MFs is *definite*, the user can easily evaluate the worth of solutions obtained after each simulation iteration. Eventually, when solutions belonging to the first front (front numbers are shown in the Figure 5.4) are obtained, the user can confidently terminate the algorithm if the aim was for minimization of objectives. In other words, all solutions in the first front have objective function values which are acceptable to the end user for a *minimization* goal.

5.1.4 Fuzzy Inspired Sorting

After performance metrics have been converted to fuzzy functions and dominant solution fronts are obtained, a given solution is evaluated through a rule-base for an overall/collective fuzzy index for all its performance metric values. A fuzzy system rule-base is a collection of statements containing "*if and then"* rules connecting the inputs to the output variables. A rule-base's general structure can be given as:

$$If\ f_1\ is\ MF_{i1}\ \ and, \ldots\ldots\ldots and f_N\ is\ \ MF_{im}\ \ then\ \ MS_i\ \ is\ \ y_i$$

where $f_1 \ldots f_N$ are performance metric values as inputs to the fuzzy system, $MF_{i1}, \ldots., MF_{im}$ are MFs for the metrics, MS_i is the membership score (which is also *output* for the rule base) for the i^{th} rule and its value in real numbers is y_i. A sum of rules N_R can be calculated from the count of MFs and the number of performance metrics using the following relation (12) [204].

$$N_R = \prod_{j=1}^{N} M_j \tag{5.6}$$

where j is the index used for performance metrics, N is the number of performance metrics, and M_j is the total number of MFs used for the j^{th} metric. Therefore, two metrics represented by four MFs (see Figure 5.1) will form a total of 4×4 or 16 rules. Likewise, two metrics using 2 or 3 MFs will have 4 or 9 rules (see Table 5.2). These rules combine all possible MF arrangements for the two metrics.

Evaluation of competing solutions involves firing its numerical performance metric values through every rule in the rule base. An OMS (overall membership score) for a candidate solution's input performance metrics is the weighted average of all rule outputs. The fuzzy system's output for a set of input performance metrics is computed with the help of a defuzzification scheme which involves the weighted average of all the rules for given input metric values [205]. This procedure is further explained in Table 5.3, which shows outputs from all the stages using hypothetical inputs. Individual rule outputs (y_i) are calculated first

and are essentially the sum of their outputs' membership scores (MS) ($\sum_{j=1}^{N}[MS(m_{ij})]$) appended with one unit to avoid a null/zero value as output (5.7).

$$y_i = 1 + \sum_{j=1}^{N}[MS(m_{ij})] \tag{5.7}$$

where i is used for the rule index, N is the number of metrics we are evaluating and $MS(m_{ij})$ is the membership score for the j^{th} performance metric in the i^{th} rule. Weights (5.9) of metrics for each rule is then computed taking into account the product of all MF values (5.8).

$$MF_{ij}(f_j, \bar{f}_{ij}, \sigma_{ij}) = ae^{-\frac{(f_j - \bar{f}_{ij})^2}{2\sigma_{ij}}} \tag{5.8}$$

$$w_i = \prod_{j=1}^{N} MF_{ij} \tag{5.9}$$

where f_j represents the input performance metrics values and $\bar{f}_{ij}$ and σ_{ij} are the respective MFs' mean and standard deviations. Successive iterations update parameters including mean and standard deviation based on performance metrics' limiting values (f_{min}, f_{max}) using equations (5.1-5.3). The constant term 'a' in (5.8) is basically $\frac{1}{\sigma\sqrt{2\pi}}$ which is related to the standard deviation.

Table 5.3. Dominant front index/OMS for a hypothetical solution having two performance metrics values as 4 & 7.

yi = (MS+1) Eq. (5.7)	MF_f1 Eq. (5.8)	MF_f2 Eq. (5.8)	Wi Eq. (5.9)	X = (wi)/sum(wi) Eq. (5.10)	Y = (X)*(yi) Eq. (5.10)	Y* Eq. (5.11)
1	0.062777	3.06E-08	1.92E-09	1.70E-90	4.4971	4
2	1	0.001973	0.000124	0.00010968		
3	0.062777	0.500553	0.031423	0.02783194		
4	1.55E-05	0.500553	0.031423	0.02783194		
2			3.06E-08	2.71E-08		
3			0.001973	0.00174721		
4			0.500553	0.44334592		
5			0.500553	0.44334592		
3			1.92E-09	1.70E-90		
4			0.000124	0.00010968		
5			0.031423	0.02783194		
6			0.031423	0.02783194		
4			4.76E-13	4.76E-13		
5			3.06E-08	2.27E-08		
6			7.77E-06	6.89E-06		
7			7.77E-06	6.89E-06		

Summarizing the above, a quantitative output from the fuzzy inferencing or the OMS of a solution is the weighted average of outputs from all the rule. In other words, the OMS, which is a representative score for all the metrics in a solution is a real number, can be computed using (5.10).

$$Y = \frac{\sum_{i=1}^{N_R}(w_i\, y_i)}{\sum_{i=1}^{N_R} w_i} \tag{5.10}$$

Finally, the fuzzy dominant front number Y^* is calculated using an operator, 'floor' (5.11) to appropriately place a competing solution in the entire solution space. This operator will round-off the fractional output from (5.10) to the nearest integer values.

$$Y^* = floor(Y) \tag{5.11}$$

It is now possible to define the fuzzy dominance criterion. This definition resembles that of the dominance criterion [61], but considering the OMS instead of individual objective values.

MF shapes and their appropriate placement in the solution space are explained in Figure 5.4 which shows the oblique dominant fronts wherein the membership scores of solutions are all same. If we fail to convert dominant front index into an integer, all the solutions shall have different values for the score. In such a case, two solutions will never share a common score and so in the evolutionary algorithm the crowding distance operator shall not be used at all. In the case when crowding distance operator is not used it is likely that the algorithm will converge to a local optimal value and may not give globally optimal results. The new fuzzy inspired EA can be defined now as below:

Definition *(dominance criterion) Dominance of a solution s_1 will be accepted over another solution s_2 provided one of the following conditions holds true:*

- *The OMS of solution s_1 is greater than the OMS score of s_2;*

- *The OMS of solution s_1 is equal to the OMS score of s_2, however, s_1 has higher crowding distance value;*

5.1.5 Fuzzy Inspired Evolutionary Algorithm (FIEA)

Convergence is an important consideration when simultaneous optimization of a multitude of solutions is under way. As explained in 5.1.3, fuzzy-based selection is applied to both parent and offspring populations, so that a parent will continue to exist in the population through successive iterations until replaced by an offspring solution with a higher dominance rank. This is a necessary condition if the entire population is to converge to Pareto optimality. In a study proposed by Rudolph [206], it was shown by homogeneous finite Markov chain analysis that an algorithm, which keeps the better solutions in the population, will result in achieving the global optimum.

Algorithm 1 Framework of the proposed FIEA

1. $P_O \leftarrow$ Initialize Population (N)
2. $k \leftarrow 0$
3. $R \leftarrow$ Initialize Rank with some integer>1
4. **while** *floor* $(R) > 1$ **do**
5. Evaluate objectives for P_k and store their limiting values, std. deviation and means
6. Fuzzification of objectives
7. Fuzzy dominant fronts based on objective fuzzification
8. $Q_k \leftarrow$ Generate Offspring Population from P_k
9. $T_k \leftarrow P_k \cup Q_k$
10. $S_k = \{T_k, r_k\} \leftarrow$ Fuzzy dominant sorting and ranking (r_k)

11. $R = min\{r_k\}$
12. Sort solutions based on ranks
13. $P_{k+1} \leftarrow 0$
14. $i \leftarrow 1$
15. **while** $|P_{k+1}| + |S_i| \leq N$ **do**
16. $P_{k+1} \leftarrow P_{k+1} \cup S_i$
17. $i \leftarrow i + 1$
18. **end while**
19. $k \leftarrow k + 1$
20. **end while**

5.2 Benefits of Fuzzy Inspired Sorting

5.2.1 Enhanced discrimination between solutions

The previous sub-chapter pointed out that, following the dominance property, the entire solution space can be split into dominant solution fronts of known worth. In order to increase visibility into differences between solutions, the number of MFs can be increased which will increase the number of fronts. However, this will also increase the size of the rule-base and will adversely affect the algorithm's efficiency. There is thus a trade-off between the number of MFs and the level of visibility into performance metrics required. Table 5.4 illustrates the proposed sorting scheme's improved ability to discriminate.

Table 5.4. Evidence of better discrimination among solutions using FIEA over NSGA-III

Solutions	Objectives				ND Front indices	Fuzzy dominant Front indices
	f_1	f_2	f_3	f_4		
Sol. 1	3.600	3.50	3.00	3.50	1	**3**
Sol. 2	4.000	3.49	3.00	4.00	1	**4**
Sol. 3	6.000	6.00	6.00	3.49	1	**11**
Sol. 4	8.000	3.48	6.00	9.00	1	**12**
Sol. 5	3.599	9.00	9.00	9.00	1	**17**

A single example problem with four performance metrics and five candidate solutions is discussed here. Although the non-dominance criterion means that all candidate solutions should share the same ND front, fuzzy dominance means that the solutions have different rankings or front indices, clearly showing that the fuzzy dominance approach makes clear discrimination between candidate solutions possible. Defining four objective functions (as in Table 5.4) with five MFs each will produce 17 dominant fronts and the front indices in Table 5.4 are obtained according to the method discussed in the previous sub-chapter. The given solutions' front indices are more representative than those for the non-dominated solutions, showing that the proposed fuzzy inspired sorting scheme and dominance method can better discriminate solutions.

5.2.2 Deterministic termination criterion

In the proposed fuzzy inspired dominance, the algorithm terminates when either the user decides the desired solutions have been achieved or the front index remains unchanged over a pre-set number of iterations. To calculate front indices for a termination criterion, objectives' limiting values may be recorded through experiment, if not known prior.

5.2.3 Final Solution from the PF

Optimization leads to finding a PF which has solutions that are all equally good, but what the end user wants is a singular solution to actually use. The selection of solution will normally depend on the user's preferred metric values, a subjective and intuitive approach. To help users make these decisions, it is proposed that the OMS index (before rounding off) from equation (5.10) be used. Every solution in the PF has a unique OMS which is a real number and the solution with the smallest OMS is the best compromise or trade-off from all the metrics when optimization's goal is minimization.

While the proposed novel approach is useful for optimizing real-world problems involving many objectives, it can be also be implemented for problems where the working range of performance metric values is not available. Problems wherein accuracy of performance metrics is not defined or where qualitative description of metrics is provided in place of quantitative values.

5.3 Performance Indices for Evaluation

5.3.1 Performance indices

Since MOEA evaluation requires checking such aspects as *accuracy* and the *diversity of solutions*, no single performance index will suffice. A number of performance indices are mentioned in [61]; they can be divided into three categories: indices that check proximity to PF; indices that provide diversity measure; and indices that evaluate both proximity to PF and diversity of solutions. For this research, one performance index has been chosen from each category to evaluate the nature of solutions obtained from various MOEAs. The indices in question are: *GD-metric* value [125]; Spacing (*S-metric* value) [61]; and Normalized Hyper-volume (*H-metric* value) [61], [67] solutions. These metrics are briefly described here.

The *GD -metric* provides the average distance of a given solution in Q (solution set) from P^* (chosen/given Pareto optimal set):

$$GD = \frac{\left(\sum_{i=1}^{|Q|} d_i^2\right)^{1/2}}{|Q|} \tag{5.13}$$

where d_i is the Euclidean distance in the solution space between the solutions $i \in Q$ and their nearest member from P^*. The Euclidean distance is given by (5.14), where f_m^{*k} describes the m^{th} performance metric value of the k^{th} member of P^*.

$$d_i = \min_{k=1}^{|P^*|} \sqrt{\sum_{m=1}^{M} (f_m^i - f_m^{*k})^2} \tag{5.14}$$

The spacing or S-metric measures the relative distance between successive solutions in the solution set (Q).

$$S = \sqrt{\frac{1}{|Q|}\sum_{i=1}^{|Q|}(d_i - \bar{d})^2} \tag{5.15}$$

where $d_i = min_{k \in Q \wedge k \neq i} \sum_{m=1}^{M} \left| f_m^i - f_m^k \right|$ and $\bar{d}$ is the mean of d_i's, which can be calculated using $\bar{d} = \sum_{i=1}^{|Q|} d_i / |Q|$. The distance measure ($d_i$) is the minimum value from the sum of the absolute differences in the performance metric values between the i^{th} solution and other competing solutions in the solution set (Q). The S-metric, therefore, measures the standard deviation of d_i values which are computed for all the solutions in the Pareto set (Q). S-metric values are smaller when spacing between solutions is almost even, and this indicates improved diversity between solutions.

The H-metric gives the volume encompassed in the solution space (or, are in the case of two metrics) by the solutions in Q. In order to find this, a hypercube v_i is created for each solution in Q and reference point *W* and solution *i* are considered as the diagonally opposite corners of this hypercube. *W* is obtained by building a vector of the worst objective function values. Hypervolume is later defined as the union of all hypercubes as (5.16):

$$HV = volume(\cup_{i=1}^{|Q|} v_i) \tag{5.16}$$

However, to avoid ambiguity in scaling objectives, H-metric's normalised value, which is the ratio of hypervolumes of Q and P^*, is used in this research:

$$H = \frac{HV(Q)}{HV(P^*)} \tag{5.17}$$

The desired value of *H* is one (for Q = P^*), when all the metrics are required to be minimised. Therefore, while the normalised hyper-volume should be unity, smaller Spacing and GD-metric values show that the solutions used for certain MOEA is better.

5.3.2 Simulation experiments for test problems suit used in CEC'09

The proposed FIEA algorithm is subsequently applied to the unconstrained test problems suit provided during CEC 2009 (Congress on Evolutionary Computation, 2009) [207]. Following parameters were used (Table 5.5):

Table 5.5. Parameter settings.

Population size	1000
Crossover prob.	0.95
Real-parameter mutation prob.	0.05
Distribution index for crossover	10
Distribution index for mutation	50
Number of independent runs	30

There was no change to parameters during experiments with the test problem suit. Each of the experiments was independently run 30 times and the performance indices which are GD-metric and S-metric calculated from the solution population. Table 5.6 shows GD-metric's mean values and standard deviations from these experiments.

A brief analysis of Table 5.6 shows that the proposed FIEA finds solutions which have better proximity to the true PF for test instances UF01, UF02, UF6 and UF07 among the seven (UF01 to UF07) two-objective problems, but performance is less good for test problems UF03 to UF05. When it comes to problems with three objectives (UF08 to UF10), FIEA provides GD-metric values for UF08 & UF09 that are lower than for test instance UF10. Among the remaining five objective test problems, FIEA was unable to find a good approximation for problem R2DTLZ3M5, but S-metric produced good results for all test instances except R2DTLZ3M5 & WFG1M5.

Table 5.6. Values of distance and spacing indices for the PF solutions received after implementing FIEA on CEC'09 test problem suit.

Problems IDs	**GD-metric Mean (Std. Dev.)**	**S-metric Mean (Std. Dev.)**
UF01	0.0041 (0.00129)	0.0059 (0.00176)
UF02	0.0052 (0.00239)	0.0014 (0.00377)
UF03	0.0104 (0.00379)	0.0059 (0.00466)
UF04	0.0776 (0.00381)	0.0101 (0.00257)
UF05	0.1022 (0.03010)	0.0088 (0.03524)
UF06	0.0061 (0.00189)	0.0084 (0.00344)

UF07	0.0079 (0.00270)	0.0099 (0.00332)
UF08	0.0731 (0.00310)	0.0165 (0.00685)
UF09	0.0438 (0.03543)	0.0126 (0.06315)
UF010	0.1581 (0.07254)	0.0364 (0.06449)
R2DTLZ2M5	0.0847 (0.00627)	0.0653 (0.00730)
R2DTLZ3M5	.658 (34.1896)	1.8652 (1.06031)
WFG1M5	1.7265 (0.01671)	0.1482 (0.01525)

5.4 Results and discussion

The proposed algorithm solved the CEC'09 problem suite well, obtaining smaller GD & S-metrics for most of the problems except for UF05, UF10, R2DTLZ3M5 & WFG1M5. The type of these test problems makes it difficult to improve PF estimation. While results given here are only indicative, FIEA shows improved performance for above test problems discussed in this research when the number of objectives is large. Compared to the results from other EAs on CEC'09 problems the proposed FIEA performed better on all the performance indices [208].

5.5 Summary

An efficient solution of multi-objective problems can be obtained by simultaneous optimization of a group of competing solutions using evolutionary algorithms. However increased number of criteria reduces EAs' ability to discriminate, with an adverse effect on selection of better from good solutions. After a number of iterations, increasing numbers of solutions become non-dominated resulting in a pseudo PF giving an incorrect notion that optimization is successful. Other difficulties motivating this research are vagueness in the algorithm termination and selection of final best solution selection from the Pareto optimal solutions set. Ways existing EAs can be improved have been identified and have been attempted by proposing a fuzzy inspired sorting approach, ***Fuzzy Inspired Evolutionary Algorithm (FIEA).***

6 Neuroevolution of Breast Cancer Data Using FIEA

Developing Neuroevolution to model breast cancer prognosis requires optimization of its hyper parameters by means of evolutionary algorithms (EAs). Selection of best EA involves identifying the one offering best AUROC, F1 score, accuracy, specificity, and sensitivity. In the course of this selection, the performance of existing EAs (NSGA-III, MOEA/D, HypE and SPEA 2) was compared with the FIEA proposed during this research. This chapter compares EAs performance on established benchmark problems and also on modelling breast cancer prognosis. Moreover, it presents five breast cancer samples as a case study in order to be understandable by medical community.

6.1 Existing evolutionary algorithms

The literature contains many proposals for evolutionary algorithms and their extended versions. Recent algorithms will be summarised here for completeness of the discussion, after which the algorithms will be evaluated and compared with the proposed FIEA.

6.1.1 NSGA-III Algorithm

NSGA-III, though operates on the same basic framework as NSGA-II, has a much improved selection mechanism. To ensure solutions are as diverse as possible, NSGA-II generally uses a crowding distance operator which facilitates selection of solutions placed far in the solution space by improving the rank attached to them. The proposed new NSGA-III algorithm substitutes this crowding distance operator with a series of distributed, adaptive reference points, use of which, it is claimed, will not only maintain diversity of solutions but also provide better convergence, while also performing well on multi-objective problems in which objectives have different scales.

Input: H structured reference points Z^s, parent population P_t

Output: P_{t+1}

00: **Begin**
01: $S_t \leftarrow \emptyset$, $i \leftarrow 1$;
02: $Q_t \leftarrow$ Variation (P_t);
03: $R_t \leftarrow P_t \cup Q_t$;
04: $(F_1, F_2, \ldots) \leftarrow$ Non-dominationed_Sort (R_t);
05: **Repeat**
06: $S_t \leftarrow S_t \cup F_i$; $i \leftarrow i+1$;
07: **Until** $|S_t| \geq N$;
08: $F_l \leftarrow F_i$; /*Last front to be included*/
09: **If** $|S_t| = N$ **then**
10: $P_{t+1} \leftarrow S_t$;
11: **Else**
12: $P_{t+1} \leftarrow \bigcup_{j=1}^{l-1} F_j$;
/*Number of points to be chosen from F_l*/
13: $K \leftarrow N - |P_{t+1}|$;
/*Normalize objectives and create reference set Z^r*/
14: Normalize $(F^M; S_t; Z^r; Z^s)$;
/*Associate each member s of S_t with a reference point*/
/*$\pi(s)$: closest reference point*/
/*$d(s)$: distance between s and $\pi(s)$*/
15: $[\pi(s), d(s)] \leftarrow$ Associate (S_t, Z^r);
/*Compute niche count of reference point $j \in Z^r$ */
16: $\rho_j \leftarrow \sum_{s \in S_t / F_l} ((\pi(s) = j)\ ?\ 1 : 0)$;
/*Choose K members one at a time from F_l to construct P_{t+1}*/
17: Niching $(K, \rho_j, \pi(s), d(s), Z^r, F_l, P_{t+1})$;
18: **End If**
19: **End**

Figure 6.1. Pseudo code of the algorithm NSGA-III.

First, a population of N feasible solutions and a broadly distributed series of G-dimensional reference points are identified. The solution space is divided into '*p*' divisions provided by the user and the reference points are so placed on a normalised hyper plane as to have unity intercept on each axis. The number of all the reference points (*H*) for a multi-objective problem is therefore given by (6.1), while the rest of the NSGA-III algorithm is also provided here in a pseudo code.

$$H = \binom{G + p - 1}{p} \tag{6.1}$$

6.1.2 MOEA/D Algorithm

Multi-Objective Evolutionary Algorithm based on Decomposition (MOEA/D) is, like NSGA-III, an evolutionary algorithm framework used for multi-objective optimization.

Step 1) Initialization:

Step 1.1) Set $\mathrm{EP} = \emptyset$.

Step 1.2) Compute the Euclidean distances between any two weight vectors and then work out the T closest weight vectors to each weight vector. For each $i = 1, \ldots, N$, set $B(i) = \{i_1, \ldots, i_T\}$, where $\lambda^{i_1}, \ldots, \lambda^{i_T}$ are the T closest weight vectors to λ^i.

Step 1.3) Generate an initial population $x^1, \ldots, x^N$ randomly or by a problem-specific method. Set $FV^i = F(x^i)$.

Step 1.4) Initialize $z = (z_1, \ldots, z_m)^T$ by a problem-specific method.

Step 2) Update:

For $i = 1, \ldots, N$, do

Step 2.1) Reproduction: Randomly select two indexes k, l from $B(i)$, and then generate a new solution y from x^k and x^l by using genetic operators.

Step 2.2) Improvement: Apply a problem-specific repair/ improvement heuristic on y to produce y'.

Step 2.3) Update of z: For each $j = 1, \ldots, m$, if $z_j < f_j(y')$, then set $z_j = f_j(y')$.

Step 2.4) Update of Neighboring Solutions: For each index $j \in B(i)$, if $g^{te}(y'|\lambda^j, z) \leq g^{te}(x^j|\lambda^j, z)$, then set $x^j = y'$ and $FV^j = F(y')$.

Step 2.5) Update of EP:

Remove from EP all the vectors dominated by $F(y')$.

Add $F(y')$ to EP if no vectors in EP dominate $F(y')$.

Step 3) Stopping Criteria: If stopping criteria is satisfied, then stop and output EP. Otherwise, go to **Step 2**.

Figure 6.2. Pseudo code of the algorithm MOEA/D.

The multi-objective optimization problem is first decomposed into single-objective optimization sub-problems which are then simultaneously optimized by means of a search heuristic.

The MOEA/D algorithm's pseudo-code is given here for reference. A population of 'N' points $x^1, x^2, \ldots . x^N$ is first generated while the current solution of sub-problem is denoted by $x^i s$. External population (EP), as well as being the optimization output, stores the non-dominated solutions obtained during optimization. 'T' is the number of weight vectors in each weight's proximity. The input variable 'N' is the number of sub-problems. The beginning of the algorithm provides the uniform spread of 'N' weight vectors ($\lambda^1, \lambda^2, \ldots . \lambda^N$). FV^i represents the objective function values for $x^i s$ and $z = (z_1, z_2, \ldots , z_m)^T$ refers to each objective's reference point while z_i is the best value obtained for objective f_i.

6.1.3 HypE Algorithm

Hypervolume-based Evolutionary Algorithm (HypE) is another optimization method that is applicable when there are many objectives. It uses Monte Carlo simulations to estimate the ranking of solutions generated by the use of the Hypervolume indicator. More information about HypE, can be found in [209].

6.1.4 SPEA2 Algorithm

The second version of the Strength Pareto Evolutionary Algorithm is an improved version of SPEA, participates only in the mating selection process and so uses a fixed size archive set. The framework is presented with some explanation. An initial population is generated by assigning an empty archive population. Individuals are selected and sorted on the basis of a fitness evaluation, and sorted individuals from population of archive enter the mating selection to fill the mating pool. The resultant population is obtained from crossover and

mutation operators. The ratio of internal population size to archive size is kept constant during optimization.

Input: N (population size), $\breve{N}$ (archive size), n (maximum number of generations)
Output: N^D (nondominated set)
Generate an initial population of P_0 and create the empty archive $Q_0 = \emptyset$, set $i = 0$
Assign fitness values (after calculations) of individuals in $P_i$and Q_i.
Selection of environment: Copy all non-dominated individuals in $P_i$and Q_i to Q_{i+1}. If size of Q_{i+1} exceeds $\breve{N}$ then reduce Q_{i+1} by means of the truncation operator, otherwise if size of Q_{i+1} is less than $\breve{N}$ then fill Q_{i+1} with dominated individual in $P_i$and Q_i.
Terminate If $i > n$ then set N^D as the set of decision vectors represented by the non-dominated individuals in Q_{i+1}.
Mating selection: Perform binary tournament selection with replacement on Q_{i+1} in order to fill the mating pool.
Variation: Apply crossover and mutation operators to the mating pool and set P_{i+1} to the resulting population. Increment generation counter ($i = i + 1$) and go to fitness assignment and reiterate the algorithm.

Figure 6.3. Pseudo code of the algorithm SPEA2.

While optimizing using SPEA2, parametric values which are selected are given as follows:

Table 6.1. Selected parametric values of SPEA2.

The crossover probability: 0.7
Mutation Probability as $\frac{1}{v}$ (v = number of variables)
$N = 200; n = 500; \breve{N} = 200$
Distribution Index (D.I.) for simulated binary crossover (SBX) operator: 15
D.I. for polynomial mutation: 15

6.2 FIEA implementation on test problems (DTLZ & ZDT)

While proposing an alternate evolutionary optimization method, the adequacy of its functioning can be reviewed with an illustrative description of the Pareto optimal front which could be a curve or a hyper-surface. However, it is desired to evaluate the performance of proposed EA on benchmark test problems. The literature contains a range of test problems but

some are intricate and do not show appreciable positions and shapes of the Pareto-optimal front [210]. It is also necessary that test problems be expandable or scalable in terms of both objectives and variables for decision making. Finally, implementation of the EA should be easy and unambiguous.

As both proximity to the PF and diverse solution distribution are required, the proposed FIEA should be tested for these two aspects. Different sets of test problems would normally be required to test solutions' convergence to PF and solution diversity, and the PF provided by the test problems should be non-convex if the diversity of solutions is to be tested. At times the PF should also be discrete and the density of solutions along the PF may vary. *Deb et al*. [211] gives a number of scalable test problems, and these among others have been used in the past to investigate MOEAs [125]. This research used test problems DTLZ1-7 from [211] and ZDT1-4 & ZDT6 from [212] to evaluate the proposed FIEA.

While the set of test problems DTLZ1-7 is used to test the proximity to the PF, the ZDT test problems were selected to test the ability of the proposed algorithm to maintain diversity of solutions. For a number of ZDT problems, the only change is the number of variables and the number of objectives are always two. We have published a description of these benchmark test problems previously [213].

To assess the proposed method's capabilities, the three existing EAs NSGAIII [110], [111], MOEA/D [125] and HypE [67] were also applied to the DTLZ and ZDT test problem suites [211], [212]. These EAs represent three different classes or approaches to evolutionary optimization and our comparison with these approaches is intended to show the wide applicability of the proposed algorithm. The final set of solutions (Q) given by these methods, including the proposed algorithm FIEA, have been subjected to further analysis.

The proposed FIEA is evaluated against other EAs using the performance indices discussed in the previous chapter. Normalized Hyper-volume (H) should be equal to one, while

lower Spacing (S) and GD-metric (GD) values show ascendancy of the MOEA being assessed (FIEA in the present case). Rigorous experiments were carried out while optimizing the sets of benchmark problems (DTLZ and ZDT problem suites). The following sub-chapters discuss parameter selection apart from the conduct of simulations.

6.2.1 Simulations with DTLZ & ZDT test problems suit

The above benchmark test problem suites DTLZ & ZDT were optimized using various approaches including the proposed FIEA, using the following experiment parameters for NSGA III [110], [111] and FIEA in all the experiments.

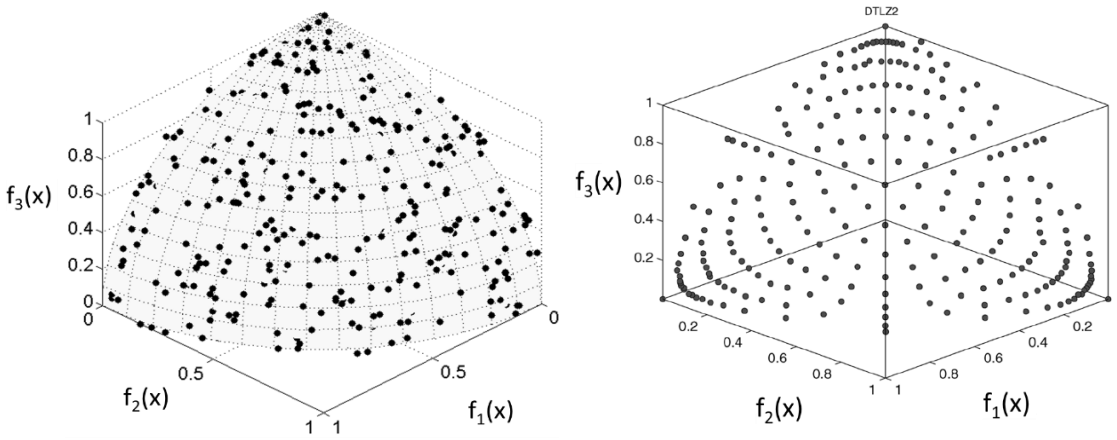
Table 6.2. Simulation parameters for FIEA and NSGA-III.

Population size	1000
Crossover prob.	0.95
Real-parameter mutation prob.	0.05
Distribution index for crossover	10
Distribution index for mutation	50

During the simulations two hundred experiments were carried out on DTLZ & ZDT problems. Each of the experiments was repeated for five times with average performance index values being recorded. While it is suggested to use a varied number of iterations for different problems, we did not change the number of experiments in order to statistically evaluate all four EAs. For all the DTLZ problems, the objective's count was increased gradually, and experiments were performed considering 2, 3, 5, 10 and 20 objectives. For ZDT problems, number of variables were increased from 2 to 20 keeping the count of objectives as two. Simulations were initialised with two objectives in DTLZ1 problem suit, consequently, a linear Pareto-optimal was obtained. In the same manner, a triangular plane for the Pareto front was obtained using three objectives in DTLZ1. On the other hand, working with DTLZ2, quarter of a generic sphere was obtained. Next, test problems DTLZ1, DTLZ2 & DTLZ4 were used to investigate the diversity of solutions obtained from EA. Test problem DTLZ3 was implemented to examine the capability of an EA to provide solutions with proximity to the Pareto front. Ability of an EA to converge solutions can be tested using DTLZ5-6 problem suit. It was

observed during the experiments that NSGA-III failed to provide solutions with good proximity to the true Pareto curve of DTLZ6, instead it resulted in a pseudo Pareto front. In contrast to previous problems, disconnected set of Pareto-optimal regions are obtained using DTLZ7 problems which in facts checks capability of an EA to maintain extra smaller populations in different regions which are also defined as Pareto-optimal. The ZDT test suite of six test problems works only with two objectives and is scalable only in terms of the number of distance parameters, but these tests are still used because they have well defined Pareto optimal fronts and can be compared with readily available standard test results from other research. The ZDT1 test function has a convex Pareto-optimal front while the ZDT2 front is non-convex. Test function ZDT3 is discrete and has a Pareto-optimal front made up of non-contiguous convex parts. Function ZDT4 gives a number of local Pareto-optimal fronts usable to test the multimodal ability of the EA. Test function ZDT5 is not normally used to evaluate EAs since it is binary coded [214]. Test problem ZDT6 has a nonconvex function and is used for evaluation of EAs because of its non-uniform search space.

6.2.2 Simulation results from FIEA and other EAs for DTLZ & ZDT test problems suit



(a) (b)

Figure 6.4. Final solutions obtained from (a) FIEA & (b) MOEA/D for DTLZ 2 test problem with three objectives.

Experiments were run for 200 iterations on the DTLZ and ZDT test suits and the consequent populations of solutions with lowest GD-metric value was analysed. Figure 6.4 shows PF solutions, after implementing FIEA & MOEA/D [125], [215] on DTLZ2 test problem with three objectives. Average distance metric (GD-metric) values of the final population of solutions from FIEA & MOEA/D were found 1.11×10^{-2} and 4.17×10^{-2} respectively which shows that improved convergence can be obtained using proposed FIEA. On the other hand, using MOEA/D the results are found to be in patterns which further means that the complete objective solution space was not explored by MOEA/D algorithm. In order to scan the complete solution space, MOEA/D would need large set of combinations of the objective functions using different weight functions which may not be realistic.

Table 6.3. Expected values of performance metrics resulted from NSGA-III, MOEA/D, HypE and FIEA employed on DTLZ problems.

No. of Objectives		NSGA-III			MOEA/D			SPEA2			FIEA		
		GD	**S**	**H**	**GD**	**S**	**H**	**GD**	**S**	**H**	**GD**	**S**	**H**
DTLZ1	2	0.0196	0.0003	0.9689	0.0317	0.0157	0.2414	0.1264	0.1440	0.8231	0.1142	0.0035	0.9257
	3	0.0498	0.0009	0.9513	0.0654	0.0170	0.2452	0.2696	0.2395	0.8150	0.1868	0.0016	0.7641
	5	0.0244	0.0086	0.9589	0.8860	0.0183	0.2528	0.2360	0.2691	0.6154	0.3629	0.0071	0.9031
	10	1.0756	0.0920	0.8733	1.5731	0.1149	0.6234	1.4954	0.4018	0.8021	1.4565	0.0885	0.8554
	20	1.0845	0.1425	0.8866	1.8489	0.6663	0.9202	1.2831	0.5078	0.8224	1.4547	0.1349	0.9441
DTLZ2	2	0.0052	0.0007	0.9865	0.0389	0.0234	0.8508	0.1319	0.0404	0.7440	0.0066	0.0073	0.9982
	3	0.0147	0.0088	0.9745	0.0417	0.0230	0.8301	1.2773	0.1403	0.8640	0.0111	0.0032	0.8452
	5	0.0096	0.0348	0.8206	1.0967	0.0241	0.84225	1.1611	0.2609	0.8080	1.0731	0.0901	0.7203
	10	0.0961	0.1890	0.7826	1.1272	0.0245	0.8453	1.0959	0.4024	0.8450	1.3902	0.2244	0.8709
	20	0.7950	0.2280	0.8287	1.3637	0.0254	0.8624	1.0598	0.6354	0.8240	1.3927	0.4297	0.8989
DTLZ3	2	0.2080	0.0046	0.8744	0.0752	0.0235	0.8623	1.7815	0.4064	0.8348	1.0159	0.0044	0.8987
	3	0.0641	0.0117	0.7492	10.670	0.6645	0.8445	3.0434	0.1245	0.8333	0.8758	0.0722	0.8347
	5	0.6184	0.0689	0.8511	10.655	1.1029	0.7931	7.6523	1.0009	0.9978	1.2311	0.0027	0.9675
	10	10.421	1.0471	0.9251	9.5083	1.3281	0.8492	8.8912	2.2921	0.9100	9.8646	1.1968	0.99
	20	11.545	1.9205	0.8863	9.2960	2.2081	0.7331	10.462	0.957	0.9090	11.3774	2.2603	0.98
DTLZ4	2	0.0058	0.0012	0.9475	0.6857	0.0066	0.8718	1.2343	0.1067	0.8770	0.0536	0.0107	0.8800
	3	0.0246	0.0094	0.8899	0.3070	0.0379	0.8775	1.0437	0.1189	0.8170	0.2860	0.0140	0.9306
	5	0.0658	0.0801	0.8790	0.9782	0.0025	0.813	1.1420	0.2219	0.8380	1.1315	0.0579	0.8626
	10	1.1515	0.2922	0.8526	1.0697	0.0203	0.8244	1.1318	0.5860	0.8270	1.2814	0.2639	0.9183
	20	1.9536	0.4172	0.8284	1.3007	0.0453	0.8105	1.9118	0.6979	0.8013	1.4819	0.3139	0.9567
DTLZ5	2	0.0664	0.0056	0.9785	0.7686	0.0108	0.8080	1.3427	0.0639	0.7980	0.7898	0.0110	0.8721
	3	0.5734	0.0541	0.9741	0.8639	0.0076	0.8230	1.3767	0.1141	0.7670	0.9537	0.0377	0.9260
	5	1.3850	0.0866	0.9261	1.1098	0.1325	0.9162	1.2664	0.2086	0.7270	1.3452	0.1047	0.9704
	10	1.4294	0.1351	0.9700	1.1883	0.1446	0.9524	1.5176	0.1848	0.7830	1.6274	0.1907	0.9997
	20	2.1141	0.2338	0.8284	1.8824	0.0101	0.8483	1.4370	0.2020	0.7825	1.8573	0.2210	0.9996
DTLZ6	2	5.2144	0.0368	0.9602	7.8628	0.0593	0.9987	8.5931	0.2795	0.8700	6.0179	0.0249	0.9987
	3	6.5176	0.0885	0.9644	8.079	0.3976	0.9992	8.9072	0.6491	0.8790	6.2622	0.1917	0.9988
	5	8.6903	0.2230	0.9636	8.9104	1.2419	0.9990	8.9802	1.0097	0.8126	7.8974	0.1774	0.9918
	10	8.7532	1.2421	0.8900	9.1532	1.8075	0.9870	9.0173	1.8722	0.8422	8.1360	1.3672	0.9969
	20	9.2134	1.8482	0.8766	8.9248	2.5721	0.9761	9.1900	2.6980	0.8068	8.9059	1.6131	0.8918
DTLZ7	2	1.1166	0.0019	0.9447	8.2671	0.0220	0.9232	11.487	0.0514	0.7033	2.9452	0.0077	0.7901
	3	5.4186	0.0178	0.8636	12.3103	0.0897	0.9279	17.984	0.1064	0.7535	7.0973	0.0679	0.8018
	5	18.5081	0.0720	0.8854	27.262	0.3264	0.8823	27.982	0.3040	0.7677	23.802	0.0650	0.8932
	10	58.6788	0.1678	0.8602	57.850	0.7150	0.9615	59.652	0.8295	0.7434	56.9858	0.0145	0.9562
	20	115.547	0.3977	0.8447	115.26	1.9394	0.9890	115.85	1.8859	0.7707	112.627	1.4947	0.9899
Av. Values		**7.78432**	**0.2619**	**0.8984**	**9.2089**	**0.4533**	**0.8224**	**9.4575**	**0.5718**	**0.8118**	**8.0942**	**0.3078**	**0.9149**

Resulting populations of solutions after each simulation were then examined using the three performance indices: GD -metric, S-metric and H-metric. Table 6.3 shows results of all experiments for DTLZ test problems. Analysing briefly, the results indicates that, when there are more than five objectives, the performance of the proposed FIEA is equal to or better than other methods. This paper has stressed all the way through that the method proposed (FIEA) discriminates better between candidate solutions, particularly given many objectives, and Table 6.3 bears this out. Considering a preferred unity value for *H* metric, FIEA outperforms every other approach for almost every DTLZ test suite problem. Experimental findings for DTLZ problems suggest that using HypE, the Hypervolume index was very poor for almost all the solutions (as low as 6.154×10^{-2}). Further, it was observed that for *S* metric, FIEA (with S_{min}=0.16×10^{-2}) outperformed HypE (S_{min}=4.04×10^{-2}) and MOEA/D (S_{min}=0.25×10^{-2}) collectively for most of the problems while FIEA performed comparably with NSGAIII (S_{min}=0.03×10^{-2}).

Table 6.4. Performance metrics from PF solutions obtained after employing NSGA-III, MOEA/D, HypE and FIEA on ZDT problem suit.

Number of Variables		NSGA-III			MOEA/D			HypE			FIEA		
		D	**S**	**H**	**D**	**S**	**H**	**D**	**S**	**H**	**D**	**S**	**H**
ZDT1	2	0.3961	0.0014	0.8346	0.1055	0.0010	0.7048	2.93	0.1240	0.5980	0.1932	0.0014	0.8624
	3	0.5362	0.0014	0.9159	0.1061	0.0012	0.7569	2.69	0.4735	0.5812	0.2430	0.0019	0.9032
	5	0.4212	0.0013	0.9278	0.425	0.0011	0.7834	2.69	0.0907	0.7460	0.2637	0.0020	0.7378
	10	0.4655	0.0014	0.8265	0.445	0.0013	0.8333	2.88	0.1125	0.6550	0.4061	0.0050	0.8377
	20	0.6059	0.0012	0.8287	0.618	0.0014	0.8738	2.81	0.1046	0.5872	0.5075	0.0015	0.8052
ZDT2	2	0.7354	0.0020	0.8777	0.1079	0.004	0.819	4.44	0.5912	0.8280	0.0727	0.0013	0.8547
	3	0.6966	0.0017	0.8922	1.126	0.004	0.821	4.55	0.3772	0.8450	0.5175	0.0012	0.7117
	5	0.7330	0.0017	0.8227	1.130	0.006	0.835	3.94	0.1285	0.7531	0.5787	0.0018	0.7781
	10	0.7309	0.0015	0.8462	0.6884	0.0054	0.9835	4.51	0.1051	0.7245	0.6423	0.0018	0.8023
	20	0.6710	0.0015	0.8811	0.7150	0.0157	0.923	4.65	0.1332	0.7675	0.6142	0.0014	0.9076
ZDT3	2	0.2496	0.0010	0.8146	0.0143	0.0465	0.396	2.66	0.2240	0.6441	0.1247	0.0017	0.9103
	3	0.2787	0.0011	0.9345	0.0364	0.0068	0.857	2.67	0.2270	0.6682	0.1142	0.0016	0.9536
	5	0.3718	0.0012	0.9354	0.198	0.0037	0.9168	2.71	0.3050	0.6808	0.1938	0.0015	0.8960
	10	0.4180	0.0012	0.8305	0.523	0.0113	0.7128	2.89	0.0757	0.6617	0.3655	0.0013	0.8532
	20	0.3765	0.0010	0.8362	0.2057	0.0078	0.7931	3.08	0.1603	0.8238	0.3084	0.0019	0.8882
ZDT4	2	3.2865	0.0012	0.9406	0.2341	0.0119	0.8733	11.42	0.0230	0.6964	0.1351	0.0018	0.9003
	3	3.2865	0.0012	0.9344	0.2641	0.0237	0.8724	14.50	0.0209	0.6438	0.1264	0.0017	0.7248
	5	3.2172	0.0013	0.8731	0.9305	0.0310	0.8435	7.619	0.2266	0.7825	0.5930	0.0020	0.8727
	10	3.2785	0.0011	0.9140	0.9443	0.0657	0.9115	6.220	0.2520	0.8404	0.6443	0.0017	0.9597
	20	3.2785	0.0011	0.8284	0.9599	0.0677	0.9267	4.308	0.2715	0.8946	0.6599	0.0016	0.9818
ZDT6	2	2.9510	0.0025	0.8324	0.0042	0.0062	0.7696	6.51	0.1391	0.5061	0.5658	0.0063	0.8237
	3	3.9005	0.0091	0.8648	0.9913	0.0306	0.7678	7.27	0.0602	0.6027	0.9129	0.0037	0.8221
	5	3.8005	0.0045	0.9290	1.1367	0.1136	0.6272	7.24	0.1094	0.7825	1.9256	0.0012	0.8709
	10	3.6899	0.0188	0.9130	1.9286	0.0018	0.8129	7.42	0.0534	0.7055	2.5347	0.0031	0.9202
	20	4.6155	0.0125	0.9347	3.6010	0.0496	0.8037	7.44	0.0330	0.7211	3.6386	0.0013	0.9693
Av. Values		**1.7196**	**0.0029**	**0.8787**	**0.6975**	**0.0207**	**0.8087**	**5.2018**	**0.1768**	**0.7096**	**0.6753**	**0.0021**	**0.8619**

Once again for *GD* metrics, both NSGAIII (GD_{min}=0.52×10^{-2}) and FIEA (GD_{min}=0.66×10^{-2}) did well while performance of HypE (GD_{min}=12.64×10^{-2}) and MOEA/D (GD_{min}=3.17×10^{-2}) was below expectations.

When the results from ZDT problems are analyzed (see Table 6.4), FIEA performed better than any other method for *H-metric* and *S-metric*, offering the smallest variation range for *H-metric* and a minimum value for *S-metric* of 0.0063.

GD-metric results from the proposed FIEA were comparable with NSGA III results and surpassed the other two methods. Analyzing the performance metrics obtained from different approaches confirms that the fuzzy based sorting method which this research proposes and is implemented through FIEA, has shown itself to be effective.

6.3 Neuroevolution of breast cancer data using EAs

This research used Neuroevolution for prognostic modelling of heterogeneous breast cancer data, with deep neural network optimization performed using four important evolutionary algorithms: NSGA-III, MOEA/D, SPEA 2 and FIEA.

The FIEA algorithm is reproduced here from the previous Chapter for ready reference; the other algorithms have been discussed earlier in this Chapter. Data pre-processing and subsequent selection of variables were explained in Chapter 3, and the Neuroevolution experiments were designed in the way described in Chapter 4. Enhancing performance by reducing network error may not be sufficiently robust and may result in over-fitting the training data. Three performance metrics were chosen to test the robustness of the prognosis prediction: accuracy, AUROC curve and F1 score.

While implementing various EAs for Neuroevolution, 1,000 DNN models were initialized with varying hyper parameters, as listed in Table 6.5.

Algorithm: Framework of the proposed FIEA

1. $P_O \leftarrow$ Initialize Population (N)
2. $k \leftarrow 0$
3. $R \leftarrow$ Initialize Rank with some integer>1
4. **while** *floor* $(R) > 1$ **do**
5. Evaluate objectives for P_k and store their limiting values, std. deviation and means
6. Fuzzification of objectives
7. Fuzzy dominant fronts based on objective fuzzification
8. $Q_k \leftarrow$ Generate Offspring Population from P_k
9. $T_k \leftarrow P_k \cup Q_k$
10. $S_k = \{T_k, r_k\} \leftarrow$ Fuzzy dominant sorting and ranking (r_k)
11. $R = min\{r_k\}$
12. Sort solutions based on ranks
13. $P_{k+1} \leftarrow 0$
14. $i \leftarrow 1$
15. **while** $|P_{k+1}| + |S_i| \leq N$ **do**
16. $P_{k+1} \leftarrow P_{k+1} \cup S_i$
17. $i \leftarrow i + 1$
18. **end while**
19. $k \leftarrow k + 1$
20. **end while**

Table 6.5. Initial hyper parameters for DNNs.

Number of neurons in each hidden layer	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

The algorithms were terminated when statistical tests (t-test and 2-variance test) suggested that the populations of parent and offspring solutions have similar mean and variance during consecutive iterations.

6.3.1 NSGA-III Implementation

Table 6.6 lists the simulation parameters considered during NSGAIII implementation.

Table 6.6. Simulation parameters for NSGA-III.

Population size	1000
Crossover prob.	0.90
Real-parameter mutation prob.	0.1
Distribution index for crossover	10
Distribution index for mutation	50

An initial population of 1,000 DNN solutions was randomly generated with four hyper parameters (Table 6.5) defined as variables within their limiting ranges. Vital DNN parameters include connection weights (initialized randomly), momentum index (kept constant at 0.9) and sigmoidal gain and threshold values (initialized as unity).

A linear activation function was chosen for initialized model's input layer. As predicting survivability is a binary classification problem, a sigmoidal activation function was chosen for the output layer.

6.3.2 MOEA/D Implementation

1,000 DNN models were initialized with varied hyper-parameters during MOEA/D implementation, and then $z = (z_1, z_2, \ldots, z_m)^T$ was initialized as a set of reference points for each performance metric where z_i is the best value obtained for performance metric f_i. The temporary elite population size (K) is decided by the number of sub-problems during the

algorithm's run which is essentially the set of best solutions obtained using the Tchebycheff aggregation function (g^{te}) or by evaluating all the sub-problems' performance metric functions . The resultant population of solutions was stored for later comparison with similar solution populations obtained from other evolutionary algorithms.

6.3.3 SPEA 2 Implementation

A fixed set of archives was selected while using SPEA2, since the algorithm was only involved in mating selection. In the pseudo code for SPEA2 algorithm at the start of this Chapter, the requisite steps are shown in bold. When the initial population is generated, an empty archive is also assigned. Later, individuals are chosen and sorted according to their fitness values. Individuals sorted from the archive are taken into the mating pool where after crossover and mutation they provide the resulting population [110]. Various SPEA2 parameters selected during this research are summarized in Table 6.7.

Table 6.7. Parameters of SPEA2.

The crossover probability	0.75
Mutation probability	1/v (v = number of variables)
N	1000
n	2000
N	1000
Distribution index (D.I.) for simulated binary crossover (SBX)	15
D.I. for polynomial mutation	15

Through all SPEA2 iterations, the proportion of internal population size to archive size is kept equal. An initial population of 1000 DNN solutions is optimized through this evolutionary algorithm to provide an optimal solution on termination.

6.3.4 FIEA Implementation

The three performance metrics, accuracy, F1 score and area under ROC curve, are converted to fuzzy performance metrics (see Figure 6.2) and then the parameters shown in

Table 6.8 are used for further Neuroevolution. Rest of the algorithm was implemented as discussed in Chapter 5 of this Thesis.

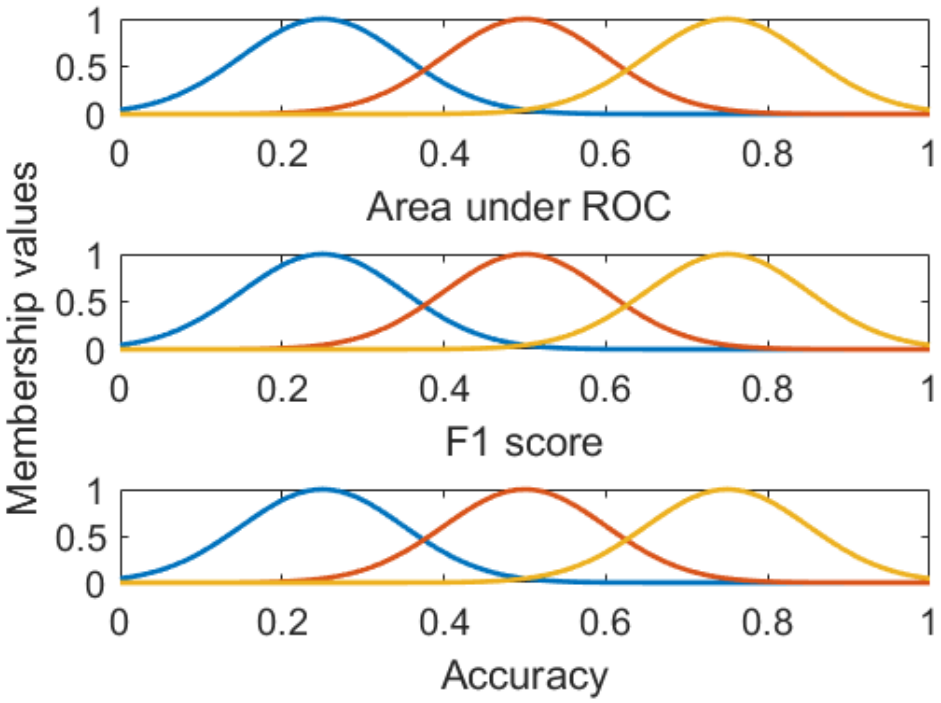


Figure 6.5. Performance metrics converted to fuzzy performance metrics.

Table 6.8. Parameters for Neuroevolution.

Population size	1000
Crossover prob.	0.90
Real-parameter mutation prob.	0.1
Distribution index for crossover	10
Distribution index for mutation	50
Termination criterion	When the parent and off spring populations are not different significantly

6.4 Results and Analysis

The breast cancer data was divided into train samples (90%) and test samples (10%), and a ten-fold cross validation performed on the train sample to produce optimal model parameters. Each experiment was performed five times and the mean and standard performance metrics values recorded. After successfully optimizing DNNs by means of evolutionary algorithms, Pareto optimal sets of DNNs were obtained. Table 6.9 shows performance metrics from individual EAs' best solutions. A single solution was selected from the set of Pareto

optimal solutions by means of a method explained in the author's work described in Chapter 4 and [172].

Table 6.9. Results from the experiments showing performance metrics from the populations of PF solutions obtained after implementation of NSGAIII, MOEA/D, SPEA 2 and FIEA for Neuroevolution of DNNs. (Mean values shown as μ whereas σ stands for the standard deviations).

		Training Accuracy	Training F1	Training ROC	Validation Accuracy	Validation F1	Validation ROC
NSGA-III	μ	0.9485	0.9646	0.9702	0.9468	0.9625	0.9681
	σ	0.0211	0.0152	0.0128	0.0211	0.0148	0.0125
MOEA/D	μ	0.9610	0.9663	0.9714	0.9591	0.9643	0.9692
	σ	0.0016	0.0012	0.0011	0.0013	0.0009	0.0006
SPEA 2	μ	0.9606	0.9666	0.9718	0.9588	0.9646	0.9697
	σ	0.0017	0.0010	0.0010	0.0013	0.0006	0.0004
FIEA	μ	0.9612	0.9665	0.9717	0.9593	0.9645	0.9696
	σ	0.0016	0.0012	0.0011	0.0013	0.0011	0.0007

Further evaluation of the proposed FIEA method's performance required examination of whether differences between the solutions from FIEA and other approaches are non-random or were significantly different.

6.4.1 The Wilcoxon signed ranks test

The *Wilcoxon signed rank test* is a non-parametric statistical test conducted during this research [216] with the aim of checking whether the difference between two sample means is significant. The Wilcoxon signed rank test is not dependent on the form of population distribution and its parameters and so this test is often used where the population may not show normal distribution. The test is briefly explained here and readers may consult [216] for further details.

While comparing populations of solutions from two evolutionary algorithms, assume that the difference between the i^{th} performance metric of the two populations on j^{th} out of n problems is Δ_i^j. Absolute values of these differences (Δ_i^j) are ranked and the sum of ranks (R_i^+) for which the proposed algorithm (FIEA) performs better is calculated. The sum of ranks (R_i^-) for which the proposed algorithm's performance has been poor is also calculated. To avoid bias

due to the different ranges of these performance metrics, they are normalised before being ranked. The null hypothesis being tested here is that the two populations of solutions being compared have equal means. Let S_i be the smaller of the two sums ($R_i^+ \& R_i^-$) for the i^{th} performance metric. Table 6.10 has the results of the Wilcoxon signed ranks test and shows that FIEA outperforms NSGA III (GD-metric), MOEA/D (S & H-metrics) and SPEA 2 with a level of significance of $\alpha = 0.01$. It outperforms NSGA III (S & H-metrics) and MOEA/D (GD-metric) with a level of significance of $\alpha = 0.05$. Further, the null hypothesis of equality of means is rejected since S_i is less than or equal to the critical values for the Wilcoxon distribution for n degrees of freedom. This demonstrates that the proposed algorithm outperforms the other algorithms with the associated p-value.

Table 6.10. Results from the Wilcoxon signed ranks test shows that FIEA outperforms NSGA III (GD-metric), MOEA/D (S & H-metrics) and SPEA 2 with a level of significance of α=0.01, NSGA III (S & H-metrics), MOEA/D (GD-metric) with a level of significance of α=0.05.

	FIEA versus NSGA III			FIEA versus MOEA/D			FIEA versus SPEA 2		
	GD-metric	S-metric	H-metric	GD-metric	S-metric	H-metric	GD-metric	S-metric	H-metric
p-Values	< 0.001	0.0497	0.0174	0.0458	0.0018	< 0.001	0.0034	< 0.001	< 0.001
R^+	1054	1081	1165	1164	1339	1404	1420	1575	1676
R^-	342	753	856	666	491	426	436	73	67

6.5 Case study

In this sub-chapter a case study is presented to evidence successful implementation of the FIEA towards prognosis. The case study consists of 5 samples randomly chosen from the SEER database. The samples, described in Table 6.11, are pre-processed before implementing the FIEA. Screenshots of the code for data pre-processing and running the FIEA along with the results are provided in Figure 6.6. As described in chapter 3 the dependent variable survival month was transformed into a binary format whereby values greater than sixty months become one and values less than sixty months were assigned a zero value. The ultimate goal of the FIEA is to correctly predict survivability for given sample case. As it is evident from the results in Figure 6.6, all the sample cases are correctly predicted.

Table 6.11. Description of the random five samples.

#	Variable name	Sample 1	Sample 2	Sample 3	Sample 4	Sample 5
1.	Age (years)	65-69	60-64	35-39	80-84	80-84
2.	Race	Black	White	White	White	White
3.	Year of birth	1939	1945	1969	1929	1929
4.	Marital status	Single (never married)	Widowed	Married (including common law)	Widowed	Widowed
5.	State	California	California	California	California	California
6.	Year of diagnosis	2007	2005	2004	2012	2013
7.	Behaviour code	In situ	Malignant	Malignant	Malignant	In situ
8.	Primary site	509	501	503	502	504
9.	Histologic type	8010	8520	8500	8500	8500
10.	Grade	Poorly differentiated; Grade III	Unknown	Moderately differentiated; Grade II	Moderately differentiated; Grade II	Moderately differentiated; Grade II
11.	Laterality	Left - origin of primary	Left - origin of primary	Right - origin of primary	Right - origin of primary	Left - origin of primary
12.	Diagnostic confirmation	Positive histology	Positive histology	Positive histology	Positive histology	Positive histology
13.	Reason no surgery	Recommended but not performed, patient refused	Surgery performed	Surgery performed	Surgery performed	Surgery performed
14.	Tumour size	13	11	110	12	27
15.	Extension	0	400	100	100	0
16.	Lymph nodes	0	0	0	0	0
17.	Metastasis	0	0	0	0	0
18.	Cause of death	Alive	Breast	Alive	Alive	Alive
19.	Survival month	0	1	1	0	0

```
In [42]:
1  X, y_true = data_preprocess(data)
```

```
In [45]:
1  model = Sequential()
2  model.add(Dense(11, kernel_initializer="uniform", input_dim=84))
3  model.add(Activation('relu'))
4  model.add(Dense(11, kernel_initializer="uniform"))
5  model.add(Activation('relu'))
6  model.add(Dense(11, kernel_initializer="uniform"))
7  model.add(Activation('relu'))
8  model.add(Dense(11, kernel_initializer="uniform"))
9  model.add(Activation('relu'))
10 model.add(Dense(1))
11 model.add(Activation('sigmoid'))
12
13 model.compile(loss='categorical_crossentropy', optimizer='adam')
```

Figure 6.6. The code for data pre-processing and running the FIEA.

```
In [48]:
1  y_pred = model.predict(X)
2
3  for i in range(len(y_pred)):
4      print('Predicted value:', y_pred[i], '%', 'True value:', y_true[i], '%')
```

```
Predicted value: 0.028 % True value: 0 %
Predicted value: 0.957 % True value: 1 %
Predicted value: 0.969 % True value: 1 %
Predicted value: 0.041 % True value: 0 %
Predicted value: 0.039 % True value: 0 %
```

Figure 6.7. The results of the FIEA.

6.6 Summary

To investigate various aspects of the proposed FIEA algorithm and assess the quality of the solutions it produces, benchmark test problems in the form of variants of DTLZ and ZDT were implemented and optimized. Three other popular methods (NSGAIII, MOEA/D and HypE) were implemented on the same set of problems. To evaluate and compare the quality and accuracy of population of solutions obtained by different approaches, the three performance indices, *GD-metric*, *S-metric* and *H-metric* were computed and analysed. The proposed method was also implemented on the Neuroevolution of breast cancer data, a complex real-life optimiation problem.

The FIEA was found by all performance indices to perform well for all test instances from the DTLZ and ZDT problem suites, compared with other evolutionary algorithms. FIEA also did well in optimizing Neuroevolution involving breast cancer data. FIEA was found to

be equally accurate as MOEA/D on training data but better on validation data. Of particular importance, FIEA performed better than NSGA-III. MOEA/D is less efficient than FIEA because it needs more function evaluations, a metric on which FIEA is comparable to NSGA-III, though its greater accuracy and improvements in other performance metrics mean that FIEA is found to be better than NSGA-III. So far in this research, the fuzzy inspired scheme's greater discriminating power has been well exhibited.

7 Conclusion and Future Work

This research's main goal was to investigate a number of issues concerning data analytics of breast cancer given the database's immense size and heterogeneous nature. Research objectives were drawn from the challenges identified in big data analytics of breast cancer. This Chapter summarizes the research methodologies developed during this research and highlights important conclusions drawn. Major contributions extending to existing science and knowledge have also been highlighted in the following sections of this Chapter.

7.1 Significant Research Outcomes and Contributions

During the course of this research, data bases concerning the breast cancer were acquired and processed for further data analytics work. Most of the data from available databases were of heterogeneous nature and therefore some sort of pre-processing of the databases were required in order to process/mine such data. The heterogeneous data consisted of the categorical as well as the continuous data types and therefore various encoding schemes were employed to convert the categorical data into a continuous domain. As one of the major contributions to the existing knowledge, a neuroevolution based model was developed for the data analytics of breast cancer databases as discussed in Chapter 4. While carrying out the modelling task, several performance indices were identified to evaluate the resulting model. Since employing a singular approach for modelling may not be reliable, existing models from the area of machine learning and artificial intelligence were implemented and compared for their performances on breast cancer data analytics. While it was found that the extant evolutionary algorithm called NSGA-III performed better in comparison with other EAs, this algorithm had some fundamental flaws which prevent its use especially when the number of

performance metrics were more than two. Further research into the working of EAs lead us to some major opportunities for improvement in the sorting schemes of existing EAs and eventually a fuzzy based approach for sorting was devised. The proposed algorithm with new sorting scheme and few other improvements is termed in this research as Fuzzy Inspired Evolutionary Algorithm (FIEA). Next the proposed FIEA was implemented on certain popular benchmark problems (termed as DTLZ & ZDT in the literature) that are normally used to test various aspects (such as convergence to the true Pareto Front and diversity in solutions). It was found that FIEA performed well especially when the number of objectives was more than two for almost all the test problems. Subsequent to testing FIEA on these benchmark test instances and comparing with other EAs, it was finally implemented on the neuroevolution for breast cancer data analytics along with other EAs including the NSGA-III. The outcomes from these simulation experiments revealed that the proposed FIEA performed well since we were using many performance metrics for evaluation of neuroevolution and other existing EAs were not capable of optimizing many objectives simultaneously. In the following subsections, each of these research contributions and their significance related to breast cancer data analytics, is discussed in detail.

7.1.1 Entity Embedding for Heterogeneous Breast Cancer Data

In order to deal with the heterogeneous nature of the databases and convert the categorical data into continuous domain, initially, factorial analysis of mixed data (FAMD) approach was adapted and implemented combining the PCA (Principal Component Analysis) and MCA (Multiple Correspondence Analysis) methods together. Use of factorial analysis facilitated the simultaneous processing of categorical data with the continuous data. Data analytics was performed using the penalized logistic regression and the neural network as classifiers. However, FAMD approach is unsupervised and does not consider the relationship

between predictor and target variables. Further it was observed that the encoding of qualitative variables into binary format during implementation of FAMD method is also not sufficient to represent the categorical variables. Therefore, an entity embedding approach was adapted later using one-hot encoding for the conversion of the categorical variables. Once again this type of encoding was also not found suitable in the present research since the databases were of high cardinality which resulted into a large feature space due to the curse of dimensionality. Eventually a neural network based entity embedding was used to produce continuous vector representation with known dimensions for each of the categorical variables in the database. An Entity Embedding Neural Networks model was later developed for survivability prediction of breast cancer victims. Improved performance of the EENNs classifier over existing classification algorithms namely, Logistic Regression, Gradient Boosting, Random Forest and Support Vector Machines was observed during many simulation experiments.

7.1.2 Neuroevolution for Breast Cancer Prognosis Modelling

It was desired that the prognosis of breast cancer data should be accurate and at the same time interpretable. The interpretability is desired in order to establish the efficacy and the robustness of the algorithms used. Therefore it was envisaged that the prognosis results be evaluated using various performance metrics. During this research three important metrics namely, F1 score, accuracy and AUROC curve were chosen to test the supremacy of algorithms that are used for prognosis. Since there are many performance criteria to optimize it was decided to use evolutionary algorithms (EAs) to optimize all the performance metrics simultaneously. Neuroevolution, which essentially is an artificial intelligence based approach, uses EAs to optimize neural networks (NNs) and as a result provide a NN that has optimal parameters, and network configuration. In this research we initially developed the basic Deep Neural Network (DNN) models for breast cancer data analytics and later used an evolutionary

algorithm called NSGA-III to optimize and provide network parameters together with the hyper parameters of DNNs. Neuroevolution eventually provide us with a set of
Pareto optimal DNN models that were capable of giving equally good breast cancer prognosis. However, we still needed to use a singular solution from the Pareto set which is the most appropriate solution for our purpose. Carrying out further research in this direction we developed a novel approach based on fuzzy inference through which it is possible to extract a singular DNN solution from equally good DNNs.

As yet another milestone, we were able to decode the relationship between the DNN hyper-parameters (these are the building blocks of DNN) and the performance metrics employed. This further can facilitate to achieve specific performance metric by modifying the set of hyper parameters. This research enhances our existing knowledge about the DNN functions and at the same time adds to the transparency and interpretability of DNN modelling. This in turn may allow us to perform training experiments more efficiently and effectively besides increasing acceptability of our trained models to the medical practitioners.

Nevertheless, during the experiments conducted for neuroevolution using NSGA-III, it was observed that the evolutionary algorithm used (NSGA-III) may not perform well while simultaneously working with many performance criteria/objectives. During simulation experiments, frequently, NSGA-III converged prematurely to a false or pseudo Pareto optimal front. It was later identified that the sorting scheme used in NSGA-III will make almost all the competing solutions as non-dominating ones after a few iterations. This investigation lead us to the conclusion that while other genetic operators such as crossover, mutation, crowding distance etc. were indispensable and were working fittingly, the sorting scheme of NSGAIII's required to be improved/modified

7.1.3 Fuzzy Inspired Evolutionary Algorithm (FIEA)

In the light of many performance metrics used for neuroevolution of breast cancer data and the NSGA-III converging prematurely, we further worked on the non-dominance based sorting scheme of EAs. It was found that the better solutions from populations were selected based on their non-dominance and not the dominance over other solutions. Working on this further we devised a scheme based on the fuzzy inference owing to which it became possible to select solutions based on their dominance over other solutions. The new approach is termed as fuzzy inspired evolutionary algorithm (FIEA) in this research. The proposed algorithm also helped in improving other aspects of the existing evolutionary algorithms such as offering better discrimination between solutions, discretely dividing the solution space into fronts of known solution qualities and thus providing a definitive criterion for algorithm termination and selecting a final solution from the Pareto optimal solutions set. In the pretext of proposing the new FIEA, ways existing EAs can be improved have been identified and have been attempted by proposing a fuzzy inspired sorting approach.

There is an interesting set of test problems offered by the Congress on Evolutionary Computation (CEC) in 2009. These problems are a mix of test instances offering varying profiles of Pareto fronts and are quite challenging for EAs. Furthering our research we implemented these 13 CEC problems and optimized using the proposed FIEA approach. Three performance indices were used to evaluate the results from simulations leading to optimization and it was encouraging to note that FIEA performed well on all the three performance indices compared to results from other EAs in the literature [208].

7.1.4 Neuroevolution of Breast Cancer Data Using FIEA

The proposed FIEA was required to be validated with some more benchmark test problems in order to establish its hegemony over existing EAs. In this direction, we used seven test problems called DTLZ and six ZDT problems which are popular problems normally used

for all the newly proposed EAs. Both, the DTLZ & ZDT test problems are expandable in terms of number of variables as well as number of objectives which further makes them interesting problems. The FIEA was implemented on these test problems and at the same time also compared with other existing popular EAs (NSGA-III, MOEA/D and HypE).

For the purpose of comparison we chose three important performance indices which covers all the aspects of EAs which are important for evaluation. Subsequent to various simulation experiments it became clear that the proposed FIEA performed better than other algorithms in particularly for the cases where the number of objectives were large.

Subsequent to the successful neuroevolution of DNN employing FIEA, a set of Pareto optimal DNN models was obtained wherein all the DNN models were able to provide accurate and robust prognosis of breast cancer data. However since we need only one single DNN model to use, once again a fuzzy based approach (described in Chapter 5) was used to compare and find the best suitable DNN model from the Pareto set.

During this research, in order to investigate the efficacy of the FIEA, a statistical test was also conducted. This test was performed to examine whether the differences between the Pareto optimal DNN solutions obtained from FIEA and similar solution sets from other EAs were significantly or non-random. Wilcoxon's signed ranks test was used and while the significant differences between the means of populations resulting from FIEA and other EAs were observed, FIEA also outperformed the other EAs with the associated level of significance and the *p*-values.

7.2 Future Work

7.2.1 Heterogeneous Breast Cancer Data

While many research questions were identified and addressed during the course of this thesis work, a number of advanced problems still remains to be investigated. First of all, during

the implementation of FAMD method which is an unsupervised learning approach we could not consider the relationship between the predictor and the target variables. An alternative supervised approach namely, Partial Least Squares exists which is capable of including predictor target relationships and requires even less dimensionality to achieve expected results and the same can be investigated in future. Further, in the present research work, the number of dimensions were selected manually on the basis of their eigenvalues which could be considered as a hyper-parameter with the number of dimensions and can be chosen automatically on the basis of a variance threshold. While using one-hot encoding, hash function can be used in future which needs fewer dimensions and maps arbitrary size data to fixed size data.

During this research, due to lack of time we could not work rigorously to resolve the issue of missing data in the databases which is quite a current research issue. However some preliminary work already has been done and we have achieved some presentable results. We endeavour to continue this research in future. Conventionally, the missing values were approximated using estimation and attribution approaches, however, these approaches lack efficiency which depends on the assumptions made on data features. Currently, surrogate decisions algorithms combined with Random Forest [217] besides complete case analysis [218] are being investigated for approximation of missing values in databases.

In the present research we used a novel approach of finding the weighted means for the missing values in place of simple estimations. Following this method, all the missing values differ by some amount (based on the weights) from the estimated value. Initially all the mixed data is converted to the continuous data domain using one hot encoding and then data columns are normalized. Later, weights are identified by obtaining distances of missing data row with the closest row. These weights are finally used to redefine the estimated missing values.

The pseudocode used for the ongoing research is given here:

```
Y = Set (Target variables)
X = Set (Training variables)
One-hot encoding of categorical variables in X
K = Set (variables from X with missing values)

For each k in K:
        x_new = k
        X_new = X.drop (x_ new) # drop x_ new from X
        X_new = X_new.replace_nans (with means) # replace missing values with means of cols
        X_new = normalize(X_new) # normalize each col in X_new

        J = Set (rows with missing values in x_new)
        I = Set (rows without missing values in x_new)

        For each j in J:
                sum = 0
                For each i in I:
                        dist = euclead_dist(j, i) # take Euclidian distance between 2 vectors
                        w_ij = 1 / dist
                        sum = ∑w_ij * x_i
                x_new_j = sum / length (I) # replace missing value in X_new
```

Table 7.1. Results from proposed algorithm on missing data.

		Logistic Regression	SVM	Random Forest	Gradient Boosting
Accuracy with mean replacement of missing values	Train	97.01	97.04	97.13	97.10
	Test	96.99	97.02	97.07	97.08
Accuracy with our new replacement of missing values	Train	97.05	97.09	97.18	97.20
	Test	96.98	97.03	97.17	97.14

7.2.2 Neuroevolution

Referring to our work on the neuroevolution, in future, we strive to establish clear analogies between hyper-parameters and the performance metrics which will allow us to achieve the desired performance by tuning the hyper parameters

7.2.3 Neuroevolution Using FIEA

Dealing with the real-world problems related to multiple objectives, it is desired that users will have different priorities for different objectives, and therefore during optimization, it may be preferred to obtain solutions that are biased towards a particular objective. User

preference should therefore be inducted before beginning the optimization. The preliminary experiments conducted indicate that user preference can be inducted in the proposed method by altering the mean and standard deviation of the Gaussian MFs that represent objective functions, reducing the width of dominant fronts in the solution space which may penalize the preferred objective function values and bias the optimization. It will be interesting to observe incorporation of user preference on an a priori basis in the proposed fuzzy inspired genetic algorithm. The other challenge posed by contemporary EAs is exploration of the complete PF solutions. It has been reported [68] that it is difficult with existing EAs to obtain the PF's extreme end solutions, and future research will explore extreme end solutions on the Pareto front and induction of user preferences.

7.2.4 Application platform

To develop a cloud/web service which can be implemented by clinicians for routine use for breast cancer prognosis assessment at the point of care level.

8 Bibliography

[1] F. Bray, J. Ferlay, I. Soerjomataram, R. L. Siegel, L. A. Torre, and A. Jemal, "Global cancer statistics 2018: GLOBOCAN estimates of incidence and mortality worldwide for 36 cancers in 185 countries," *CA. Cancer J. Clin.*, vol. 68, no. 6, pp. 394–424, 2018.

[2] American Cancer Association, "Estimated number of new cancer cases, all EU countries, 2018," 2018.

[3] 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. Treat.*, vol. 172, no. 3, pp. 703–712, 2018.

[4] American Cancer Society, "Breast Cancer Facts & Figures 2012-2014," *Breast Cancer Facts Fig.*, pp. 1–44, 2013.

[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, 2018.

[6] L. A. Byers, "Molecular Profiling," *Lung Cancer Fourth Ed.*, vol. 1606, no. c, pp. 48–66, 2014.

[7] M. Vahdaninia, S. Omidvari, and A. Montazeri, "What do predict anxiety and depression in breast cancer patients? A follow-up study," *Soc. Psychiatry Psychiatr. 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 after a cancer diagnosis in the elderly," *Cancer Nurs.*, vol. 31, no. 5, pp. 334–344, 2008.

[9] K. Park, A. Ali, D. Kim, Y. An, M. Kim, and H. Shin, "Robust predictive model for evaluating breast cancer survivability," *Eng. Appl. Artif. Intell.*, vol. 26, no. 9, pp. 2194–2205, 2013.

[10] D. Kalaitzopoulos, "The potential of Precision Medicine," *New Horizons Transl. Med.*, vol. 3, no. 2, pp. 63–65, 2016.

[11] J. Yoon, C. Davtyan, and M. van der Schaar, "Discovery and Clinical Decision Support for Personalized Healthcare," *IEEE J. Biomed. Heal. Informatics*, vol. 21, no. 4, pp. 1133–1145, Jul. 2017.

[12] S. E. Dilsizian and E. L. Siegel, "Artificial intelligence in medicine and cardiac imaging: Harnessing big data and advanced computing to provide personalized medical diagnosis

and treatment," *Curr. Cardiol. Rep.*, vol. 16, no. 1, 2014.

[13] E. A. Eisenhauer *et al.*, "New response evaluation criteria in solid tumours: Revised RECIST guideline (version 1.1)," *Eur. J. Cancer*, vol. 45, no. 2, pp. 228–247, Jan. 2009.

[14] M. A. Makary and M. Daniel, "Medical error-the third leading cause of death in the US," *BMJ*, vol. 353, no. May, pp. 1–5, 2016.

[15] K. Kourou, T. P. Exarchos, K. P. Exarchos, M. V. Karamouzis, and D. I. Fotiadis, "Machine learning applications in cancer prognosis and prediction," *Comput. Struct. Biotechnol. J.*, vol. 13, pp. 8–17, 2015.

[16] J. A. Cruz and D. S. Wishart, "Applications of machine learning in cancer prediction and prognosis," *Cancer Inform.*, vol. 2, pp. 59–77, 2006.

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

[18] J. Liu *et al.*, "An End-to-End Deep Learning Histochemical Scoring System for Breast Cancer TMA," *IEEE Trans. Med. Imaging*, vol. 38, no. 2, pp. 617–628, 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. Informatics*, vol. 14, no. 7, pp. 3205–3215, 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, 2018.

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

[22] S. Purushotham, C. Meng, Z. Che, and Y. Liu, "Benchmarking deep learning models on large healthcare datasets," *J. Biomed. Inform.*, vol. 83, no. March, pp. 112–134, 2018.

[23] M. Hall, "Correlation-based Feature Selection for Machine Learning," *Methodology*, vol. 21i195-i20, no. April, pp. 1–5, 1999.

[24] J. R. Quinlan, "Induction of Decision Trees," *Mach. Learn.*, vol. 1, no. 1, pp. 81–106, 1986.

[25] I. Kononenko, "Estimating attributes: Analysis and extensions of RELIEF," pp. 171–182, 1994.

[26] I. Guyon and A. Elisseeff, "An Introduction to Variable and Feature Selection," *J. Mach. Learn. Res.*, vol. 3, no. 3, pp. 1157–1182, 2003.

[27] I. Guyon and A. Elisseeff, "Feature Extraction, Foundations and Applications: An

introduction to feature extraction," *Stud. Fuzziness Soft Comput.*, vol. 207, pp. 1–25, 2006.

[28] J. J. Hughey and A. J. Butte, "Robust meta-analysis of gene expression using the elastic net," *Nucleic Acids Res.*, vol. 43, no. 12, pp. 1–11, 2015.

[29] Y. Saeys, I. Inza, and P. Larra??aga, "A review of feature selection techniques in bioinformatics," *Bioinformatics*, vol. 23, no. 19, pp. 2507–2517, 2007.

[30] A. C. Haury, P. Gestraud, and J. P. Vert, "The influence of feature selection methods on accuracy, stability and interpretability of molecular signatures," *PLoS One*, vol. 6, no. 12, pp. 1–12, 2011.

[31] G. Brown, A. Pocock, M.-J. Zhao, and M. Lujan, "Conditional Likelihood Maximisation: A Unifying Framework for Mutual Information Feature Selection," *J. Mach. Learn. Res.*, vol. 13, pp. 27–66, 2012.

[32] J. Tang, S. Alelyani, and H. Liu, "Feature Selection for Classification: A Review," *Data Classif. Algorithms Appl.*, pp. 37–64, 2014.

[33] J. Gui, Z. Sun, S. Ji, S. Member, D. Tao, and T. Tan, "Feature Selection Based on Structured Sparsity : A Comprehensive Study," *IEEE Trans. Neural Networks Learn. Syst.*, vol. 28, no. 7, pp. 1–18, 2016.

[34] M. H. Zena and D. F. Gillies, "A Review of Feature Selection and Feature Extraction Methods Applied on Microarray Data," *Adv. Bioinformatics*, vol. 2015, no. 1, 2015.

[35] G. Doquire and M. Verleysen, "Mutual information based feature selection for mixed data," *Esann*, no. April, pp. 27–29, 2011.

[36] D. Beaton, M. Kriegsman, J. Dunlop, F. M. Filbey, and H. Abdi, "Imaging genetics with partial least squares for mixed-data types (MiMoPLS)," *Springer Proc. Math. Stat.*, vol. 173, pp. 73–91, 2016.

[37] M. Chavent, V. Kuentz-Simonet, A. Labenne, and J. Saracco, "Multivariate analysis of mixed data : The PCAmixdata R package," *arXiv e-prints*, pp. 1–31, 2014.

[38] J. Pagès, *Multiple Factor Analysis by Example Using R*. 2015.

[39] X. Zhang, C. Mei, D. Chen, and J. Li, "Feature selection in mixed data: A method using a novel fuzzy rough set-based information entropy," *Pattern Recognit.*, vol. 56, pp. 1–15, 2016.

[40] G. Doquire and M. Verleysen, "an Hybrid Approach To Feature Selection for Mixed Categorical and Continuous Data," *Proc. Int. Conf. Knowl. Discov. Inf. Retr.*, pp. 394–401, 2011.

[41] D. R. Wilson and T. R. Martinez, “Improved heterogeneous distance functions,” *J. Artif. Intell. Res.*, vol. 6, pp. 1–34, 1997.

[42] W. Tang and K. Z. Mao, “Feature selection algorithm for mixed data with both nominal and continuous features,” *Pattern Recognit. Lett.*, vol. 28, no. 5, pp. 563–571, 2007.

[43] C. C. Hsu and W. H. Huang, “Integrated dimensionality reduction technique for mixed-type data involving categorical values,” *Appl. Soft Comput. J.*, vol. 43, pp. 199–209, 2016.

[44] C. Guo and F. Berkhahn, “Entity Embeddings of Categorical Variables,” no. 1, pp. 1–9, 2016.

[45] F. Wu, Y. Yuan, and Y. Zhuang, “Heterogeneous feature selection by group lasso with logistic regression,” *... Int. Conf. Multimed.*, pp. 983–986, 2010.

[46] G. E. Hinton, “Learning distributed representations of concepts.”

[47] Y. Kim, Y. Jernite, D. Sontag, and A. M. Rush, “Character-aware neural language models,” *Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence*. AAAI Press, pp. 2741–2749, 2016.

[48] R. Sennrich, B. Haddow, and A. Birch, “Neural Machine Translation of Rare Words with Subword Units,” in *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 2016, vol. 1, pp. 1715–1725.

[49] X. Zhang, J. Zhao, and Y. Lecun, “Character-level Convolutional Networks for Text Classification *.”

[50] T. Mikolov, K. Chen, G. Corrado, and J. Dean, “Efficient Estimation of Word Representations in Vector Space,” pp. 1–12, 2013.

[51] J. Pennington, R. Socher, and C. Manning, “Glove: Global Vectors for Word Representation,” *Proc. 2014 Conf. Empir. Methods Nat. Lang. Process.*, pp. 1532–1543, 2014.

[52] P. Bojanowski, E. Grave, A. Joulin, and T. Mikolov, “Enriching Word Vectors with Subword Information,” *Trans. Assoc. Comput. Linguist.*, vol. 5, no. 1, pp. 135–146, 2017.

[53] K. Zhang *et al.*, “From Categorical to Numerical: Multiple Transitive Distance Learning and Embedding,” *Proc. 2015 SIAM Int. Conf. Data Min.*, pp. 46–54, 2015.

[54] D. Ravi *et al.*, “Deep Learning for Health Informatics,” *IEEE J. Biomed. Heal. Informatics*, vol. 21, no. 1, pp. 4–21, 2017.

[55] T. Praczyk, “Cooperative co–evolutionary neural networks,” *J. Intell. Fuzzy Syst.*, vol.

30, no. 5, pp. 2843–2858, Apr. 2016.

[56] B. Shabash and K. C. Wiese, “EvoNN - A Customizable Evolutionary Neural Network with Heterogenous Activation Functions,” *GECCO ’18 Companion*, pp. 1449–1456, 2018.

[57] K. Deb and H. Jain, “An Evolutionary Many-Objective Optimization Algorithm Using Reference-point Based Non-dominated Sorting Approach, Part I: Solving Problems with Box Constraints,” *Ieeexplore.Ieee.Org*, vol. 18, no. c, pp. 1–1, 2013.

[58] J. M. Kontoleon, “Optimum Link Allocation of Fixed Topology Networks,” *IEEE Trans. Reliab.*, vol. R-28, no. 2, pp. 145–147, 1979.

[59] X. Yao, “Evolving artificial neural networks,” *Proc. IEEE*, vol. 87, no. 9, pp. 1423–1447, 1999.

[60] L. Rachmawati and D. Srinivasan, “Incorporating the notion of relative importance of objectives in evolutionary multiobjective optimization,” *IEEE Trans. Evol. Comput.*, vol. 14, no. 4, pp. 530–546, 2010.

[61] K. Deb, “Multi-Objective Optimization Using Evolutionary Algorithms: An Introduction Multi-Objective Optimization Using Evolutionary Algorithms: An Introduction,” *Wiley-Interscience Ser. Syst. Optim.*, no. June, pp. 1–24, 2001.

[62] C. A. Coello Coello, “Multi-objective evolutionary algorithms in real-world applications: Some recent results and current challenges,” *Comput. Methods Appl. Sci.*, vol. 36, pp. 3–18, 2015.

[63] H. Y. Park, A. Datta-Gupta, and M. J. King, “Handling conflicting multiple objectives using Pareto-based evolutionary algorithm during history matching of reservoir performance,” *J. Pet. Sci. Eng.*, vol. 125, pp. 48–66, 2015.

[64] C. J. Carmona, P. González, M. J. Del Jesus, and F. Herrera, “Non-dominated multi-objective evolutionary algorithm based on fuzzy rules extraction for subgroup discovery,” *Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics)*, vol. 5572 LNAI, pp. 573–580, 2009.

[65] R. R. Chan and S. D. Sudhoff, “An Evolutionary computing approach to robust design in the presence of uncertainties,” *IEEE Trans. Evol. Comput.*, vol. 14, no. 6, pp. 900–912, 2010.

[66] E. Zitzler, “Two decades of evolutionary multi-criterion optimization: A glance back and a look ahead,” *Proc. 2007 IEEE Symp. Comput. Intell. Multicriteria Decis. Making, MCDM 2007*, no. Mcdm, p. 318, 2007.

[67] J. Bader and E. Zitzler, “HypE : An algorithm for fast optimization,” *Evol. Comput.*, vol. 19, no. 1, pp. 45–76, 2011.

[68] X. Zou, Y. Chen, M. Liu, and L. Kang, “A new evolutionary algorithm for solving many-objective optimization problems,” *IEEE Trans. Syst. Man, Cybern. Part B Cybern.*, vol. 38, no. 5, pp. 1402–1412, 2008.

[69] H. Burke, H. Burke, D. Rosen, D. Rosen, P. Goodman, and P. Goodman, “Comparing the prediction accuracy of artificial neural networks and other statistical models,” *Adv. Neural Inf. Process. Syst.*, pp. 2213–2216, 1995.

[70] H. B. Burke *et al.*, “Artificial neural networks improve the accuracy of cancer survival prediction,” *Cancer*, vol. 79, no. 4, pp. 857–862, 1997.

[71] H. B. Burke, “Artificial neural networks for cancer research: Outcome prediction,” *Semin. Surg. Oncol.*, vol. 10, no. 1, pp. 73–79, 1994.

[72] O. Gevaert, F. De Smet, D. Timmerman, Y. Moreau, and B. De Moor, “Predicting the prognosis of breast cancer by integrating clinical and microarray data with Bayesian networks,” *Bioinformatics*, vol. 22, no. 14, pp. 184–190, 2006.

[73] D. Delen, G. Walker, and A. Kadam, “Predicting breast cancer survivability: A comparison of three data mining methods,” *Artif. Intell. Med.*, vol. 34, no. 2, pp. 113–127, 2005.

[74] A. H. Chen, G. T. Chen, J. C. Hsieh, and C. H. Lin, “BCPP: An intelligent prediction system of breast cancer prognosis using microarray and clinical data,” *2009 WRI World Congr. Comput. Sci. Inf. Eng. CSIE 2009*, vol. 5, pp. 28–32, 2009.

[75] A. Ali, U. Khan, A. Tufail, and M. Kim, “Analyzing potential of SVM based classifiers for intelligent and less invasive breast cancer prognosis,” *2010 2nd Int. Conf. Comput. Eng. Appl. ICCEA 2010*, vol. 2, pp. 313–319, 2010.

[76] R. S. Olson, W. La Cava, Z. Mustahsan, A. Varik, and J. H. Moore, “Data-driven Advice for Applying Machine Learning to Bioinformatics Problems,” 2017.

[77] D. M. Soria Jonathan Garibaldi, E. Biganzoli, and I. O. Ellis, “A Comparison of Three Different Methods for Classification of Breast Cancer Data,” *Mach. Learn. Appl.*, vol. 2008, pp. 11–13, 2008.

[78] E. D. Übeyli, “Implementing automated diagnostic systems for breast cancer detection,” *Expert Syst. Appl.*, vol. 33, no. 4, pp. 1054–1062, Nov. 2007.

[79] O. F. Baker and S. A. Kareem, “ANFIS models for prognostic and survival rate analysis ‘nasopharyngeal carcinoma’.,” in *2008 4th IEEE International Conference on*

Management of Innovation and Technology, 2008, pp. 537–541.

[80] H. Hamdan and J. M. Garibaldi, "Adaptive neuro-fuzzy inference system (ANFIS) in modelling breast cancer survival," *Fuzzy Syst. (FUZZ), 2010 IEEE Int. Conf.*, pp. 1–8, 2010.

[81] J. M. Garibaldi, S.-M. Zhou, X.-Y. Wang, R. I. John, and I. O. Ellis, "Incorporation of expert variability into breast cancer treatment recommendation in designing clinical protocol guided fuzzy rule system models," *J. Biomed. Inform.*, vol. 45, no. 3, pp. 447–459, Jun. 2012.

[82] S.-W. Chang, S. Abdul-Kareem, A. F. Merican, and R. B. Zain, "Oral cancer prognosis based on clinicopathologic and genomic markers using a hybrid of feature selection and machine learning methods.," *BMC Bioinformatics*, vol. 14, no. 1, p. 170, 2013.

[83] I. Kononenko, "ReliefF for estimation and discretization of attributes in classification, regression, and ILP problems," *Artif. Intell. …*, pp. 1–15, 1996.

[84] R. Tibshirani, "Regression Shrinkage and Selection via the Lasso," 1996.

[85] H. Zou and T. Hastie, "Regularization and variable selection via the elastic-net," *J. R. Stat. Soc.*, vol. 67, no. 2, pp. 301–320, 2005.

[86] E. Bair, T. Hastie, D. Paul, and R. Tibshirani, "Prediction by Supervised Principal Components," *J. Am. Stat. Assoc.*, vol. 101, no. 473, pp. 119–137, 2006.

[87] S. Sharifzadeh, A. Ghodsi, L. H. Clemmensen, and B. K. Ersbøll, "Sparse supervised principal component analysis (SSPCA) for dimension reduction and variable selection," *Eng. Appl. Artif. Intell.*, vol. 65, no. February, pp. 168–177, 2017.

[88] B. Ding and R. Gentleman, "Classification Using Generalized Partial Least Squares," *J. Comput. Graph. Stat.*, vol. 14, no. 2, pp. 280–298, 2005.

[89] A.-L. Boulesteix and K. Strimmer, "Partial least squares: a versatile tool for the analysis of high-dimensional genomic data," *Brief. Bioinform.*, vol. 8, no. 1, pp. 32–44, 2006.

[90] Q. Hu, D. Yu, J. Liu, and C. Wu, "Neighborhood rough set based heterogeneous feature subset selection," *Inf. Sci. (Ny).*, vol. 178, no. 18, pp. 3577–3594, 2008.

[91] Q. Hu, J. Liu, and D. Yu, "Mixed feature selection based on granulation and approximation," *Knowledge-Based Syst.*, vol. 21, no. 4, pp. 294–304, 2008.

[92] D. Chen and Y. Yang, "Attribute Reduction for Heterogeneous Data Based on the Combination of Classical and Fuzzy Rough Set Models," *Fuzzy Syst. IEEE Trans.*, vol. 22, no. 5, pp. 1325–1334, 2014.

[93] K. O. Stanley and R. Miikkulainen, "Evolving Neural Networks through Augmenting

Topologies Please Note :," vol. 10, no. 2, p. 78712, 2002.

[94] J. E. Auerbach and J. C. Bongard, "Evolving Complete Robots with CPPN-NEAT: The Utility of Recurrent Connections," *Gecco-2011 Proc. 13th Annu. Genet. Evol. Comput. Conf.*, pp. 1475–1482, 2011.

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

[96] 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, 1997.

[97] V. Khare, X. Yao, and K. Deb, "Performance Scaling of Multi-objective," *Computer (Long. Beach. Calif).*, pp. 376–390.

[98] R. C. Purshouse, P. J. Fleming, and M. S. Sheffield, "Evolutionary Many-Objective Optimisation : An Exploratory Analysis," pp. 2066–2073, 2003.

[99] W. Sheng, K. Y. Liu, Y. Liu, X. Meng, and Y. Li, "Optimal Placement and Sizing of Distributed Generation via an Improved Nondominated Sorting Genetic Algorithm II," *IEEE Trans. Power Deliv.*, vol. 30, no. 2, pp. 569–578, 2015.

[100] M. Li, S. Yang, K. Li, and X. Liu, "Evolutionary Algorithms With Segment-Based Search for Multiobjective Optimization Problems," *IEEE Trans. Cybern.*, vol. 44, no. 8, pp. 1295–1313, Aug. 2014.

[101] H. Sato, H. E. Aguirre, and K. Tanaka, "Controlling dominance area of solutions and its impact on the performance of MOEAs," *Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics)*, vol. 4403 LNCS, pp. 5–20, 2007.

[102] N. Drechsler and R. Drechsler, "Robust Multi-Objective Optimization in High," pp. 715–726, 2007.

[103] M. Koppen and K. Yoshida, "Substitute distance assignments in NSGA-II for handling many-objective optimization problems," *Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics)*, vol. 4403 LNCS, pp. 727–741, 2007.

[104] T. Wagner, N. Beume, and B. Naujoks, "Methods in Many-Objective Optimization," pp. 742–743, 2007.

[105] J. Cheng, G. G. Yen, and G. Zhang, "A grid-based adaptive multi-objective differential evolution algorithm," *Inf. Sci. (Ny).*, vol. 367–368, pp. 890–908, Nov. 2016.

[106] X. Zhang, Y. Tian, and Y. Jin, "Approximate non-dominated sorting for evolutionary many-objective optimization," *Inf. Sci. (Ny).*, vol. 369, pp. 14–33, 2016.

[107] K. Nag, T. Pal, and N. R. Pal, "ASMiGA: An archive-based steady-state micro genetic algorithm," *IEEE Trans. Cybern.*, vol. 45, no. 1, pp. 40–52, 2015.

[108] S. Fettaka, J. Thibault, and Y. Gupta, "A new algorithm using front prediction and NSGA-II for solving two and three-objective optimization problems," *Optim. Eng.*, vol. 16, no. 4, pp. 713–736, Dec. 2015.

[109] X. Wang, L. Tang, and X. Wang, "A Hybrid Multiobjective Evolutionary Algorithm for Multiobjective Optimization Problems A Hybrid Multiobjective Evolutionary Algorithm for Multiobjective Optimization Problems," *IEEE Trans. Evol. Comput.*, vol. 17, no. FEBRUARY 2013, pp. 20–45, 2015.

[110] 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, 2014.

[111] 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, 2014.

[112] T. Goel and N. Stander, "A non-dominance-based online stopping criterion for multi-objective evolutionary algorithms," *Int. J. Numer. Methods Eng.*, vol. 84, no. 6, pp. 661–684, Nov. 2010.

[113] S. Sharma and G. P. Rangaiah, "An improved multi-objective differential evolution with a termination criterion for optimizing chemical processes," *Comput. Chem. Eng.*, vol. 56, pp. 155–173, 2013.

[114] B. J. Reardon, "Fuzzy logic versus niched Pareto multiobjective genetic algorithm optimization," *Model. Simul. Mater. Sci. Eng.*, vol. 6, no. 6, pp. 717–734, Nov. 1998.

[115] A. Mukhopadhyay, U. Maulik, and S. Bandyopadhyay, "A survey of multiobjective evolutionary clustering," *ACM Comput. Surv.*, vol. 47, no. 4, 2015.

[116] E. Fernandez, N. Cancela, and R. Olmedo, "Deriving a final ranking from fuzzy preferences: An approach compatible with the Principle of Correspondence," *Math. Comput. Model.*, vol. 47, no. 1–2, pp. 218–234, 2008.

[117] Y. Liu, J. Sun, and S. Wang, "The concept of approximation based on fuzzy dominance relation in decision-making," *Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics)*, vol. 2639, pp. 382–385, 2003.

[118] K. Deb, M. Abouhawwash, and J. Dutta, "An optimality theory based proximity measure

for evolutionary multi-objective and many-objective optimization," *Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics)*, vol. 9019, pp. 18–33, 2015.

[119] K. Sindhya, K. Miettinen, and K. Deb, "A hybrid framework for evolutionary multi-objective optimization," *IEEE Trans. Evol. Comput.*, vol. 17, no. 4, pp. 495–511, 2013.

[120] L. Martí, J. García, A. Berlanga, and J. M. Molina, "A stopping criterion for multi-objective optimization evolutionary algorithms," *Inf. Sci. (Ny).*, vol. 367–368, pp. 700–718, 2016.

[121] C. Kahraman, S. Birgün, and V. Z. Yenen, "Fuzzy Multi-Attribute Scoring Methods with Applications," Springer, Boston, MA, 2008, pp. 187–208.

[122] V. Yepes, T. García-Segura, and J. M. Moreno-Jiménez, "A cognitive approach for the multi-objective optimization of RC structural problems," *Arch. Civ. Mech. Eng.*, vol. 15, no. 4, pp. 1024–1036, 2015.

[123] C. A. C. Coello, "Handling preferences in evolutionary multiobjective optimization: A survey," *Proc. 2000 Congr. Evol. Comput. CEC 2000*, vol. 1, pp. 30–37, 2000.

[124] K. Deb and S. Tiwari, "Multi-objective optimization of a leg mechanism using genetic algorithms," *Eng. Optim.*, vol. 37, no. 4, pp. 325–350, 2005.

[125] 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, 2007.

[126] L. Ke, Q. Zhang, and R. Battiti, "MOEA/D-ACO: A multiobjective evolutionary algorithm using decomposition and AntColony," *IEEE Trans. Cybern.*, vol. 43, no. 6, pp. 1845–1859, 2013.

[127] X. Cai, Y. Li, Z. Fan, and Q. Zhang, "An External Archive Guided Multiobjective Evolutionary Algorithm Based on Decomposition for Combinatorial Optimization," *IEEE Trans. Evol. Comput.*, vol. 19, no. 4, pp. 508–523, 2015.

[128] K. Li, S. Kwong, Q. Zhang, and K. Deb, "Interrelationship-based selection for decomposition multiobjective optimization," *IEEE Trans. Cybern.*, vol. 45, no. 10, pp. 2076–2088, 2015.

[129] J. Andreu-Perez, C. C. Y. Poon, R. D. Merrifield, S. T. C. Wong, and G.-Z. Yang, "Big Data for Health," *IEEE J. Biomed. Heal. Informatics*, vol. 19, no. 4, pp. 1193–1208, Jul. 2015.

[130] M. Viceconti, P. Hunter, and R. Hose, "Big Data, Big Knowledge: Big Data for Personalized Healthcare," *IEEE J. Biomed. Heal. Informatics*, vol. 19, no. 4, pp. 1209–

1215, Jul. 2015.

[131] UCI, “UCI Machine Learning Repository: Breast Cancer Wisconsin (Diagnostic) Data Set,” *http://archive.ics.uci.edu/ml/datasets/Breast+Cancer+Wisconsin+%2528Diagnostic%2529*. 2011.

[132] Y. Fong, J. Evans, D. Brook, J. Kenkre, P. Jarvis, and K. Gower-Thomas, “The Nottingham Prognostic Index: Five- and ten-year data for all-cause survival within a screened population,” *Ann. R. Coll. Surg. Engl.*, vol. 97, no. 2, pp. 137–139, 2015.

[133] National Cancer Institute, “SEER Incidence Database - SEER Data & Software,” 2017. [Online]. Available: https://seer.cancer.gov/data/.

[134] 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 *2015 IEEE 15th International Conference on Bioinformatics and Bioengineering (BIBE)*, 2015, pp. 1–6.

[135] J. Thongkam, G. Xu, and Y. Zhang, “AdaBoost algorithm with random forests for predicting breast cancer survivability,” *Proc. Int. Jt. Conf. Neural Networks*, pp. 3062–3069, 2008.

[136] S. Møller *et al.*, “The clinical database and the treatment guidelines of the Danish Breast Cancer Cooperative Group (DBCG); its 30-years experience and future promise,” *Acta Oncol. (Madr).*, vol. 47, no. 4, pp. 506–524, 2008.

[137] B. S. Ko *et al.*, “Changing patterns in the clinical characteristics of Korean Breast Cancer from 1996-2010 using an online nationwide breast cancer database,” *J. Breast Cancer*, vol. 15, no. 4, pp. 393–400, 2012.

[138] S. Eaker, P. W. Dickman, L. Bergkvist, and L. Holmberg, “Differences in management of older women influence breast cancer survival: Results from a population-based database in Sweden,” *PLoS Med.*, vol. 3, no. 3, pp. 321–328, 2006.

[139] X. Jiang, A. Wells, A. Brufsky, and R. Neapolitan, “A clinical decision support system learned from data to personalize treatment recommendations towards preventing breast cancer metastasis,” *PLoS One*, vol. 14, no. 3, pp. 1–3, 2019.

[140] M. Yi *et al.*, “Novel staging system for predicting disease-specific survival in patients with breast cancer treated with surgery as the first intervention: Time to modify the current American Joint Committee on Cancer staging system,” *J. Clin. Oncol.*, vol. 29, no. 35, pp. 4654–4661, 2011.

[141] C. C. Chang and S. H. Chen, “Developing a Novel Machine Learning-Based Classification Scheme for Predicting SPCs in Breast Cancer Survivors,” *Front. Genet.*, vol. 10, no. September, pp. 1–6, 2019.

[142] C. Ming, V. Viassolo, N. Probst-Hensch, P. O. Chappuis, I. D. Dinov, and M. C. Katapodi, “Machine learning techniques for personalized breast cancer risk prediction: Comparison with the BCRAT and BOADICEA models,” *Breast Cancer Res.*, vol. 21, no. 1, pp. 1–11, 2019.

[143] A. LG and E. AT, “Using Three Machine Learning Techniques for Predicting Breast Cancer Recurrence,” *J. Heal. Med. Informatics*, vol. 04, no. 02, pp. 2–4, 2013.

[144] M. Slaoui, F. Z. Mouh, I. Ghanname, R. Razine, M. El Mzibri, and M. Amrani, “Outcome of breast cancer in Moroccan young women correlated to clinic-pathological features, risk factors and treatment: A comparative study of 716 cases in a single institution,” *PLoS One*, vol. 11, no. 10, pp. 2–4, 2016.

[145] P. Ferroni, F. M. Zanzotto, S. Riondino, N. Scarpato, F. Guadagni, and M. Roselli, “Breast cancer prognosis using a machine learning approach,” *Cancers (Basel).*, vol. 11, no. 3, pp. 1–9, 2019.

[146] P. J. García-Laencina, P. H. Abreu, M. H. Abreu, and N. Afonoso, “Missing data imputation on the 5-year survival prediction of breast cancer patients with unknown discrete values,” *Comput. Biol. Med.*, vol. 59, pp. 125–133, 2015.

[147] B. J. Hearne, M. D. Teare, M. Butt, and L. Donaldson, “Comparison of nottingham prognostic index and adjuvant online prognostic tools in young women with breast cancer: Review of a single-institution experience,” *BMJ Open*, vol. 5, no. 1, pp. 1–3, 2015.

[148] S. K. M. Hamouda, M. E. Wahed, R. H. Abo Alez, and K. Riad, “Robust breast cancer prediction system based on rough set theory at National Cancer Institute of Egypt,” *Comput. Methods Programs Biomed.*, vol. 153, pp. 259–268, 2018.

[149] S. Hegselmann, L. Greulich, J. Varghese, and M. Dugas, “Reproducible Survival Prediction with SEER Cancer Data,” *Mlhc-2018*, no. 2017, pp. 1–26, 2018.

[150] H. Lotfnezhad Afshar, M. Ahmadi, M. Roudbari, and F. Sadoughi, “Prediction of Breast Cancer Survival Through Knowledge Discovery in Databases,” *Glob. J. Health Sci.*, vol. 7, no. 4, pp. 392–398, Jan. 2015.

[151] H. M. Zolbanin, D. Delen, and A. Hassan Zadeh, “Predicting overall survivability in comorbidity of cancers: A data mining approach,” *Decis. Support Syst.*, vol. 74, pp. 150–

161, Jun. 2015.

[152] N. Shukla, M. Hagenbuchner, K. T. Win, and J. Yang, “Breast cancer data analysis for survivability studies and prediction,” *Comput. Methods Programs Biomed.*, vol. 155, pp. 199–208, 2018.

[153] H. Lu, H. Wang, and S. W. Yoon, “A dynamic gradient boosting machine using genetic optimizer for practical breast cancer prognosis,” *Expert Syst. Appl.*, vol. 116, pp. 340–350, 2019.

[154] K.-M. Wang, B. Makond, W.-L. Wu, K.-J. Wang, and Y. S. Lin, “Optimal Data Mining Method for Predicting Breast Cancer Survivability,” *Int. J. Innov. Manag. Inf. Prod.*, vol. 4, no. 2, pp. 28–33, 2013.

[155] M. Bozorgi, K. Taghva, and A. Singh, “Cancer survivability with logistic regression,” *Proc. Comput. Conf. 2017*, vol. 2018-Janua, no. July, pp. 416–420, 2018.

[156] A. Bellaachia and E. Guven, “Predicting Breast Cancer Survivability using Data Mining Techniques,” *Age (Omaha).*, vol. 58, 2006.

[157] H. Wang, B. Zheng, S. W. Yoon, and H. S. Ko, “A support vector machine-based ensemble algorithm for breast cancer diagnosis,” *Eur. J. Oper. Res.*, vol. 267, no. 2, pp. 687–699, 2018.

[158] K. J. Wang, B. Makond, and K. M. Wang, “An improved survivability prognosis of breast cancer by using sampling and feature selection technique to solve imbalanced patient classification data,” *BMC Med. Inform. Decis. Mak.*, vol. 13, no. 1, 2013.

[159] S. Hussain, N. Z. Quazilbash, S. Bai, and S. Khoja, “Reduction of Variables for Predicting Breast Cancer Survivability Using Principal Component Analysis,” *2015 IEEE 28th Int. Symp. Comput. Med. Syst.*, pp. 131–134, 2015.

[160] S. C. Formenti, A. A. Arslan, and S. M. Love, “Global Breast Cancer: The Lessons to Bring Home,” *Int. J. Breast Cancer*, vol. 2012, pp. 1–7, 2012.

[161] W. L. Donegan, “Tumor-related prognostic factors for breast cancer.,” *CA. Cancer J. Clin.*, vol. 47, no. 1, pp. 28–51, 1997.

[162] B. Fisher, N. H. Slack, and I. D. F. Bross, “Cancer of the breast: Size of neoplasm and prognosis,” *Cancer*, vol. 24, no. 5, pp. 1071–1080, 1969.

[163] H. Abdi and L. J. Williams, “Principal component analysis,” *Wiley Interdiscip. Rev. Comput. Stat.*, vol. 2, no. 4, pp. 433–459, 2010.

[164] S. Maitra and J. Yan, “Principle Component Analysis and Partial Least Squares :,” *Not A Pap.*, pp. 79–90, 2008.

[165] I. T. Jolliffe, J. Cadima, and J. Cadima, “Principal component analysis : a review and recent developments Subject Areas : Author for correspondence :,” 2016.

[166] H. Abdi, “Multiple correspondence analysis,” *Encycl. Meas. Stat.*, vol. 95, no. 2, pp. 116–28, 2007.

[167] D. H. Wolpert and W. G. Macready, “No free lunch theorems for optimization,” *IEEE Trans. Evol. Comput.*, vol. 1, no. 1, pp. 67–82, 1997.

[168] J. Braeken and M. A. L. M. Van Assen, “An empirical Kaiser criterion.,” *Psychol. Methods*, vol. 22, no. 3, pp. 450–466, 2017.

[169] H. F. Kaiser, “the Application of Electronic Computers To Factor Analysi,” *Measurement*, vol. XX, no. 1, pp. 141–151, 1960.

[170] H. Yu, J. Yang, and J. Han, “Classifying large data sets using SVMs with hierarchical clusters,” in *Proceedings of the ninth ACM SIGKDD international conference on Knowledge discovery and data mining - KDD ’03*, 2003, p. 306.

[171] T. N. Do and F. Poulet, “MINING VERY LARGE DATASETS WITH SVM AND VISUALIZATION,” in *Proceedings of the Seventh International Conference on Enterprise Information Systems*, 2005, pp. 127–134.

[172] B. Abdikenov, Z. Iklassov, A. Sharipov, S. Hussain, and P. K. Jamwal, “Analytics of Heterogeneous Breast Cancer Data Using Neuroevolution,” *IEEE Access*, vol. 7, pp. 18050–18060, 2019.

[173] P. C. Pendharkar, “An empirical study of design and testing of hybrid evolutionary–neural approach for classiÿcation,” *Omega*, vol. 29, pp. 361–374, 2001.

[174] K. Senhaji, H. Ramchoun, and M. Ettaouil, “Multilayer Perceptron: NSGA II for a New Multi-Objective Learning Method for Training and Model Complexity,” *J. Electron. Syst.*, vol. 7, no. 4, pp. 105–114, 2017.

[175] Z. Chen, C. K. Yeo, B. S. Lee, C. T. Lau, and Y. Jin, “Evolutionary multi-objective optimization based ensemble autoencoders for image outlier detection,” *Neurocomputing*, vol. 309, pp. 192–200, 2018.

[176] 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. J.*, vol. 71, pp. 826–834, 2018.

[177] K. Mason, J. Duggan, and E. Howley, “A multi-objective neural network trained with differential evolution for dynamic economic emission dispatch,” *Int. J. Electr. Power Energy Syst.*, vol. 100, no. November 2017, pp. 201–221, 2018.

[178] P. Melin and D. Sánchez, “Multi-objective optimization for modular granular neural networks applied to pattern recognition,” *Inf. Sci. (Ny).*, vol. 460–461, pp. 594–610, 2018.

[179] 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, 2002.

[180] 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,” *Futur. Gener. Comput. Syst.*, vol. 88, pp. 571–585, 2018.

[181] K.-H. Yu *et al.*, “Predicting non-small cell lung cancer prognosis by fully automated microscopic pathology image features,” *Nat. Commun.*, vol. 7, p. 12474, 2016.

[182] F. Cao *et al.*, “An Algorithm for Clustering Categorical Data with Set-Valued Features,” *IEEE Trans. Neural Networks Learn. Syst.*, vol. 29, no. 10, pp. 4593–4606, 2018.

[183] L. A. Zadeh, “Fuzzy sets,” *Inf. Control*, vol. 8, no. 3, pp. 338–353, Jun. 1965.

[184] D. K. Pratihar and N. B. Hui, “Evolution of Fuzzy Controllers and Applications,” in *Advances in Evolutionary Computing for System Design*, L. C. Jain, V. Palade, and D. Srinivasan, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2007, pp. 47–69.

[185] C. J. Carmona, P. González, M. J. Del Jesus, and F. Herrera, “NMEEF-SD: Non-dominated multiobjective evolutionary algorithm for extracting fuzzy rules in subgroup discovery,” *IEEE Trans. Fuzzy Syst.*, vol. 18, no. 5, pp. 958–970, 2010.

[186] F. Optimization, Z. Deng, F. Chung, and S. Wang, “Robust Relief-Feature Weighting , Margin,” vol. 18, no. 4, pp. 726–744, 2010.

[187] R. Alcalá, P. Ducange, F. Herrera, B. Lazzerini, and F. Marcelloni, “A multiobjective evolutionary approach to concurrently learn rule and data bases of linguistic fuzzy-rule-based systems,” *IEEE Trans. Fuzzy Syst.*, vol. 17, no. 5, pp. 1106–1122, 2009.

[188] S. Li and C. Hu, “Two-step interactive satisfactory method for fuzzy multiple objective optimization with preemptive priorities,” *IEEE Trans. Fuzzy Syst.*, vol. 15, no. 3, pp. 417–425, 2007.

[189] R. Abbasnia, A. Afshar, and E. Eshtehardian, “Time-Cost Trade-off Problem in Construction Project Management, Based on Fuzzy Logic,” *J. Appl. Sci.*, vol. 8, no. 22, pp. 4159–4165, Dec. 2008.

[190] E. Fernandez, E. Lopez, S. Bernal, C. A. Coello Coello, and J. Navarro, “Evolutionary multiobjective optimization using an outranking-based dominance generalization,”

Comput. Oper. Res., vol. 37, no. 2, pp. 390–395, Feb. 2010.

[191] G. Wang and J. J. Wu, "A new fuzzy dominance GA applied to solve many-objective optimization problem," *Second Int. Conf. Innov. Comput. Inf. Control. ICICIC 2007*, no. 1, pp. 3–6, 2007.

[192] M. Nasir, A. K. Mondal, S. Sengupta, S. Das, and A. Abraham, "An improved Multiobjective Evolutionary Algorithm based on decomposition with fuzzy dominance," *2011 IEEE Congr. Evol. Comput. CEC 2011*, pp. 765–772, 2011.

[193] K. W. Chau and C. L. Wu, "A hybrid model coupled with singular spectrum analysis for daily rainfall prediction," *J. Hydroinformatics*, vol. 12, no. 4, pp. 458–473, 2010.

[194] W. chuan Wang, K. wing Chau, D. mei Xu, and X. Y. Chen, "Improving Forecasting Accuracy of Annual Runoff Time Series Using ARIMA Based on EEMD Decomposition," *Water Resour. Manag.*, vol. 29, no. 8, pp. 2655–2675, 2015.

[195] R. Taormina and K. W. Chau, "Data-driven input variable selection for rainfall-runoff modeling using binary-coded particle swarm optimization and Extreme Learning Machines," *J. Hydrol.*, vol. 529, pp. 1617–1632, 2015.

[196] C. L. Wu, K. W. Chau, and Y. S. Li, "Methods to improve neural network performance in daily flows prediction," *J. Hydrol.*, vol. 372, no. 1–4, pp. 80–93, Jun. 2009.

[197] J. Zhang and K. W. Chau, "Multilayer ensemble pruning via novel multi-sub-swarm particle swarm optimization," *J. Univers. Comput. Sci.*, vol. 15, no. 4, pp. 840–858, 2009.

[198] S. Zhang and K.-W. Chau, "Dimension Reduction Using Semi-Supervised Locally Linear Embedding for Plant Leaf Classification," in *Emerging Intelligent Computing Technology and Applications*, 2009, pp. 948–955.

[199] J. Feng and L. Kong, "A Fuzzy Multi-objective Genetic Algorithm for QoS-based Cloud Service Composition," *Proc. - 2015 11th Int. Conf. Semant. Knowl. Grids, SKG 2015*, pp. 202–206, 2016.

[200] Z. He, G. G. Yen, and J. Zhang, "Fuzzy-based pareto optimality for many-objective evolutionary algorithms," *IEEE Trans. Evol. Comput.*, vol. 18, no. 2, pp. 269–285, 2014.

[201] A. Starkey, H. Hagras, S. Shakya, and G. Owusu, "A multi-objective genetic type-2 fuzzy logic based system for mobile field workforce area optimization," *Inf. Sci. (Ny).*, vol. 329, pp. 390–411, Feb. 2016.

[202] X. Zhao, L. Shen, X. Peng, and W. Zhao, "Toward SLA-constrained service composition: An approach based on a fuzzy linguistic preference model and an

evolutionary algorithm," *Inf. Sci. (Ny).*, vol. 316, pp. 370–396, Sep. 2015.

[203] P. K. Jamwal and S. Hussain, "Multicriteria Design Optimization of a Parallel Ankle Rehabilitation Robot: Fuzzy Dominated Sorting Evolutionary Algorithm Approach," *IEEE Trans. Syst. Man, Cybern. Syst.*, vol. 46, no. 5, pp. 589–597, 2016.

[204] P. K. Jamwal, S. Q. Xie, Y. H. Tsoi, and K. C. Aw, "Forward kinematics modelling of a parallel ankle rehabilitation robot using modified fuzzy inference," *Mech. Mach. Theory*, vol. 45, no. 11, pp. 1537–1554, Nov. 2010.

[205] D. K. Pratihar, *Soft Computing*. Alpha Science International, Ltd, 2007.

[206] G. Rudolph, "Canonical Genetic Algorithms," *IEEE Trans. Neural Networks*, vol. 5, no. 1, pp. 96–101, 1994.

[207] Q. Zhang, A. Zhou, S. Zhao, P. N. Suganthan, and W. Liu, "Multiobjective optimization Test Instances for the CEC 2009 Special Session and Competition," *2009 IEEE Congr. Evol. Comput. (CEC 2009)*, no. January 2015, pp. 1–30, 2009.

[208] W. Khan Mashwani, A. Salhi, O. Yeniay, H. Hussian, and M. A. Jan, "Hybrid non-dominated sorting genetic algorithm with adaptive operators selection," *Appl. Soft Comput.*, vol. 56, pp. 1–18, Jul. 2017.

[209] A. Auger, J. Bader, D. Brockhoff, and E. Zitzler, "Hypervolume-based multiobjective optimization: Theoretical foundations and practical implications," *Theor. Comput. Sci.*, vol. 425, pp. 75–103, Mar. 2012.

[210] J. D. Schaffer, "Some experiments in machine learning using vector evaluated genetic algorithms," *Other Information Thesis Ph D.* p. 166, 1984.

[211] K. Deb, L. Thiele, M. Laumanns, and E. Zitzler, "Scalable multi-objective optimization test problems," in *Proceedings of the 2002 Congress on Evolutionary Computation. CEC'02 (Cat. No.02TH8600)*, 2002, vol. 1, no. i, pp. 825–830.

[212] E. Zitzler, K. Deb, and L. Thiele, "Comparison of multiobjective evolutionary algorithms: empirical results.," *Evol. Comput.*, vol. 8, no. 2, pp. 173–195, 2000.

[213] P. K. Jamwal, B. Abdikenov, and S. Hussain, "Evolutionary optimization using equitable fuzzy sorting genetic algorithm (EFSGA)," *IEEE Access*, vol. 7, 2019.

[214] S. Huband, P. Hingston, L. Barone, and L. While, "A review of multiobjective test problems and a scalable test problem toolkit," *IEEE Trans. Evol. Comput.*, vol. 10, no. 5, pp. 477–506, Oct. 2006.

[215] Hui Li and Qingfu 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.

[216] J. Derrac, S. García, D. Molina, and F. Herrera, “A practical tutorial on the use of nonparametric statistical tests as a methodology for comparing evolutionary and swarm intelligence algorithms,” *Swarm Evol. Comput.*, vol. 1, no. 1, pp. 3–18, Mar. 2011.

[217] J. Xia *et al.*, “Adjusted weight voting algorithm for random forests in handling missing values,” *Pattern Recognit.*, vol. 69, pp. 52–60, Sep. 2017.

[218] Z. Zhang, “Missing data imputation: Focusing on single imputation,” *Ann. Transl. Med.*, vol. 4, no. 1, 2016.