A Modified Feed Forward Neural Network with Particle Swarm Optimization

A thesis

Submitted to the Council of the

College of Science at the University of

Sulaimani in partial fulfillment of the requirements

for the Degree of Master of

Science in Computer

By

Asia Latef Jabbar

B.Sc. Computer Science (2010), University of Kirkuk

Supervised by

Dr. Tarik A. Rashid

Professor

November 2016 **Sermawaz 2716**

بسم الله الرحمن الرحيم

وما أوتيتم من العلم إلا قليلا

صدق الله العظيم

الاسراء 85

Supervisor Certification

I certify that the preparation of thesis titled "A **Modified Feed Forward Neural Network with Particle Swarm Optimization**" accomplished by (**Asia Latef Jabbar**) was prepared under my supervision in the college of Science, at the University of Sulaimani, as partial fulfillment of the requirements for the degree of Master of Science in (Computer).

Signature:

Supervisor: Dr. Tarik A. Rashid

Scientific Title: Professor

Date: 15/ 7/ 2016

In view of the available recommendation, I forward this thesis for debate by the examining committee.

Signature: A . A . M

Name: Dr. Aree Ali Mohammed

Scientific Title: Assistant Professor

Head of the Department

Date: 20/ 7/ 2016

Linguistic Evaluation Certification

I hereby certify that this thesis titled "A **Modified Feed Forward Neural Network with Particle Swarm Optimization** " prepared by (**Asia Latef Jabbar**), has been read and checked and after indicating all the grammatical and spelling mistakes; the thesis was given again to the candidate to make the adequate corrections. After the second reading, I found that the candidate corrected the indicated mistakes. Therefore, I certify that this thesis is free from mistakes.

Signature: Sarah

Name: Dr. Sarah K. Othman

Position: English Department, School of Languages, University of Sulaimani.

Date: 25/09/ 2016

Examining Committee Certification

We certify that we have read this thesis entitled "A **Modified Feed Forward Neural Network with Particle Swarm Optimization** " prepared by (**Asia Latef Jabbar**), and as Examining Committee, examined the student in its content and in what is connected with it, and in our opinion it meets the basic requirements toward the degree of Master of Science in Computer.

Signature:
Name: Dr. Nzar A. Ali
Title: Assistant Professor
Date: 3 / 12/ 2016
(Chairman)

Signature:
Name: Dr. Nooruldeen Nasih Qader
Title: Lecturer
Date: 3 / 12/ 2016
(Member)

Signature:
Name: Dr. Alaa K. Jumaa
Title: Lecturer
Date: 3 / 12/ 2016
(Member)

Signature:
Name: Dr. Tarik A. Rashid
Title: Professor
Date: 2 / 12/ 2016
(Member- Supervisor)

Approved by the Dean of the College of Science.

Signature:
Name: Dr. Bakhtiar Qader Aziz
Title: Professor
Date: / / 2016

Dedication

This thesis is dedicated to:

My father and mother with my love

My sisters and brothers

Acknowledgement

First of all, my great thanks to **Allah** for giving me the ability and faith to fulfil this work.

Foremost, I would like to express my sincere gratitude to my supervisor **Prof. Dr. Tarik A. Rashid** for his valuable guidance, encouragement, and expert advice for showing me the right way of carrying out this thesis.

Special thanks to my parents, for their help and endless moral support during my life, who were with me in every step throughout my entire studies.

I would also like to thank Mr. Dilshad M. Shokur at the information technology center at the University of Sulaimani and Mr. Forat F. Hasan for their generous support and help.

Finally, I would like to thank the lecturers of the Computer Department / Faculty of Science and Science Education, School of Science for their support, opportunities and facilities for carrying out this work.

A. L. Jabbar

Abstract

In today's economic transformation setting round the globe, many challenges are occurred. One of the challenges was with the organization strategy in the labor market in the field of the human resource management. Where it is one of the important parts in an organization.

This field contained several challenges such as manual handling of employees information, identifying an existing talent among all the employees in the organization, endorsing an employee for promotion that he/she deserves it, accepting new employee, preventing an excellent employee from leaving the organization. All these problems will lead to human error and time consuming. Therefore, there has been growing interest in human resource management of corporations and its consequence on revenues of these corporations.

In this thesis, surveying approach was used for collecting data from different companies in Kurdistan Region. The collected data was prepared by going through steps of preprocessing techniques for handling missing values in the dataset, and then balancing class samples in the dataset.

Then, in the classification phase Forward Neural Network, Fuzzy Rough Nearest Neighbor, Naïve Bayes, and Decision Tree were used for evaluating accuracy. In addition, a new modified model called FNNPSOED is developed via using Particle Swarm Optimization. The Particle Swarm Optimization is used for optimizing the weights and biases of Forward Neural Network with using Euclidean Distance method for improving Particle Swarm Optimization. The FNNPSOED produced the best results with using four types of test dataset (350, 400, 500, 600 instances), and the results were 100.00%, 99.500%, 99.00%, 98.833% respectively.

List of Contents

Chapter Three: Proposed System Methodology

Chapter Four: Results and Discussion

Chapter Five: Conclusions and Future Recommendation

List of Tables

List of Figures

List of Abbreviations

Abbreviations	Meaning
AC	Acceleration Coefficient
AI	Artificial Intelligence
ANN	Artificial Neural Network
CCI	Correctly Classified Instances
DT	Decision Tree
ED	Euclidean Distance
FN	False Negative
FNN	Forward Neural Network
FNNPSO	Forward Neural Network with Particle Swarm Optimization
FNNPSOED	Forward Neural Network with Particle Swarm Optimization using Euclidean Distance
FNR	False Negative Rate
FP	False Positive
FPR	False Positive Rate
FRNN	Fuzzy Rough Nearest Neighbor
FST	Fuzzy Set Theory
HR	Human Resource
HRIS	Human Resource Information System
HRM	Human Resource Management
ICI	Incorrectly Classified Instances
IW	Inertia Weight

KDD	Knowledge Discovery in Database
KNN	K Nearest Neighbor
MAE	Mean Absolute Error
MIS	Management Information System
MLP	Multilayer Perceptron
MSE	Mean Square Error
NB	Naïve Bayes
NN	Neural Network
NNSOA	Neural Network Simultaneous Optimization Algorithm
NoP	Number of Particles
OB	Organizational Behavior
PE	Processing Element
PSO	Particle Swarm Optimization
RMSE	Root Mean Square Error
RST	Rough Set Theory
SAS	Statistical Analysis System
SEMMA	Sample Explore Modify Model Assess
SMOTE	Synthetic Minority Over Sampling Technique
TN	True Negative
TNR	True Negative Rate
TP	True Positive
TPR	True Positive Rate

Chapter One

Introduction

Chapter One

Introduction

1.1 Overview

Conventionally, finance and economics concentrated on the labor market rather than looking inside the "black box" of firms. In the current global economy, companies must develop human capital continuously. Commercial sociologists and psychologists made the running in Human Resource Management (HRM). HRM is considered as a major field in today's economic transformation [1, 2].

In the era of new knowledge, economy environment competitive, a superior employee with more activities in an enterprise gains more weight and value than before that becomes the solution to a success of companies in today's economic transformation and labor market. Employees in companies are remarkable because of their importance for development in companies as their attitudes and behaviors play an important role in the quality of work. In this case, it is the responsibility of employees for providing a preferable competitive for the organizations. Therefore, the success of the companies relies on managing and retaining employees [3].

Study of Organizational Behavior (OB) is very interesting and challenging too. It is used to refer to individuals and group of people working together in teams. It relates to the expected behavior of an individual in the organization. In a particular department there must not have two individuals with the same behavior even if it is necessary and needed from the organization itself. Clearly, there are no absolutes in human behavior. Each individual is the major factor that is contributor to the productivity and increasing the revenues in an organization.

Therefore, the study of human behavior is imperative. Researchers and managements must understand credentials of an individual, his/her background, educational update groups and other situational factors on the behavior of the employees. Accordingly, it is the responsibility of the managers to expect, explain, predict, evaluate and modify human behavior that will fundamentally be governed by knowledge, skill and experience of the manager in dealing with a large group of people in various cases [4].

Every organization must have a procedure to respond to its needs for talented person, for competition in the marketplace. This procedure is referred to as Talent Management. For instance, Talent Management must be a fully integrated system where all parts of this system are interrelated with each other for succession in managing the existing talent, where human resource management must hand-pick the individuals that they considered to be as an existing 'talent', and place them in a strategic places or positions, in which he\she can make its innovation in the domain. Sometimes wrong human resources can be in the wrong positions as a result from wrong managing these talents, which may lead to the time consuming, and decreasing in the quality of the productions. Extracting the exact talents by the manager is still questionable despite of handling the existing talents from the human resources depending on their experts. One of the short comings of this method is that many talents have never been evaluated, and they are rather lost without benefiting them. This is considered a loss for the company [5].

Nowadays, productivity improvement, good customer service, greater profitability and the whole organizational survival can be determined by effective HRM, Talent Management, Talent Strategy, and Succession Management. Management must not only face contemporary issues of human resource so as to connect the HRM and productivity quality successively, but it also has to deal

effectively with future challenges that HRM might encounter. HRM` that deals with human capital where it is a critical case aims at facilitating organizational competitiveness; enhancing productivity and quality; promoting individual growth and development, and complying with legal and social obligation [6].

Recently, an organization has to concentrate effectively in terms of cost, quality, service or innovation. All these depend on having enough right people with the right skills, employed in the right position. Talent management involves a lot of managerial decisions and these types of decisions may be ambiguous and difficult. The process of identifying the existing talents in an organization is among the top talent management challenges and it is an important issue. To ensure that the right person is in the right job, various factors such as: human experience, knowledge, preference and judgment must be considered [7].

HRM applications that are embedded with Artificial Intelligence (AI) techniques can solve unstructured and indistinct decision making problems. These applications can help decision makers 'managers' to solve inconsistent, inaccurate and unpredicted decision problems. In the advancement of AI technology, there are many techniques that can be used to improve the capabilities of HRM application. Data Mining is one of AI technologies that has been developed for exploring, analyzing and creating meaningful patterns and rules in large quantities of data that can provide a good resource for knowledge discovery for solving challenges in HRM departments [8]. Classification, Clustering, Association rule mining are approaches that can be useful in this case for solving previous mentioned problems.

For solving their problems there are many areas which adopted this approach such as in finance, medicine, marketing, telecommunication, manufacturing, customer relationship and so on. Over the years, data mining has evolved various

techniques to perform tasks including database oriented techniques, statistics, machine learning, pattern recognition [6, 7]. Classification and prediction techniques are among the popular tasks in data mining for solving human resource problems that are used in this thesis, for predicting better solutions in any organization.

1.2 Problem Statement

Since manual handling of employee information has raised a number of challenges occurred. Among the challenges in the field of human resource are managing an organization talents which involves a lot of managerial decisions. These decisions are very uncertain and difficult. Besides, the process of identifying an existing talent in an organization is among the top talent management issues and challenges.

This is evident in procedures such as recommendation for promotion, accepting new employees if needed, and leaving management where an employee is required to fill in a form which may take several weeks or months to be approved.

The use of paper work in handling some of these processes could lead to human error; papers may end up in the wrong hands and not forgetting the fact that this is time consuming. Thus, the current structure in organizing firms in Kurdistan is non-systematic and manually performed, as a result, in some cases performance of employees' cause a low level of acceptance among the staff.

This thesis focuses on finding solutions to the mentioned problems and reduce the time spent to speed up the process using data mining techniques for building intelligent expert systems.

1.3 Literature Survey

To avoid employee leaving and a huge drop in company's revenue, a company must try to decrease the employees' turnover and make a preferable decision among employees, which makes him/her faithful to their work. As a result, to support companies in building an intelligent system for predicting their employees leaving, researchers and companies tried to build systems and worked in the area for solving employees leaving problems. Here is a survey on some of the related research works.

Sextona, McMurtrey, Michalopoulos and Smith in 2004, attempted to explain why employees leave and how to prevent the drain of employee talent. They focused on using artificial neural networks to predict turnover. If turnover is found to be predictable, the identification of at-risk employees will allow us to focus on their specific needs or concerns in order to retain them in the workforce. Also, by using a Modified Genetic Algorithm to train the artificial neural networks, also relevant predictors or inputs can be identified, which can provide information about how the work environment as a whole can be enhanced. In this work, Neural Network Simultaneous Optimization Algorithm (NNSOA) is performed exceedingly well for optimizing an artificial neural network while simultaneously eliminating unnecessary weights in the artificial neural networks structure during the training process for the employee turnover problem. According to this, benefit in identifying unneeded weights in the solution is the identification of irrelevant variables in the artificial neural networks model. This research founded that NNSOA trained in a 10-fold cross validation experimental design can predict the turnover rate with a high degree of accuracy for a small mid-west manufacturing company [9].

Hsin-Yun Chang in 2009 proposed a new method that could select subsets more efficiently. In addition, the reasons of employers voluntarily turnover were also investigated in order to increase the classification accuracy and to help managers to prevent employers' turnover. The mixed feature subset selection used in this study combined Taguchi method and Nearest Neighbor Classification Rules, used to select feature subset and analyze the factors to find the best predictor of employer turnover. The results showed that through the mixed feature subset selection method, total 18 factors were found important to the employers. In addition, the accuracy of the correct selection was 87.85% which was higher than before using this feature subset selection method (80.93%) [10].

In 2010, Jantan, Hamdan and Othman [11], done research about the decision tree C4.5 classification algorithm to generate the classification rules for human talent performance records for predicting the potential human talent. In this study, recommendation for promotion (yes/no) is considered as the target class in the classification process. For human talent dataset, employees' data from one of Malaysian higher learning institutions was used as a training dataset. In the first phase of mining process, the training dataset is prepared using the data mining preprocessing task. In the second phase, the C4.5 classifier is used to generate talent performance knowledge from yearly performance evaluation database. In that case, the hidden and valuable knowledge is discovered in the related databases that will be summarized in the decision tree structure. In addition, the accuracy of correctly classified instances was 95.0847 %.

In 2011, Jantan, Hamdan, and Othman [12], in another research work, they proposed the potential Data Mining Techniques for talent forecasting and identifying potential talent by predicting their performance using past experience knowledge. They attempted to use classifier algorithm C4.5 and Random Forest

for decision tree; and Multilayer Perceptron (MLP) and Radial Basic Function Network for neural network. In the initial stage of this study, they run the selected classifier algorithms for the sample of employee data. In this case, they focused on the accuracy of the techniques to find the suitable classifier for HRM data. The employee data contains 53 related attributes from the five evaluation performance factors, they were namely; Background, Previous Performance, Knowledge and Expertise, Management skill, Personal characteristics. The accuracy for each of the classifier algorithm was as follows: C4.5/J48 95.14%, Random forest 74.91%, Multilayer Perceptron 87.16%, and Radial basis function network 91.45%.

In 2012, Al-Radaideh and Al Nagi [13], used decision tree with two versions, ID3 and C4.5 and Naïve Bayes classifier in three experiments to predict the performance of employees. For each experiment, the accuracy was evaluated using 10-folds cross-validation, and hold-out method. The accuracy for each classifier in each experiment was as follows: in the first experiment the accuracy with 10-Fold Cross Validation and Hold-Out (% 60) for ID3 was 36.9% and 36.5% respectively, for C4.5 (J4.8) was 42.3% , and 48.1% respectively, and for Naïve Bayes was 40.7% and 44.2% respectively. While in the second experiment decision tree used different starting node. The accuracy for each classifier in the second experiment was as follows: with 10-Fold Cross Validation and Hold-Out (%60), for ID3 was 37.8% and 26.7% respectively, C4.5 (J4.8) was 48.6%, 53.3%, and Naïve Bayes was 37.8%, 46.7% respectively. Finally, in the third experiment the accuracy percentages resulted from applying the algorithms of ID3, C4.5 and Naïve Bayes on the dataset the accuracy percentages with 10-Fold Cross Validation and Hold-Out (%60) were as follow: 50%, 43.7% respectively for ID3, 60.5%, 56.2% respectively for C4.5 (J4.8), 65.8%, and 68.7% respectively for Naïve Bayes.

In 2013, Florence and Savithri [14], proposed a system by using C4.5 classifier algorithm to identify the skill set in order to evaluate the performance of the individual. This technique will be used to construct classification rule set obtained to the Human Resource dataset to predict the potential talent which helps in determining whether the individual is fit for the appraisal or not. In this research, 200 data records are used, 14 attributes are used as input values and the Promotion Recommendation attribute serving as the target class in the human resource dataset. In this work, 98% of the data set is used in the training phase. The target class attribute is discrete in nature with Yes/No as the values.

Another work in 2013 is conducted by Bangsuk Jantawan and Cheng-Fa Tsai [15]. They proposed a system to predict whether a graduate has been employed, remains unemployed, or is in an undetermined situation after graduation, as graduates remains increase the number of graduates produced by higher education institutions each year. Graduates are facing more competition to ensure their employment in the job market. This study attempts to identify the attributes that influence graduate employment based on actual data obtained from the graduates themselves 12 months after graduation. They performed this prediction based on a series of classification experiments using various algorithms under Bayesian and decision methods to classify a graduate profile as employed, unemployed, or other. Results show that the Bayse algorithm, achieved the highest accuracy of 99.77%. The average accuracy of other Tree algorithms was 98.31%.

In 2014, Tamizharasi, UmaRani [16]. used neural networks, decision trees and logistic regression as solutions of turnover problem in human resource management. The research confirmed that the costs associated with losing a good employee and training a new one can be equal to 1.5 times the salary of the exiting employee. This study work utilized data extracted from current employees by

questionnaires and data of exiting employees of the Company, which included the individual reasons given for leaving the organization. In this research, steps of Sample, Explore, Modify, Model, and Assess (SEMMA) methodology are followed. The model, developed in Statistical Analysis System (SAS) Enterprise Miner, based on SEMMA methodology.

In this research work, it is realized in the literature that little or no research works were carried out to tackle the problem of predicting employee behavior using natural inspired techniques like Particle Swarm Optimization for optimizing artificial neural networks weights and biases, and since classification of this application system has not been used in Kurdistan region, thus, University of Sulaimani found it very significant to regulate an appropriate system to resolve the problems of predicting employee behavior. In this system, supervised classification methods with suitable preprocessing and optimization techniques are applied for the purpose of providing the best results. Euclidean Distance (ED) method was used for improving Particle Swarm Optimization.

1.4 The Aim of the Thesis

This thesis aims to build a system for predicting employees' behavior in one organization and identify critical jobs according to the features and attributes that companies and organizations depend on. It is worthy to remember that sometimes people who are in the critical jobs are not the best performers, and vice versa the best performers are not in the critical jobs. There should be a clear process for identifying and developing high potentials. The process should be able to identify the top performers and ensure their development through which they get higher chances of better performances and retention within the organization.

This thesis provides a survey in the field of human resource management in the companies and organizations in Kurdistan. So, classification techniques are used to classify the employee's performance. In this case, the class level for the performance is whether the employee gets recommendation for promotion or not. For this purpose, employee's information is collected and used from the selected organizations as our dataset. Therefore, the purpose of this thesis is to suggest the best decision for an employee future performance through some experiments using the selected classification algorithms. In the proposed approach, Artificial Neural Network classification technique is used for the classification purpose, and the Particle Swarm Optimization is used as an optimization technique for the weights and biases of artificial neural networks.

1.5 Thesis Outlines

The rest of the thesis is organized as follows:

- **Chapter Two (Human Resource Management and Data mining):**

This chapter presents a description about Human Resource Information System (HRIS), components of HRIS, features of HRIS with its advantages, and describing HRIS phase models. So, the relation of data mining with HRIS is discussed in this chapter by explaining its steps including preprocessing, used classification technique, used technique for optimization purpose, and used method for improving the optimization technique PSO.

- **Chapter Three (Proposed System Methodology):**

This chapter is about the proposed intelligent system for improving the employee behavior in any organization. System structure, train and test data sets, preprocessing, proposed system for HRM are presented in this chapter.

- **Chapter Four (Results and Discussions):**

In this chapter, results of the applied classification techniques, including the proposed system in the experimental cases with its discussions are presented, each experiment case with its training and testing datasets.

- **Chapter Five (Conclusions and Future Recommendation):**

Is a presentation of the summarization of some concluded points from the results of the implemented system with discussion, besides of a number of the future work recommendation to be the key points of guiding for improving the proposed system.

Chapter Two

Human Resource Management and Data Mining

Chapter Two
Human Resource Management and Data Mining

2.1 Overview

It has been said that employees are the most credible component in any business. In fact, the key elements of competitive advantage are people and the management of people in any organization. Unlike conventional vision for competitive advantage that confirms other points as barriers that interfere to entry to economic scale, access to capital, and regulated competition, but more recent views have highlighted an organization's strategic management of its human resources as a source of competitive advantage, which cannot be easily obtained or imitated [17].

HRM strategy scholars discussed that a success of an organization depends on its employees and their behavior in carrying out the strategies of the business. Employees and employee management skillfulness must be considered in the same way that companies are depending on for improving their performance [18].

In the knowledge economic period, human resource management is raised to a higher level, and many techniques become an important part of HRM. Even so, some problems also appeared. Thus, an advance technology would be found inevitably. Data mining is good at finding a model from data which has been applied in many fields and obtained good economic results [19].

Generally speaking, the research of data mining has an earlier start, it was first discovered in the late 1980s. Considerable developments were occurred in the 1990s continued through the 21st century. Data mining is the process that have been developed to discover and analyze the interesting, unexpected and valuable

constructions from the huge amounts of data to define significant patterns and rules [20].

In an organization the role of Human Resource (HR) professionals is improved besides of business related technological achievements. HR professionals are now able to customize more time to strategic decisions in an organization. While handling data processing manually is no longer needed by HR professionals, they should not abandon their relation to data collected and about the organization's employees. Decision-making processes can be done and supported as human resources data is available within organizations. The challenge is in identifying useful information in vast human resources databases that are the important factor of the automation of HR-related transaction processing [21].

In huge data bases, data mining can be considered as an evolving approach to data analysis that become a useful tool to HR professionals. Data mining involves extracting knowledge based on patterns of data in the huge databases. Organizations that employ thousands of employees and track a multitude of employment-related information might find valuable information patterns contained within their databases to provide insights in such areas as employee retention and compensation planning [21].

In this chapter, components of Human Resource Information Systems and gain an understanding of the steps in applying data-mining techniques to HR Information Systems will be discussed.

2.2 Human Resource Information System

The concept of human resource information system (HRIS) has been derived from the concept of management information system (MIS). MIS is defined as systematic collection, maintenance, and retrieving data for supporting the management, analysis, and decision-making functions in an organization [22].

Organizations and companies are generally implemented information systems and information technologies to improve the quality and providing human resource management system services in today's global competitive business environment. With the evolution of information systems and technology, meeting information requirements has been greatly enhanced through the creation of HRIS [23].

2.3 Components of Human Resource Information System

HRIS is a key management tool on people and jobs. All the relevant data are integrates in the system, which otherwise might have been lying in a fragmented and scattered way at various points; converts this data into meaningful information makes it accessible to the persons, who need it for their decisions. HRIS major functional components are represented in Figure (2.1) [25].

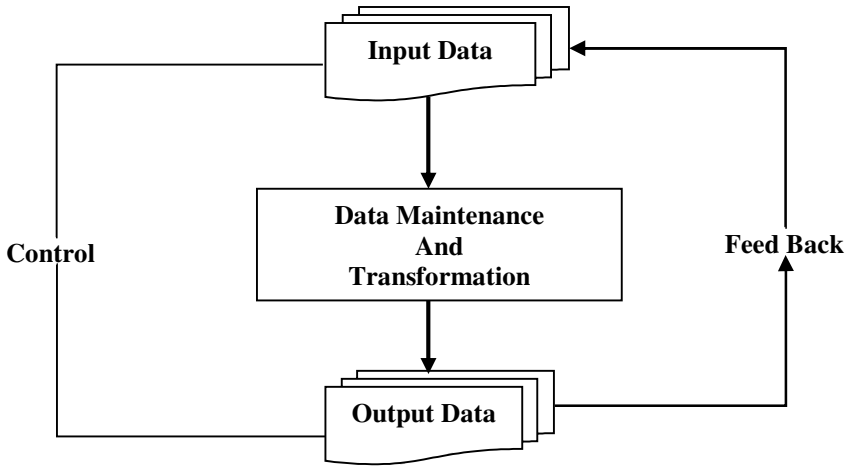


Figure (2.1): Components of Human Resource Information System [22]

1. **Input:** - Input function provides the capabilities of getting human resource data into the HRIS. Personnel information such as education, age, experience, present salary, whether promoted or not, and other necessary detailed information relating to the human resources in the organization are used as an input data into the HRIS [23].

2. **Data Maintenance and Transformation: -** Gained information fed to the computer can be transformed into more meaningful and necessary information that is exactly required by the organization. This is the conversion stage of computerized HRIS [23].

3. **Output: -** Output refers to the printouts of the transformed material from the computer printer like salary statement, report on the performance of an employee, budget estimates, etc. This function of HRIS is the most visible one because the majority of HRIS uses are not involved with collecting, editing, and updating human resource data; rather they are concerned with information and reports to be used by the systems [23].

4. **Feedback and Control: -** Whether the output obtained is relevant and useful or not, it must be known. The method of ensuring it is known as feedback. Feedback establishes control over the system.

2.4 Features of HRIS

An HRIS should be designed to provide information that is [22]:

1- **Timely:** A manager must have access to up-to-date information.
2- **Accurate:** A manger must be able to rely on the accuracy of the information provided.
3- **Concise:** A manger can absorb only so much information at any one time.

4- **Relevant:** A manger should receive only the information needed in a particular situation.

5- **Complete:** A manager should receive complete, not partial information.

2.5 The Advantages of Human Resource Information System

The advantages of HRIS can be outlined as follows [24]:-

1. Reduction in the cost of stored data in human resource data base.
2. Higher speed of retrieval and processing of data and availability of accurate and timely data about human resources.
3. Better analysis leading to more effective decision making and more meaningful career planning and counseling at all levels.
4. Improved quality of reports and more transparency in the system.
5. Better ability to respond to environmental changes.

2.6 HRIS Engineering Phase Model

An organization that needs to set up HRIS; it has to go through a set of levels, which are shown in the Figure (2.2). These levels are [25]:

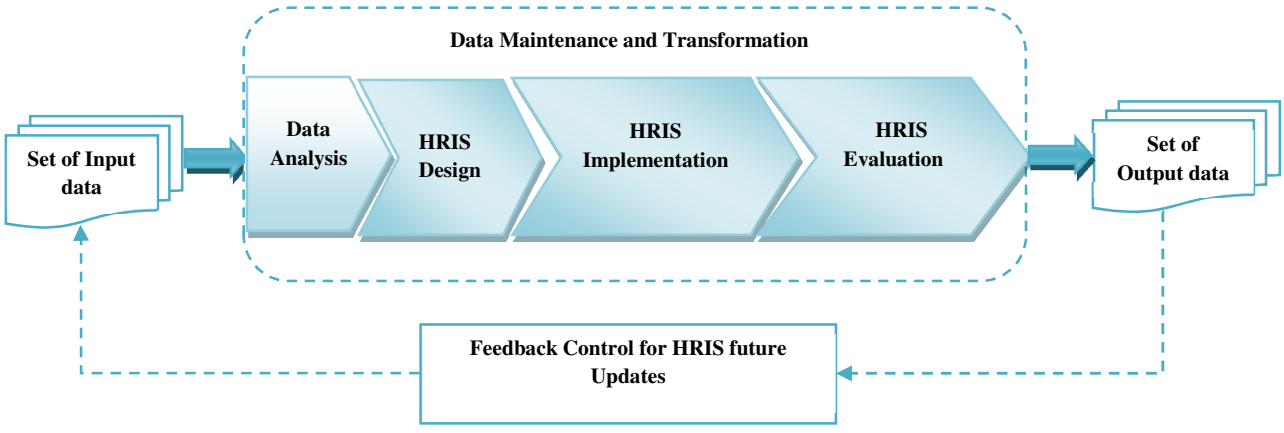


Figure (2.2): HRIS Engineering Phase Model [25].

1. Data Analysis: HRIS designing requires information gathering, using suitable methods with getting suitable research tools, data collection is related to this phase.

2. Design: Collected data must be relived (i.e. reducing missing values, balancing data, filtering of useful data with rejecting irrelevant) as well as relived data summarized into suitable model like decision tables, decision-tree, mathematical models…etc., with keeping in mind and decide what method is suitable for each collected refined portion of data.

3. Implementation: This Phase-3, is a complement step to Phase-2 to change HRIS models with selected models into actual HRIS-GUI for HR process & planning using suitable coding language which best matches to models with taking the organizational needs into account.

4. Evaluation: HRIS-testing must be verified for the developed HRIS. It can simply be defined as "it is the process of checking whether an HRIS user's requirement engineered in HRIS in the form of process and function". If testing is OK and acceptable, then it is implemented, otherwise through HRIS feedback control, facts (data) about shortcomings and errors in the developed HRIS are gathered for updating, modification based on HRIS users need.

2.7 Data Mining and Human Resource Management

Enormous amount of data are stored in files, data bases, and other warehouses. These stored data are important to develop and improve powerful means for analysis, interpretation for knowledge discovery that can lead to decision making [26].

Knowledge Discovery in Databases (KDD) is an interactive discovery process, exploratory analysis and modeling of large data repositories. KDD is the organized process of identifying valid, useful, and understandable patterns from huge and complex data sets [27]. The process generalizes to non-database sources of data, although it emphasizes databases as a primary source of data. Data Mining is one approach of KDD consists many steps, each attempting to complete a particular discovery task and each accomplished by the application of a discovery method [28].

Data mining is a powerful new technology with great potential in information system. The goal of this process is to mine patterns, associations, changes, anomalies, and statistically significant structures from large amount of data [29].

Simply stated, data mining refers to extracting or “mining” knowledge from large amounts of data [30]. This is defined as a sophisticated data search capability that uses statistical algorithms to discover patterns and correlations in data [31].

The major reason that data mining has attracted a great deal of attention in the information industry and in society, is due to the wide availability of huge amounts of data and the imminent need for turning such data into useful information and knowledge. The information and knowledge gained can be used for applications ranging from market analysis, fraud detection, and customer retention, to production control and science exploration [30].

KDD has several steps that should be followed for gaining the useful knowledge, these steps are represented in Figure (2.3).

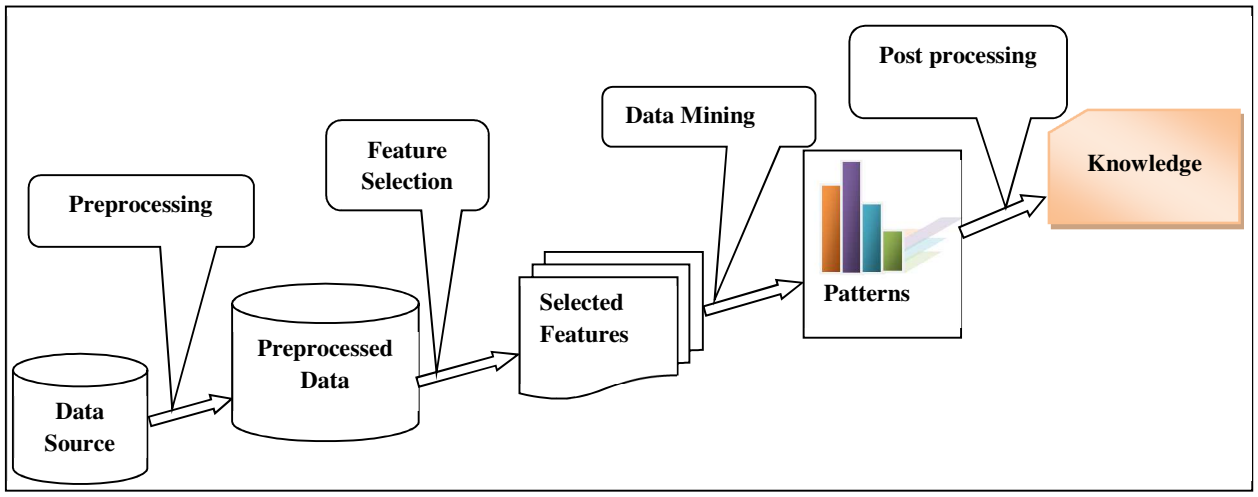


Figure (2.3): Knowledge discovery process [29]

Each step of knowledge discovery that are used in this thesis will be explained in the following sections.

2.8 Data Preprocessing

Today's real-world databases are highly susceptible to missing values, and inconsistent data due to their typically huge size (often several gigabytes or more) and their likely origin from multiple, heterogeneous sources. Low-quality data will lead to low-quality mining results [30].

Data pre-processing is a critical task in the knowledge discovery process for ensuring good data quality [32].

2.8.1 Handling Missing Value

In any dataset, missing value is the value of an attribute or feature that was not obtained during data collection, which intended to obtain. Missing values can appear because respondent did not answer all questions in questionnaire, during manual data entry process, incorrect measurement, faulty experiment, and many others [33]. Handling these missing values in the data set is an important situation or problem with the aim to recover the missing values as close as possible to the original values [32]. Replacing this missing value can be done by using the mean of an attribute, where the mean is calculated based on all known values of the attribute. This method is usable only for numeric attributes and is usually combined with replacing missing values with most common attribute values for symbolic attributes [33]. The pseudo code of the algorithm is explained in Appendix B1.

2.8.2 Class Balancing

Data sets possibly have an imbalanced class distribution, where one class is represented by a large number of samples while the others are represented by small numbers. On such data learning classification methods generally perform poorly because the classifier often learns better the majority class. The reason for this is that learning classifiers attempt to reduce global quantities such as the error rate, and do not take the data distribution into consideration. As a result, samples from the dominant class are well-classified whereas samples from the minority class tend to be misclassified [34].

Synthetic Minority Over-sampling Technique (SMOTE) is an over-sampling technique whereby synthetic minority examples are generated. It combines informed over-sampling of the minority class with random under-sampling of the

majority class. Using the over-sampling approach the minority class is over-sampled by creating artificial examples of k nearest class neighbors. This technique creates artificial samples to increase the size of minority class. It balances the data by increasing the number of minority instances by over-sampling them [34]. Appendix B2 can explain steps of this algorithm.

2.9 Classification

Data mining consists of a set of techniques that can be used to extract relevant and interesting knowledge from data. These techniques have several tasks such as association rule mining, classification and prediction, and clustering [13].

Classification and prediction are two forms of data analysis that can be used to extract models describing important data classes or to predict future data trends [30]. It is a supervised learning method which requires labeled training data to generate rules for classifying test data into predetermined classes. It is a two-phase process, the first phase is the learning phase, where the training data is analyzed and classification rules are generated. The next phase is the classification, where test data is classified into classes according to the generated rules. Since classification algorithms require that classes can be defined based on data attribute values. So it consists of predicting the value of a (categorical) attribute (the class) based on the values of other attributes (the predicting attributes) [35].

Let a_{ij} denote the value of specific characteristics or attributes of a population of elements in a data set, where each element i ($i=1,\ldots,m$) is described by attribute j ($j=1,\ldots,n$).

A decision rule identifies to classify these elements in a manner to correctly determine whether a given vector A_i= (a_{i1},…,a_{in}) should belong among the elements of Group 1 or instead among those of Group 2 [36].

2.9.1 Artificial Neural Network

An Artificial Neural Network (ANN), often just called a "neural network" (NN), is a mathematical or computational model based on biological neural networks [37]. Neural networks represent a brain metaphor for information processing. These models are biologically inspired rather than an exact replica of how the brain actually functions. It has been postulated that a model or a system that is enlightened and supported by the results from brain research, with a structure similar to that of biological neural networks, could exhibit similar intelligent functionality.

The human brain is composed of special biological cells called neurons that process information from one neuron to another neuron with the help of some electrical and chemical change [38]. It is composed of a cell body or soma and two types of out reaching tree like branches: the axon and the dendrites. The cell body has a nucleus that contains information about hereditary traits and plasma that holds the molecular equipment or producing material needed by the neurons. The whole process of receiving and sending signals is done in particular manner like a neuron receives signals from other neuron through dendrites [39]. Figure (2.4) illustrated the structure of biological neural network cells.

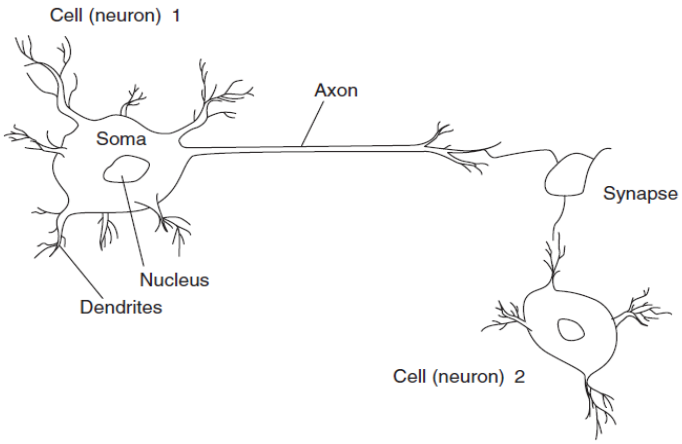


Figure (2.4): Two interconnected biological cells [38]

An ANN model was introduced depending on a biological neural network. It uses a very restricted set of concepts from biological neural network systems for computing operations. Neural concepts are usually implemented as software simulations of the massively parallel processes that involve processing elements (also called artificial neurons, or neurons) interconnected in network architecture. The artificial neuron receives input data from other neurons as biological neuron receives electrochemical impulses from other neurons. The output of the artificial neuron corresponds to signals sent out from a biological neuron over its axon. These artificial signals can be changed by weights in a manner similar to the physical changes that occur in the synapses, Figure (2.5) [38].

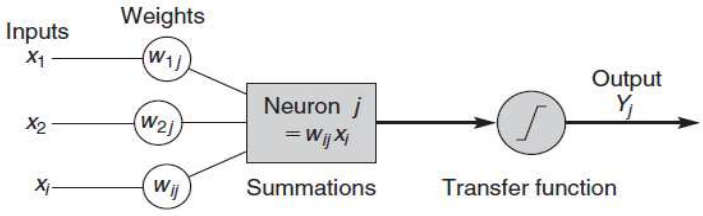


Figure (2.5): Processing Information in an Artificial Neuron [38]

2.9.2 Elements of ANN

As it was discussed, neural network is composed of a collection of basic processing units which are called neurons; these neurons are grouped in layers and organized in different ways to form the network structure. There are many topologies to organize neurons. One of the popular topology is known as feed forward-back propagation paradigm (or simply back propagation), where all neurons in one layer are linked to the neurons in the next layer but, it does not allow any feedback linkage. Each one of the Processing Elements (PE) or neurons receives inputs, processes them, and delivers a single output, as shown in Figure (2.5). Once the structure of a neural network is determined, information can be processed as shown in Figure (2.6), it contains three layers: *input, intermediate (hidden layer), and output* [38].

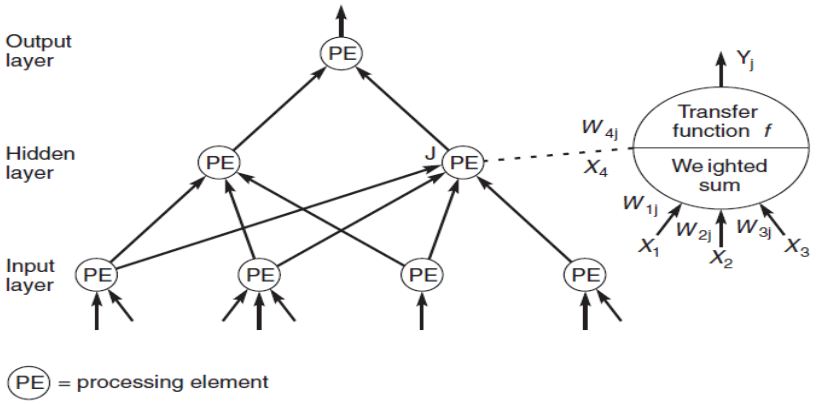


Figure (2.6): Neural Network Structure with One Hidden Layer [38]

The major elements to construct this structure are explained as follows:-

1) **Inputs:** Each input corresponds to a single attribute, where the input can be raw input data or the output of other processing elements. For example, if the problem is to decide on a recommendation of promotion for an employee, some attributes could be the applicant's income level, age, and

education, etc. The numeric value, or representation, of an attribute is the input to the network. Several types of data, such as text, pictures, and voice can be used as inputs [38].

2) **Hidden Layer:** A hidden layer is a layer of neurons that takes input from the previous layer and converts those inputs into outputs for further processing. Complex practical applications require one or more hidden layers between the input and output neurons and a correspondingly large number of weights, although it is quite common to use only one hidden layer. The role of each hidden neuron is affected by the input neurons and the weights on the connections between the input and the hidden neurons [38].

3) **Outputs**: The outputs of a network can be the solution to a problem (final result) or it can be inputs to other neurons. For example, in the case of a recommendation of promotion, the outputs can be yes or no. The ANN assigns numeric values to the outputs, such as 0 for yes and 1 for no. Often, post processing of the outputs is required because some networks use two outputs: one for yes and another for no. It is common to round the outputs to the nearest 0 or 1. The behavior of the output neurons relies on the activation of the hidden neurons and the weights between the hidden and the output neurons [40].

4) **Connection Weights:** The key elements in an ANN are Connection Weights. They express the relative strength (or mathematical value) of the input data or the many connections that transfer data from layer to layer. In other words, weights express the relative importance of each input to a processing element and, ultimately, the outputs. Weights are crucial where

they store learned patterns of information. It is through repeated adjustments of weights that a network learns [40].

5) **Summation Function:** computes the weighted sums of all the input elements entering each processing element. A summation function multiplies each input value by its weight and totals the values for a weighted sum *Y*. The formula for *n* inputs in one processing element Eq. (2.1) is:

$$Y = \sum_{i=1}^{n} X_i W_i + B_i \qquad \text{Eq. 2.1}$$

For the ***j*th** neuron of several processing neurons in a layer, the formula is:

$$Y_j = \sum_{i=1}^{n} X_i W_{ij} + B_{ij} \qquad \text{Eq. 2.2}$$

6) **Transformation (Transfer) Function:** the output of a neuron must be passed and computed through a specific activation function. Based on this level, the neuron may or may not produce an output mean that if the output is "1" means that the neuron is activated otherwise it is "0". The **activation function** combines (i.e., adds up) the inputs coming into a neuron from other neurons and then produces an output based on the choice of the transfer function. Selection of the specific function affects the network's operation. The sigmoid (logical activation) function (or sigmoid *transfer function*) is an S-shaped transfer function in the range of "0" to "1" see Figure (2.7), and it is a popular as well as useful nonlinear transfer function [38]:

$$f(Y) = \frac{1}{\left(1 + e^{-Y_j}\right)} \qquad \text{Eq. 2.3}$$

Where *f*(Y) is the transformed (i.e., normalized) value of *Y*. This transformation is performed before the output reaches the next level. Without such a transformation, the value of the output becomes very large, especially when there are several layers of neurons.

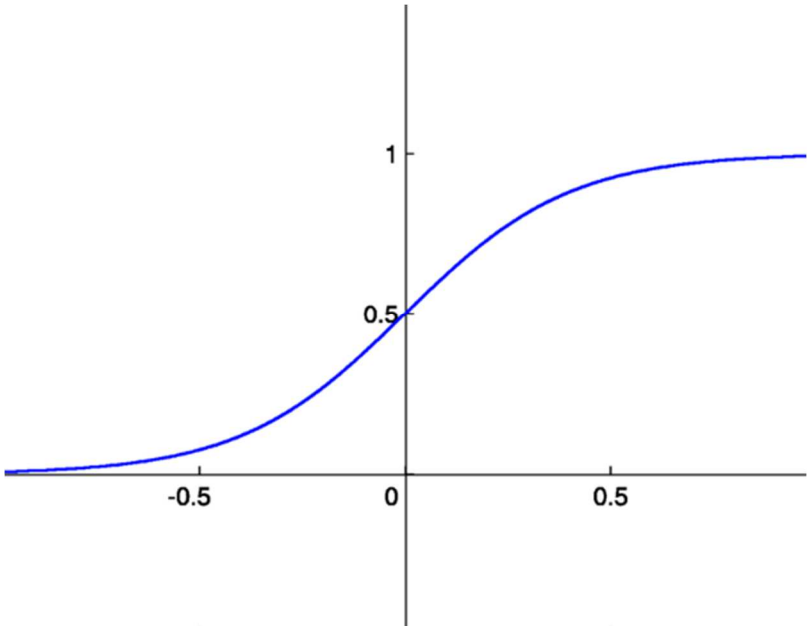


Figure (2.7): Sigmoid Activation Function [40]

Sometimes, instead of a transformation function, a *threshold value* is used. A threshold value is a hurdle value for the output of a neuron to trigger the next level of neurons. If an output value is smaller than the threshold value, it will not be passed to the next level of neurons. For example, any value of 0.5 or less becomes 0, and any value above 0.5 becomes 1. A transformation can occur at the output of each processing element, or it can be performed only at the final output nodes, see Figure (2.8) [38].

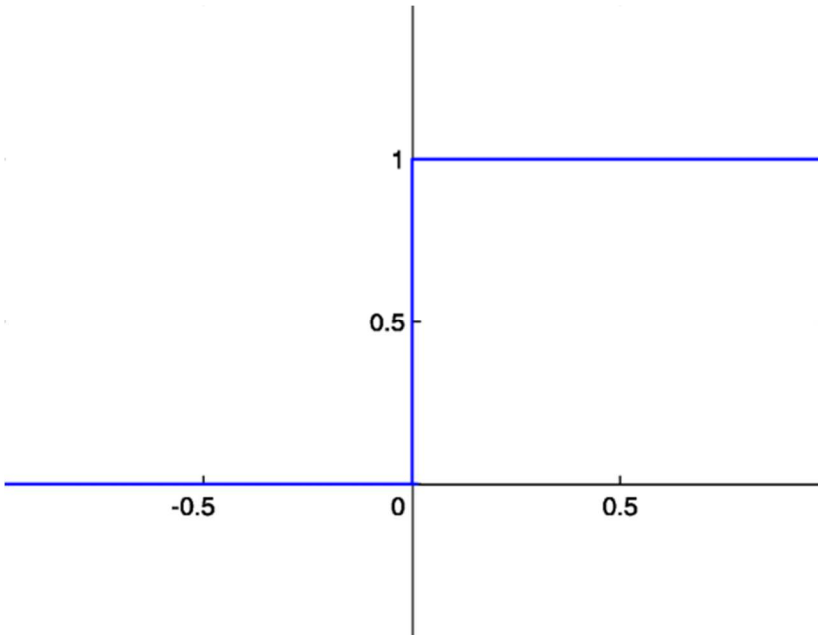


Figure (2.8): Threshold Activation Function [40]

2.9.3 Neural Network Learning Process

The following diagram Figure (2.9) presents how a single neuron in a network learns; where if we have two input values with a single output in the network. The neuron must be trained to recognize the input patterns and classify them to give the corresponding outputs. These input values must calculate with weights, then adjusting these weights with each train case until reaching the best weights. All of these goes through a number of steps:

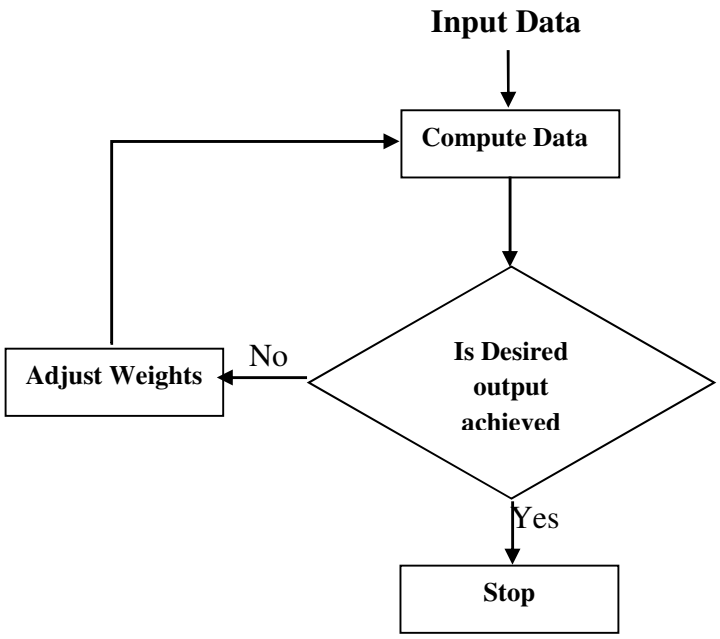


Figure (2.9): Learning Process of ANN [38]

The procedure is to present to the neuron the sequence of the input patterns; these input values must be computed by the summation function (Eq. 2.2). The output of the previous step must be computed by an activation function threshold or sigmoid activation function, (Eq. 2.3). After calculating outputs, a measure of the error (i.e., delta) between the output and the desired values is used to update the weights, subsequently reinforcing the correct results. At any step in the process for a neuron *j* we have:

$$delta = Z_j - Y_j \qquad \text{Eq. 2.4}$$

Where *Z* and *Y* are the desired and actual outputs, respectively. Then, the updated weights are:

$$W_i(final) = W_i(initial) + alpha \times delta \times X_i \qquad \text{Eq. 2.5}$$

This step is repeated until the weights converge to a uniform set of values that allow the neuron to classify each of the inputs correctly, where *alpha* is a parameter that controls how fast the learning takes place. This is called a *learning*

rate. The choice of the learning rate parameter can have an impact on how fast and how correctly a neural network learns. A high value for the learning rate can lead to too much correction in the weight values, resulting in going back and forth among possible weights values and never reaching the optimal, which may lie somewhere in between the endpoints. Too low a learning rate may slow down the learning process. In practice, a neural network analyst may try using many different choices of learning rates to achieve optimal learning. Most implementations of the learning process also include a counter balancing parameter called *momentum* to provide a balance to the learning rate. Essentially, whereas learning rate is aimed at correcting for the error, momentum is aimed at slowing down the learning [38].

2.9.4 The Advantages of Neural Networks

Advantages of neural networks can be outlined as follows [37]:-

1) High Accuracy: Neural networks are able to approximate complex non-linear mappings.
2) Noise Tolerance: Neural networks are very flexible with respect to incomplete, missing and noisy data.
3) Independence from prior assumptions: Neural networks do not make priori assumptions about the distribution of the data, or the form of interactions between factors.
4) Ease of maintenance: Neural networks can be updated with fresh data, making them useful for dynamic environments.
5) Neural networks can be implemented in parallel hardware.
6) When an element of the neural network fails, it can continue without any problem by their parallel nature.

2.10 Fuzzy Rough Nearest Neighbor

Fuzzy Rough Nearest Neighbor (FRNN) was an improved version of K Nearest Neighbor (KNN), where an important drawback of the KNN algorithm is that it considers each of the K neighbors as equally important during the classification of a target instance t, independent of the neighbor's distance to t. To overcome this problem, fuzzy set theory was introduced into the classical KNN decision rule. By means of an indiscernibility relation, instances can now partially belong to the set of nearest neighbors and are weighted accordingly. Later on, two other techniques that aim to improve Fuzzy Nearest Neighbor by means of fuzzy rough set theory were introduced [41]. The concept of Fuzzy Rough Set Theory can be constructed based on two other theories that are rough set theory and fuzzy set theory [42]. The two theories are explained as follows: -

2.10.1 Rough Set Theory

The core of Rough Set Theory (RST) is described via the indiscernibility concept. Let *(X, A)* be an information system, where *X* is a non-empty set of finite objects (the universe of discourse) and *A* is a non-empty finite set of attributes such that $a : X \rightarrow Va$ for every $a \in A$. *Va* is the set of values that attribute *a* may take. With any $B \subseteq A$ there is an associated equivalence relation R_B:

$$R_B = \{(X, Y) \in X^2 \mid \forall a \in B, a(X) = a(Y)\} \qquad \text{Eq. 2.6}$$

If $(x,y) \in R_B$, then *x* and *y* are indiscernible by attributes from *B*. The equivalence classes of the B-indiscernibility relation are denoted $[x]_B$. Let $A \subseteq X$. *A* can be approximated using the information contained within *B* by constructing the *B-lower* and *B-upper* approximations of *A*:

$$R_B \downarrow A = \{x \in X \mid [x]_B \subseteq A\} \qquad \text{Eq. 2.7}$$

$$R_B \uparrow A = \{x \in X \mid [x]_B \cap A \neq \emptyset\} \qquad \text{Eq. 2.8}$$

The tuple ($R_B \downarrow A$, $R_B \uparrow A$) is called a rough set [43].

2.10.2 Fuzzy Set Theory

The fuzzy set theory (FST) is implemented to detect the imprecision existent in the data sets which would be difficult by using conventional set theory in which elements could belong to either a set or no. This idea is protracted via FST, in which degrees of membership of elements to sets are allowed. Prior to the establishment of the concept of FST, elements would either be a membership of 1 or a membership of 0. This limitation is removed via using FST through allowing memberships to take values in between [0, 1]. A set of $A = \{x, \mu A \mid x \in U\}$ is a fuzzy set. The function $\mu A(x)$ is the membership function for A, literally, this means representing each element of the universe U to a membership degree in between [0, 1] figure (2.10) represented the concepts of membership function. A normal fuzzy set which includes at least one element with a membership degree of 1, notice that the universe may be discrete or continuous [42].

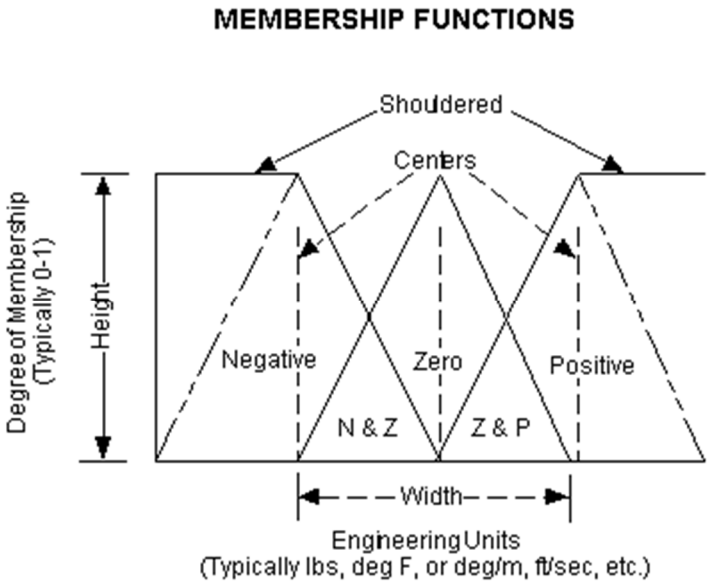


Figure (2.10): The Concept of Membership Function [54]

2.10.3 Fuzzy Rough Set Theory

The main purpose of the fuzzy set theory approach is to create vague information, whereas the rough set theory approach creates imperfect information. Both approaches are not opposing but are complementing each other. Therefore, this approach of fuzzy rough set can be described to focus predominantly on fuzzifying the lower and upper approximations, and is obtained by encompassing the conforming crisp rough set notions. The fuzzy P-lower and P-upper approximations are described via Eq. (2.9), and Eq. (2.10) as follow:

$$(R \downarrow A)(x) = \inf_{y \in U} I(R(x, y), A(y)) \qquad \text{Eq. 2.9}$$

$$(R \uparrow A)(x) = Sub_{y \in U} T(R(x, y), A(y)) \qquad \text{Eq. 2.10}$$

Here, *I* is an implicator and *T* is a t-norm. When *A* is a crisp (classical) set and *R* is an equivalence relation in *X*, the traditional lower and upper approximation are recovered [42].

Fuzzy Rough Nearest Neighbor concepts was from combining fuzzy rough approximations with the ideas of the classical Fuzzy Nearest Neighbor approach, which can be seen in Appendix B3. Where the nearest neighbors are used to construct the fuzzy lower and upper approximations of decision classes, and test instances are classified based on their membership to these approximations [43]. The FRNN algorithm first checks the *K* nearest neighbors of a desired sample *t* and then categorizes the desired target instance to the class *C* in which the sum is maximal, with *R* a fuzzy indiscernibility function [42], this is expressed in Eq. (2.11).

$$(R \downarrow C)(t) + (R \uparrow C)(t) \qquad \text{Eq. 2.11}$$

The upper and lower approximations only consider the examples of NN and are expressed a follows:-

$$(R \downarrow C)(t) = \min_{x \in NN} I(R(x,t), C(x)) \quad \text{Eq. 2.12}$$

$$(R \uparrow C)(t) = \max_{x \in NN} T(R(x,t), C(x)) \quad \text{Eq. 2.13}$$

Where *(R↓C) (y)* value is great, then, it means every value of *y's* neighbors is included in class *C*. A high value of *(R↑C),* would state that at least one neighbor is included in the class [42].

2.11 Decision Tree

A decision tree (DT) is a classifier expressed as a recursive partition of the instance space. The decision tree consists of nodes that form a *rooted tree*, meaning it is a *directed tree* with a node called "root" that has no incoming edges. All other nodes have exactly one incoming edge. All other nodes are called leaves (also known as terminal or decision nodes). According to a certain discrete function, the instance space splits into two or more subspaces by the internal node. In the simplest and most frequent case, each test considers a single attribute, such that the instance space is partitioned according to the attribute's value. In the case of numeric attributes, the condition refers to a range [27].

Each leaf is assigned to one class representing the most appropriate target value. Alternatively, the leaf may hold a probability vector indicating the probability of the target attribute having a certain value. Instances are classified by guiding them from the root of the tree down to a leaf, according to the outcome of the tests along the path [27]. The structure of working this classifier is explained in Appendix B4.

2.12 Naïve Bayes

The Naive Bayes algorithm is a simple probabilistic classifier that calculates a set of probabilities by counting the frequency and combinations of values in a

given data set. The algorithm uses Bayes theorem and assumes all attributes to be independent given the value of the class variable [44].

More formally, this classifier is defined by discriminant functions:

$$f_i(x) = \prod_{j=1}^{N} p(xj/ci)p(ci) \qquad \text{Eq. 2.14}$$

Where $X = (x1, x2... xN)$ denotes a feature vector and $ci, i = 1, 2,..., N$, denotes possible class labels.

The training phase for learning a classifier consists of estimating conditional probabilities $P(xj/ci)$ and prior probabilities $P(ci)$. Here, $P(ci)$ are estimated by counting the training examples that fall into class ci and then dividing the resulting count by the size of the training set. Similarly, conditional probabilities are estimated by simply observing the frequency distribution of feature xj within the training subset that is labeled as class ci. To classify a class-unknown test vector, the posterior probability of each class is calculated, given the feature values present in the test vector; and the test vector is assigned to the class that is of the highest probability [45].

2.13 Particle Swarm Optimization

Particle Swarm Optimization (PSO) is one of the optimization methods used for obtaining the best results with the data mining techniques. It was introduced in 1995 by Kennedy and Eberhart, which is used for optimizing hard numerical functions on metaphor of social behavior of flocks of birds and schools of fish [46, 47]. In this algorithm, each solution of the optimization problem is like searching a bird in the space, which is called as the "particle" [48]. These particles fly around in a multidimensional search space. During flight, each particle adjusts its position according to its own experience, and the experience of neighboring particles. Thus each particle makes use of the best position encountered by itself and its neighbors

[46]. Four vectors are included with each particle, assuming that the current position of the *i*-th particle in D-dimension is $X_i = (x_{i1}, x_{i2}, \ldots, x_{iD})$, the best position founded for the particle is represented by $P_i = (p_{i1}, p_{i2}, \ldots, p_{iD})$, the best position found by its neighborhood so far $n_g = (n_{g1}, n_{g2}, \ldots, n_{gD})$ and $V_i = (v_{i1}, v_{i2}, \ldots, v_{iD})$ is its velocity which represents its direction of searching. In iteration process, each particle keeps the best position p_{id} found by itself, besides, it also knows the best position n_{id} searched by the neighborhood particles, and changes its velocity according to two best positions [49]. The velocity and the position for each particle are calculated by using the following formula:

$$v_{id}^{k+1} = w v_{id}^{k} + c_1 r_1 (p_{id} - x_{id}^{k}) + c_2 r_2 (n_{gd} - x_{id}^{k}) \qquad \text{Eq. 2.15}$$

$$x_{id}^{k+1} = x_{id}^{k} + v_{id}^{k+1} \qquad \text{Eq. 2.16}$$

In which: $i = 1,2,\ldots,N$; N is the population of the group particles; $d = 1,2,\ldots,D$ is the dimension space; k is the maximum number of iteration; r_1 ,r_2 are the random values between [0,1], which are used to keep the diversity of the group particles; c_1 ,c_2 are the learning coefficients, also are called acceleration coefficients; v_{id}^{k} is the number of d component of the velocity of particle i in k-th iterating in Eq. (2.15) [49]. And x_{id}^{k+1} is the new position of the particles, x_{id}^{k} is the previous position, and v is the velocity within Eq. (2.16).

The procedure of standard PSO is as following:

1) Initialize the original position and velocity of particle swarm;
2) Calculate the fitness value of each particle;
3) For each particle, compare the fitness value with the fitness value of *pbest*, if current value is better, then renew the position with current position, and update the fitness value simultaneously;

4) Determine the best particle of group with the best fitness value, if the fitness value is better than the fitness value of *gbest*, then update the *gbest* and its fitness value with the position;
5) Check the finalizing criterion, if it has been satisfied, quit the iteration; otherwise, return to step 2 [49].

2.14 Performance Measurement

When a system implemented or proposed there must be measurements for the accuracy performance of the system. One of the important measurement units is the confusion matrix, which can be used for detecting the number of correctly classified instances, incorrectly classified instances, Table (2.1) illustrated the components of the confusion matrix [51].

Classified As	**P**ositive	**N**egative
Positive	TP	FN
Negative	FP	TN

Table (2.1): Confusion Matrix Components

Where:

True Positive (TP): is the classification of an instance with **positive** class value as a **positive** case.

False Negative (FN): is the classification of an instance with **positive** class value as a **negative** case.

False Positive (FP): is the classification of an instance with **negative** class value as a **positive** case.

True Negative (TN): is the classification of an instance with **negative** class value as a **negative** case.

Here are some of the performance measures that is evaluated to assess the proposed system's performance [51]:

Sensitivity or True Positive Rate (TPR) = TP/ (TP+FN); Eq. 2.18

Miss rate or False Negative Rate (FNR) = FN/ (FN+TP); Eq. 2.19

Fall-out or False Positive Rate (FPR) = FP/ (FP+TN); Eq. 2.20

Specificity or True Negative Rate (TNR) = TN/ (TN+FP); Eq. 2.21

Accuracy of the implemented system are evaluated according to the following measurement:

Accuracy = (TP+TN)/ (TP+TN+FP+FN); Eq. 2.22

Commonly additional performance metrics used are referred to as, precision, Recall and F-measure:

Precision = TP/ (TP+FP); Eq. 2.23

Recall = TP/ (TP+FN); Eq. 2.24

F-measure = (2*precision*recall) / (precision + recall); Eq. 2.25

Mean Absolute Error (**MAE)** is the average absolute difference between classifier predicted output and actual output, while Root Mean Square Error (**RMSE)** is the square root of the **Mean Square Error (MSE)**, which is the average of the sum of squared differences between classifier predicted output and actual output [52,53].

$$MAE = \frac{1}{N}\sum_{1}^{N}|Desired\ output - Actualoutput| \quad \text{Eq. 2.26}$$

$$MSE = \frac{1}{N}\sum_{1}^{N}(Desired\ output - Actualoutput)^2 \quad \text{Eq. 2.27}$$

$$RMSE = \sqrt{1/N\sum_{1}^{N}(Desired\ output - Actualoutput)^2} \quad \text{Eq. 2.28}$$

Chapter Three

Proposed System Methodology

Chapter Three

Proposed System Methodology

3.1 Introduction

In the previous two chapters, the importance of managing human resource management and its role in improving productivity and quality of service, then human resource management theory are discussed with the existent problems and solving these problems by using data mining techniques as a system that can be useful for organizations.

In this chapter, the structure of the proposed system for human resource management will be described in detail, starting with the data collection, preprocessing, and explaining the stages of implementing the proposed system with classification techniques and optimization technique for getting better results, testing the proposed system.

3.2 System Structure

The proposed system for HRM is going through the number of steps, as they are presented in Figure (3.1), Figure (3.2), and Figure (3.3). They will be described in the following subsections in detail.

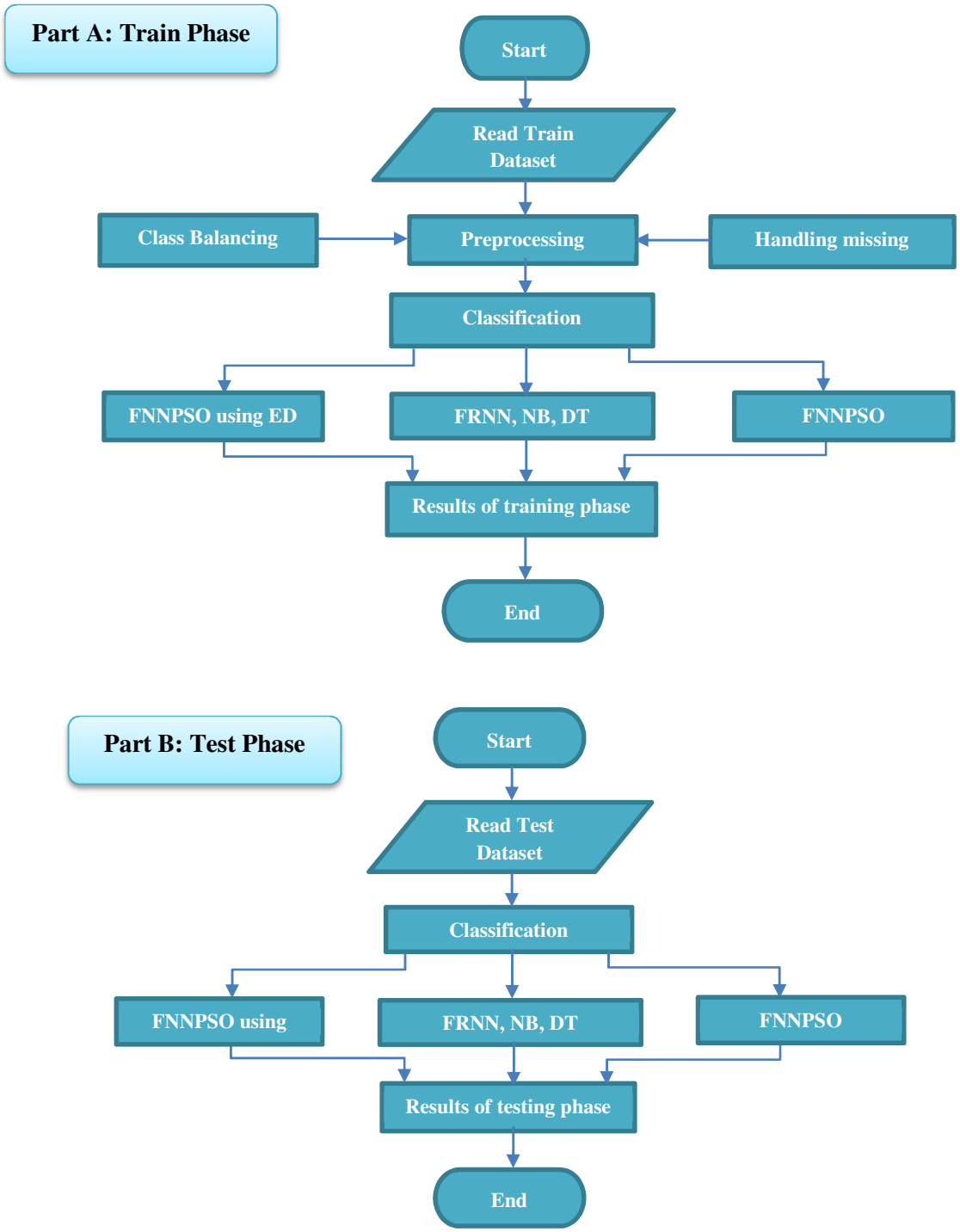


Figure (3.1): Proposed System for Human Behavior Decision Making.
Part A and B represent the train and test phases of proposed system

Where FNNPSO is the used standard system, natural inspired algorithm PSO is used for training the network and getting suitable weights and biases.

FNNPSO using ED is the improved version of the standard system, here Euclidian Distance is used as a random number generator for PSO. Results and effect on the classification accuracy will be described later in Chapter Four. FRNN, NB, DT classification techniques also used in this thesis.

3.3 Data Collection

For any research and information system an appropriate dataset is required. A common and useful way to collect data can be via a survey approach which has the advantage of being very structured, in addition, it is easily replicable, and possible to compare the results with surveys that had been previously undertaken. Researchers who are interested in the results are actually not physically close to participants who fill in the survey. Thus, it also allows for privacy and anonymity, and facilitate people to respond in a more honest way. Surveys can be carried out by a large number of participants. Thus, because of these advantages of survey approach, the data collection in the research work is done through a survey that was given to participants from companies and organizations in Kurdistan.

Table (3.1), shows relevant variable or attributes for an employee that contains 30 attributes, 29 of them are employee features and 30^{th} is the class or the decision for each case, the table has two parts, one part is filled by the employee, and the other part is filled by the director or the supervisor who oversees the employee where the class is recommendation for promotion and its value must be Yes or NO [42]. Each feature contains at least two values that an employee can tick it according to his or her information and his experience.

Table (3.1): Relevant features and attributes for employee data set

Filled by employees			
No.	***Variable name***	***No.***	***Variable name***
1	**ID**	**17**	**Department**
2	**Gender**	**18**	**Computer skills**
3	**Age**	**19**	**Job security**
4	**Education background (qualification)**	**20**	**Smoking**
5	**Language**	**21**	**Transportation**
6	**Marriage**	**22**	**Vacation days**
7	**Partner working**	**23**	**Nationality**
8	**Number of children**	**24**	**Employment type**
9	**Average age of their children**	**Filled by Director**	
10	**Resident**		
11	**Job time**	**25**	**Number of activities**
12	**Hours of work**	**26**	**Number of penalties**
13	**Salary**	**27**	**Term Reason**
14	**Years of service**	**28**	**Rise in income received**
15	**Social assurance**	**29**	**Employee disciplined**
16	**Position**	**30**	**Recommendation for promotion**

3.4 Data Analysis and Preprocessing

As it was explained in the previous chapter any dataset may contain noise, missing value…etc., that affects the quality and the performance of the information systems. Accordingly, noisy data set must be cleaned from the noise, outlier, and replaced the missing value by using data mining techniques to gain better learning and results. After collecting all the questionnaires, the process of preparing the data was accomplished. Some attributes like age and years of service are entered in continuous values. So, they are modified and illustrated via ranges. Other attributes like Language is generalized to include fewer discrete values than they already have.

So, our dataset contained missing values especially in Salary attribute because the respondent did not answer all questions in the questionnaire during the dataset collection. Thus, each missing value is replaced with the mean of the attribute, this is how the missing values are treated. Obviously, the mean is calculated according to all known attribute values. This method is convenient with numeric attributes only. Subsequently it is used for handling missing values in the Salary attribute [42].

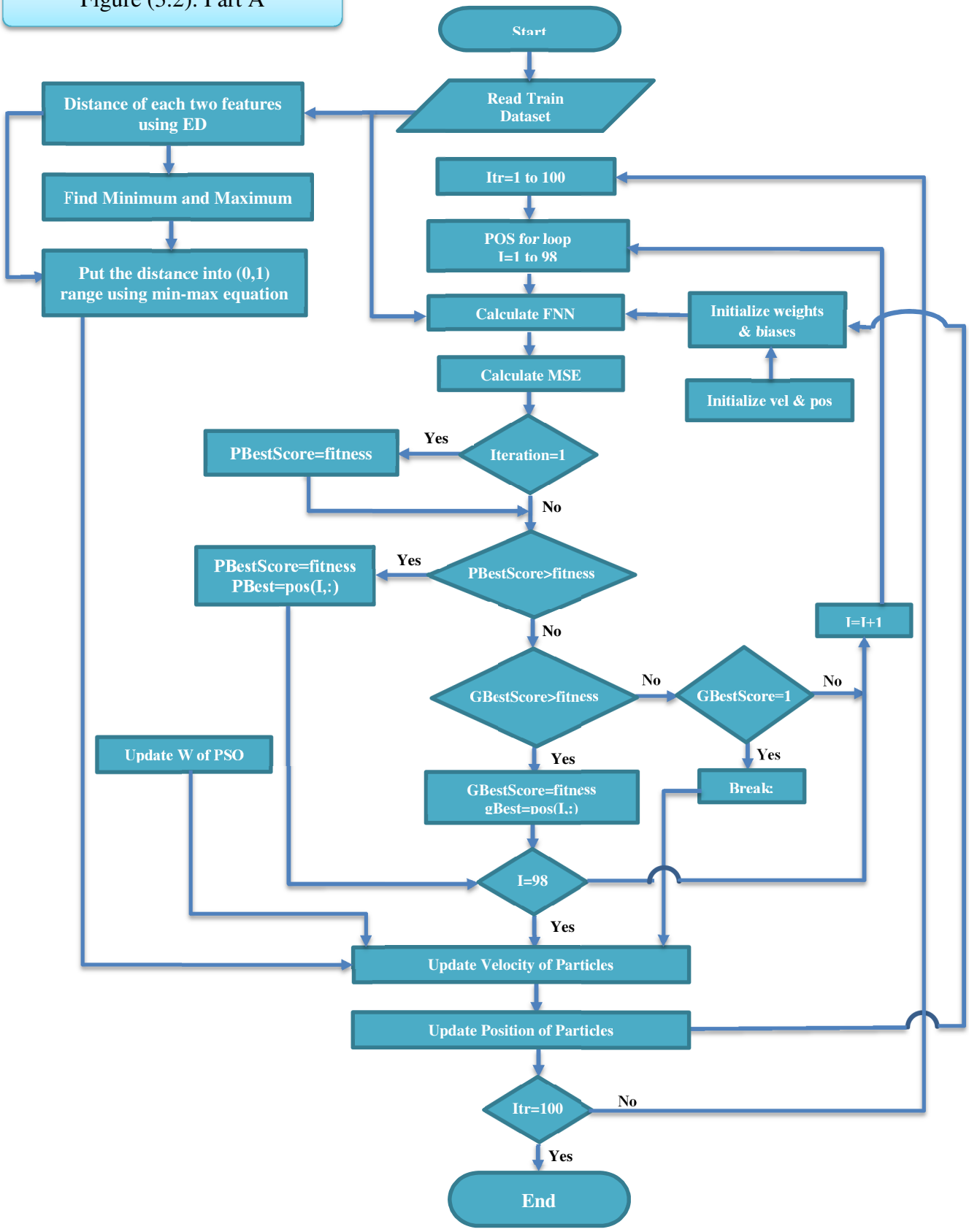
After that, the prepared excel sheet file converted into (.arff) file format to be compatible with WEKA data mining tool kit. For handling these missing values *ReplaceMissingValues* method is used when replacing each missing value with the mean of the attribute.

Imbalanced class was detected in our dataset, which is affecting the quality of the data. So, for better results and for the best prediction, SMOTE method is used for balancing the used dataset.

3.5 The Proposed System for HRM

The FNNPSO system is enhanced using Euclidian Distance equation. Then FNNPSO via ED is regarded as a new version called FNNPSOED is used for obtaining the distance between each of the two features in the training dataset, then using this obtained distance in PSO optimization technique in updating the velocity of particles. In this research and case study, the obtained distance is used instead of random number in updating velocity in PSO. Figure (3.2) explained the proposed system for training Forward Neural Network in detail which has two parts. Part A is the primary part for gaining weights and biases for FNN, whereas Part B is the calculation of FNN and gaining the accuracy of results.

Figure (3.2): Part A



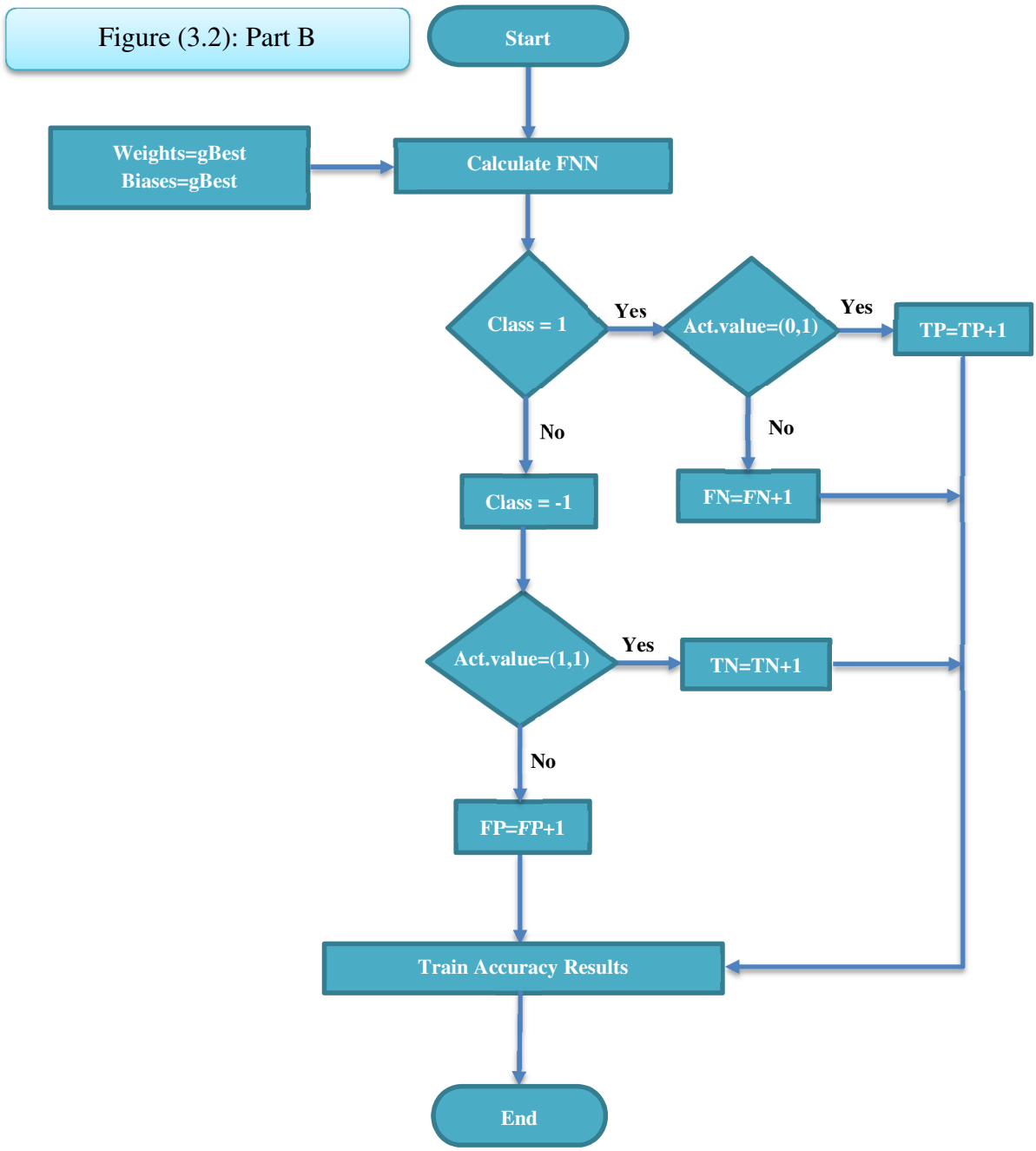


Figure (3.2): Proposed Algorithm at Training Phase with Two Parts
Part A: represented procedure of getting Weights and Biases
Part B: is the calculation of the classification rate and the accuracy

Figure (3.2) consists of two parts, its variables can be described as follows: Part A: Is the important part in this work where consists of reading the training dataset which is used with calculating FNN and ED, then min-max is used with Eq. (3.3)

for standardizing or normalizing the distance from the previous step, *vel* is the velocity of particle *i* at each iteration; *pos* is the position of each particle in the dimension of particles; *pBestScore* represents a memory of the previous best position which is compared with the fitness for getting the *pBest* value and updating the velocity of particles; *pBest* is the best solution that is stored in the *pBestScore* memory which is the cognitive component; *gBestScore* is the memory of the best solution visited by any particle; *gBest* represents the social components is the global best position which is the solution. *pBest* and *gBest* are the factors that helped in updating the velocity and position of the particles in the dimension, *w* is the Inertia Weight used to control the velocity.

Part B: *R* is the Rate counter for calculating the classification rate, *TP* is the True Positive case counter, also (*FN,TN,FP*) are used for confusion matrix calculation as counter variables which are namely False Negative, True Negative, False Positive explained in the previous chapter.

The proposed system is tested for obtaining its performance according to a number of steps. These steps are illustrated in detail and by steps in Figure (3.3) which explains the tested system that has two parts; Part A and Part B. The first part explains the initialization of FNN parameters and calculating MSE for gaining error rate in the proposed system. The second part contained explaining calculating FNN and getting accuracy results. Also it has a number of variables mostly defined in the previous sections. Here Weights and Biases are the best solutions that are obtained from the training phase of the implemented system which represents the important main part of the proposed system.

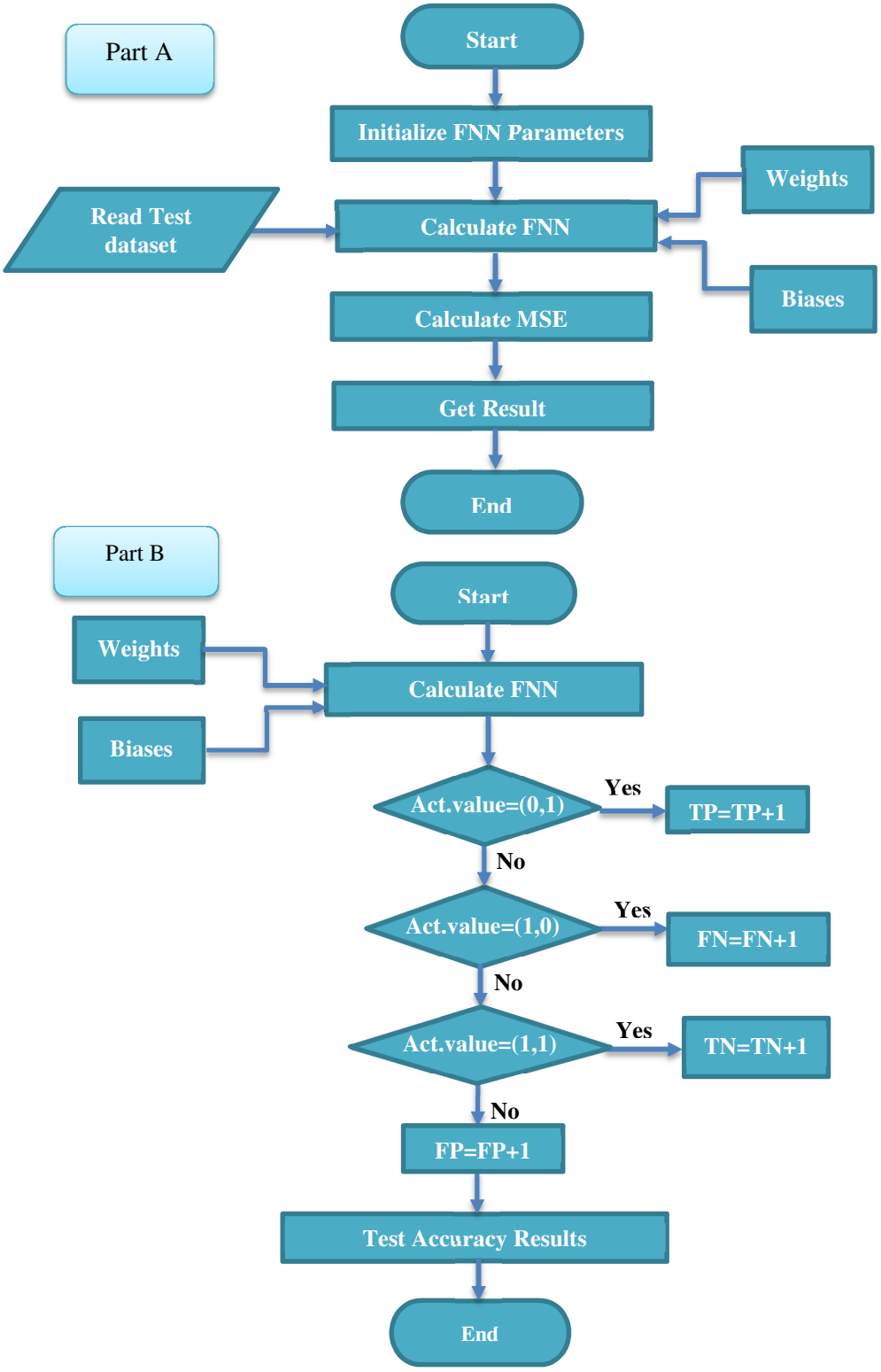


Figure (3.3): Testing the Proposed Algorithm Using FNNPSOED
Part A: represented the calculation of MSE
Part B: represented the calculation of accuracy

3.6 Classification

In this thesis, Feed Forward Neural Network is used for classification in the learning and testing phases, for decision on employee behavior in private and public sector companies according to the collected data from the companies. The used FNN consisted of layers namely are: input, hidden and output layers. In fact, FNNs with one hidden, output layers are the most popular neural networks with most practical applications. Each layer consisted of a number of neurons. Each neuron in one layer is connected to neurons in the succeeding layer via a link which contains specific weight and bias values. Neuron values must be calculated with each weight and biases for getting the next layers node value. The goal is to find the best combination of connection weights and biases in order to achieve the minimum error. According to this, Forward Neural Network is highly dependent on the initial values of weights, biases, and its parameters. These parameters include learning rate, momentum and hidden nodes in the hidden layer.

Figure (3.4) shows the structure of FNN with 29 input nodes, 38 hidden nodes, 2 output nodes.

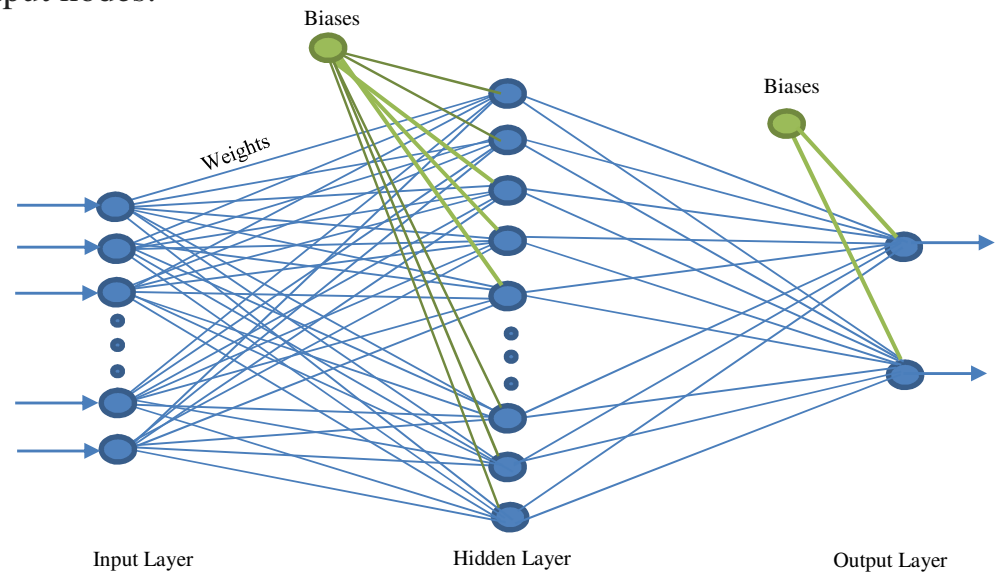


Figure (3.4): Forward Neural Network Structure

FNN in this case study, works in two phases which are training and testing phases. The first phase attempts to train the network as a supervised system. Each instance or case is composed of two parts, input and the output that is the desired or target that must be reached. In the supervised models the obtained results are compared with the actual class value in each iteration. Also, there is unsupervised model for learning. In this model, the network requires to reach the target without having the class that can be compared with the output (which is not our case study).

In this model, the employee dataset are used as the input data for the system, where each feature value in each instance acts as an input neurons value in each train case.

After getting the input data, it must be fed to the hidden layer neurons, then after calculating each value with neuron weights and biases and then applying the activation function, finally, the result acts as the input for the next layer.

The class in the dataset is used for measuring the performance of the system and determining error rate by comparing each of the class with the output of the system after getting the input from the previous hidden neuron.

According to the calculation of the network, weights and biases are updated continuously after each training case in the learning phase. After getting the best or acceptable output value with the least error rate or completing the training, then the system must be terminated.

When the learning phase is completed, weights and biases are gained, the second phase must be accomplished which is the testing phase. The saved weights and biases must be used in this phase. Finally, the testing dataset must be loaded into the system to evaluate the network performance.

Besides of using FNN, other classifier techniques are used in this thesis, such as Fuzzy Rough Nearest Neighbors, Decision Tree, and Naïve Bayes. Each one will be discussed in the following sections.

Fuzzy Rough Nearest Neighbor is used in this thesis for classifying and predicting employee behavior. This technique tries to solve the crisp problem in other techniques that is very important in many cases. In other words, the values of a specific instance of any feature must classify and belong to class yes or not, where this a problem. For example, if a value has very little difference with the actual value that must be, this mean that the case shouldn't belong to that class and this is not a judgment for deciding about this case. The main concept of this approach for solving crisp problem is that the lower and upper approximations of a decision class are premeditated via the nearest neighbors of a test *y* object.

The FRNN algorithm first checks the *K* nearest neighbors of a desired sample *t* and then categorizes the desired target instance to the class *C* in which the sum is maximal, with *R* a fuzzy indiscernibility function.

When $(R \downarrow C)(y)$ value is great, then, this means every value of *y's* neighbors is included in class *C*. A high value of $(R \uparrow C)$, would state that at least one neighbor is included in the class [42].

For determining better learning and results, Decision Tree is also used as a classification technique. According to the *IF…THEN* rule *DT* can build a tree starting from the root of the tree, that must be one of the attributes in the dataset and going through the path until reaching the leaf of the tree. Where the root is the starting point (a feature that leads us to the goal with the few number of nodes and paths), the path is the link between nodes according to the *if…then* rule as mentioned above. Finally, leaf is the goal that wanted to reach or it is the class. It

is clear that every attribute value pair alongside a specified path can create a conjunction in the rule antecedent or the *IF* part. The leaf node can have the class prediction, which would create the rule consequent or *then* part [42].

The last classifier which is used in this thesis for predicting employee behavior is Naïve Bayes classifier which is based on Bayes rule to train a classifier that will output the probability distribution over possible values of Y, for each new instance X that is asked to classify. In naïve bayes learning, each instance is described by a set of features and takes a class value from a predefined set of values. When a feature is assumed to be class-conditionally independent, it really means that the effect of a variable value on a given class is independent of the values of other variables that dramatically reduces the number of parameters to be estimated.

3.7 Euclidean Distance

As a definition to distance measure, can be supposed that there are a set of points, called a space. A distance measure on this space is a function $d(x, y)$ that takes two points in the space as arguments and produces a real number, and satisfies the following axioms:

1) $D(x, y) \geq 0$ (no negative distances).
2) $D(x, y) = 0$ if and only *if* $x = y$ (distances are positive, except for the Distance from a point to itself).
3) $D(x, y) = d(y, x)$ (distance is symmetric).
4) $D(x, y) \leq d(x, z) + d(z, y)$ (the triangle inequality).

The most familiar distance measure is the one we normally think of as "distance". An n-dimensional Euclidean space is one where points are vectors of n real numbers. The conventional distance measure in this space, which is defined:

$$d([x_1, x_2, ..., x_n], [y_1, y_2, ..., y_n]) = \sqrt{\sum_{i=1}^{n} (x_i - y_i)^2} \qquad \text{Eq. 3.1}$$

Where *Xi* represented the instances in the first attribute and *Yi* represented the second attribute instances.

That is, the distance in each dimension must be squared, sum the squares, and take the positive square root. It is easy to verify the first three requirements for a distance measure as satisfied. The Euclidean distance between two points cannot be negative, because the positive square root is intended. Since all squares of real numbers are nonnegative, any i such that $x_i \neq y_i$ forces the distance to be strictly positive. On the other hand, if $x_i = y_i$ for all i, then the distance is clearly 0. Symmetry follows because $(x_i - y_i)^2 = (y_i - x_i)^2$. The triangle inequality requires a good deal of algebra to verify. However, it is well understood to be a property of Euclidean space: the sum of the lengths of any two sides of a triangle is no less than the length of the third side [50].

3.8 Modified Particle Swarm Optimization

To find the best weight and biases for neural network, PSO is used. These weights and biases are used for testing neural networks in the test phase. After initializing particles for PSO with random numbers, evaluation of the desired optimization fitness function for each particle is done. By comparing this fitness function evaluation value with its *pBest*, if current value is better than *pBest*, then *pBest* is equal to the current value, and P_i is equal to current location X_i.

After that, each of position and velocity of particles must be updated according to their experience and other companions experience to reach the best position. In this thesis, random number generator is proposed for PSO which can update particles velocity by using ED equation for determining the distance between feature instances in each training case and using this distance instead of random number as a matrix of vectors that exists in Eq. (2.15) after standardizing this distance to become as follows:

$$v_{id}^{k+1} = wv_{id}^{k} + c_1 \overrightarrow{ED} \ (p_{id} - x_{id}^{k}) + c_2 \overrightarrow{ED} \ (n_{gd} - x_{id}^{k}) \qquad \text{Eq. 3.2}$$

Where *ED* is the normalized Euclidian distance value where its value gained by using Eq. (3.3). By taking each two features as input to the Eq. (3.1) and obtaining these distances then it must be normalized to be between the (0,1) interval by using the following equation:

$$ED = (d - \min / \max - \min) \qquad \text{Eq. 3.3}$$

Where *d* is the calculated distance from Eq. (3.1), *min* is the minimum value within the distance values, *max* is the maximum value in the gained distances. Standardized value is the generated number that must be used in the Eq. (3.2) for updating the velocity of the particles with the PSO.

Inertia Weight which is *w* in Eq. (3.2) is updated using the following equation:

$$W = W_{\max} - Iteration * (W_{\max} - W_{\min}) / Max\_Iteration \qquad \text{Eq. 3.4}$$

Eq. (2.16) is used for updating the position of the particles in the dimension.

Figure (3.5) explains the mechanism working of PSO according to the new modification that are made in this thesis.

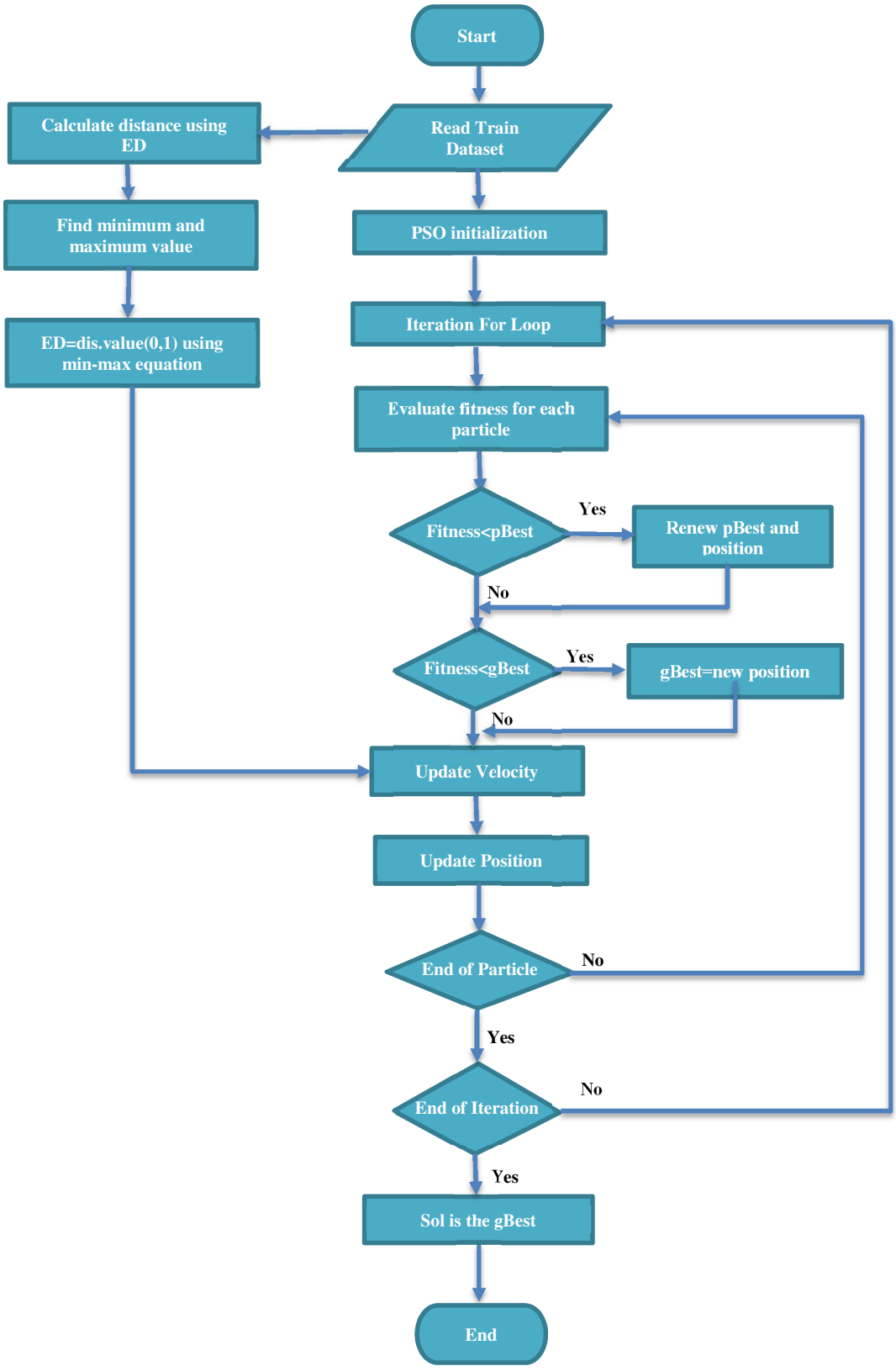


Figure (3.5): Structure of Improved PSO

3.9 Simulation Techniques Used for Proposed System

In any work, the necessary part that must exist as a complement part of the thesis is the practical part. In this thesis, a software is built for FNN with PSO using ED for optimizing its weights and biases using Matlab programming language R2015a, 64 bit. Besides that, wekaWEKA toolkit, version 3.7.12 are used for implementing other three techniques which are namely (FRNN, NB, and DT).

It is clear that using data in any system is the primary and the main part for that system. Here, in this thesis two types of file formats are used. After collecting data in excel sheet these data are converted into (.TXT, AREFF) file format for dealing with it.

For training and testing the proposed system (FNNPSOED) and the standard system FNNPSO with matlab language TXT file format is used, and for the other classification techniques with WEKA AREFF file format is used.

Chapter Four

Results and Discussions

Chapter Four

Results and Discussions

4.1 Introduction

Obtaining accurate results for a system is regarded as one of the important phases for any research work. In this chapter, the training and testing input data used for classifiers with their parameters, the results of some conducted experimental tests and various cases are studied to choose the most suitable models and assessment of the performance of the proposed system, is described. This chapter consists of three experimental cases. All the above mentioned points will be described in the following sections via visual tables and figures.

4.2 Training and Testing Dataset

Collected employee behavior dataset which contains 30 attributes, 29 of them are the features that describe the employees' behavior and the 30^{th} is the class that determines if the system recommends for promotion or not (Yes or No), depending on them, the proposed system can learn and make decisions. This dataset is divided into two parts namely training and testing datasets. The training dataset is used for learning the system and redirecting the system for making decisions in the other phases, whereas the testing dataset is used to evaluate the performance of the proposed system.

According to the searching and gathering information from the companies in Kurdistan, the collected data set contain a set of features each one has its own value and type. The collected dataset contains numeric and nominal feature values. A dataset with 800 instances is used for training both FNNPSO and FNNPSO

using ED. Four types of datasets are used for testing the above mentioned techniques with different number of instances, these datasets contain 350, 400, 500, and 600 instances.

A dataset with 800 instances is used for gaining the performance of the other three techniques. These techniques are FRNN, NB, and DT.

4.3 Experiment 1: Optimizing Forward Neural Network with PSO

In this stage, PSO is used for optimizing the weights and biases which provide a minimum error for FNN. Employee behavior dataset with 29 features and a class with (yes, no) values is used for recommending for promotion or not. The network has only one hidden layer with the output layer. Containing two model with different parameters in this Experiment as described bellow:

First Model: FNN parameters like input features, hidden neurons, number of output class, and Training number are presented bellow in Table (4.1) in the training phase of the FNN.

Table (4.1): First Model – FNN Classifier Parameters

Input Features	Hidden Neurons	Output Class	Training NO.
29	35	2	800

Table (4.2) shows the used parameters for PSO like number of particles (NoP), dimension of particles in PSO (Dim), inertia weight (IW) where it's value is updated beside of updating the velocity of particles, it's value after this update becomes between (0,1), max inertia weight (Wmax), min inertia weight (Wmin), acceleration coefficient (AC), Min Position within best positions of particles, Max

Position within particle positions, Max Iteration (Max_Itr), and momentum, which are used with FNN for getting better Weights and Biases values with the goal of decreasing the error rate in the training phase and getting higher accuracy and classification rate.

Table (4.2): First Model- PSO Optimizer Parameters

NoP	Dim	IW	Wmax	Wmin	AC	Min Position	Max Position	Max_Itr	Momentum
98	1998	2	0.7	0.5	2	-16.5574	39.6886	100	0.8

According to Table (4.2) the best positions of the particles is between two numbers that is differ from each other, where there is a big range in this case.

The confusion matrix for this model in the training and testing phase is presented in the Table (4.3). This table is divided into two parts A and B, where part A presents the result of the confusion matrix in the training phase with training dataset that contains 800 instances, and part B presents the confusion matrix in the testing phase with four testing datasets with different number of instances which are 350, 400, 500, and 600 instances. According to the Table (4.3) with its two parts, it is clear that there are a number of correctly classified instances, with a number of cases that are classified incorrectly in both cases. This makes the system to be fair to the error more than other systems with less incorrectly classified cases, which lead the system to misclassification during test phases as presented below.

Table (4.3): First Model: Confusion Matrix of Training and Testing Phase

Classified As	Yes	No
Yes	436	13
No	5	346

Part A: Confusion Matrix of Train Phase

Classified As	350 Test Dataset		400 Test Dataset		500 Test Dataset		600 Test Dataset	
	Yes	No	Yes	No	Yes	No	Yes	No
Yes	44	16	74	25	130	55	557	13
No	25	265	22	279	25	290	14	16

Part B: Confusion Matrix of Testing Phase

The evaluation results of FNNPSO for the first model in both training and testing phases are presented in Table (4.4). In this table each of Accuracy, CCI Correctly Classified Instances with its percentage, ICI Incorrectly Classified Instances with its percentage, Sensitivity, Fall-out, Specificity,…, and MSE which is the error rate in the system are presented. 800 instances are used for the training session and four different sets of dataset (350,400,500, and 600 instances) are used for testing session. This table presented that each case has a good accuracy but beside of that the number of error rate will affect the system with classifying instances it will be improved for getting lower error rate value and have a robust system for classifying any cases. In this table the relation between each of the accuracy and the error rate can be noticed. As the accuracy increased the rate of the error decreased and vice versa where the error rate in the train phase is low but this is different with the test cases. Besides that each of other measurements like CCI, ICI, and others related to the accuracy of the system. In this model the elapsed time of each case is presented in both train and test case for one iteration.

Table (4.4): First Model- Evaluation Results of FNNPSO (Training and Testing)

Parameter	Training	Testing			
	800	350	400	500	600
Accuracy	0.97750	0.88571	0.88250	0.84000	0.95500
CCI	782	309	353	420	573
CCI %	97.75 %	88.28 %	88.25 %	84.00 %	95.50 %
ICI	18	41	47	80	27
ICI %	2.25 %	11.71 %	11.75 %	16.00 %	04.50 %
Sensitivity	0.97104	0.73333	0.74747	0.84680	0.97719
Fall-out	0.01424	0.08620	0.07309	0.17730	0.46667
Specificity	0.98575	0.91379	0.92691	0.82270	0.53333
Miss-rate	0.02895	0.26667	0.25253	0.15320	0.022807
Precision	0.98866	0.63768	0.77083	0.92401	0.97548
Recall	0.97104	0.73333	0.74747	0.84680	0.97719
F-measure	0.97977	0.68217	0.75897	0.88372	0.97634
MSE	0.02654	0.54922	0.53896	0.55353	0.42634
Elapsed Time	0.011161s	0.0064138s	0.00085002s	0.00042449s	0.00041294s

Gained weights and biases in the first model that is used for building and testing the proposed system are presented in the Figure (4.1) and Figure (4.2). In these two figures, the range of weights and biases can be seen in the small domain and it is clear that most of the values are zeros or near from zero. This made the system not to accept noises and outliers softly, whereas it is possible if there are errors with entering data from the users, there will lead to misclassification with the system.

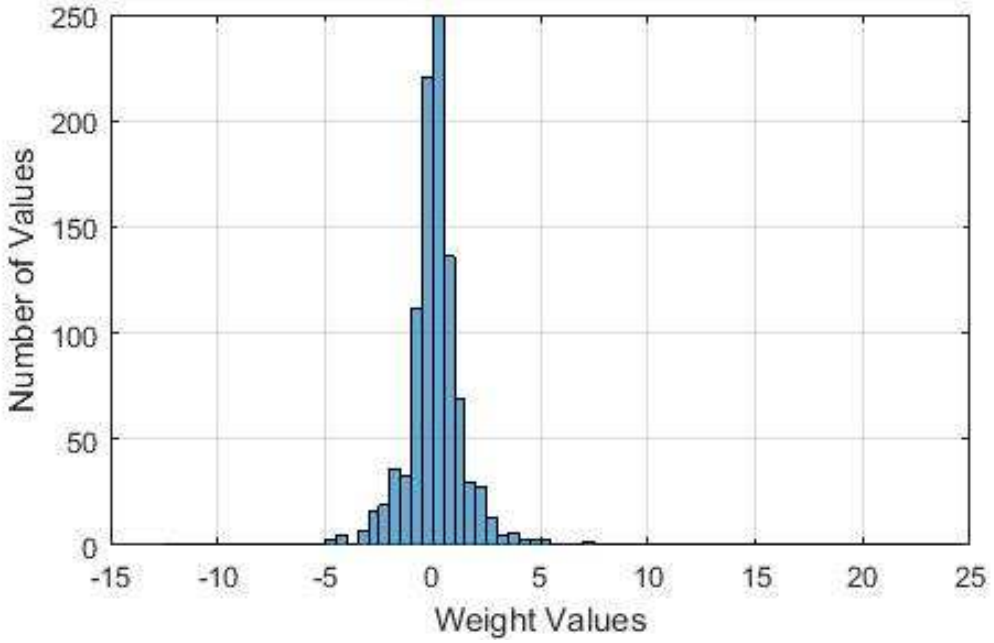


Figure (4.1): First Model - Obtained Weights with FNNPSO

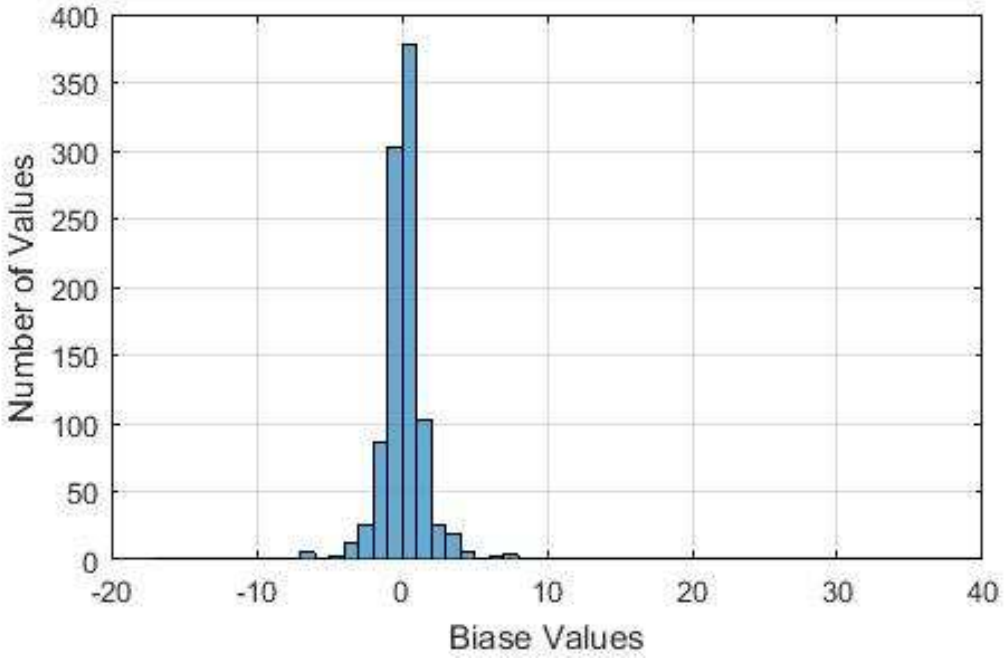


Figure (4.2): First Model - Obtained Biases with FNNPSO

Second Model: FNNPSO is used with different parameters for obtaining better results for training the system then testing it with four different testing datasets to overcome the mentioned problem previously with the first model. Training and testing results are presented in the following tables. Table (4.5) shows the used input features, hidden neurons, number of output class, Iteration number, and Training number as a second model for FNNPSO.

Table (4.5): Second Model – FNN Classifier Parameters

Input Features	Hidden Neurons	Output Class	Training NO.
29	38	2	800

Table (4.6) shows the used different parameters of the second model for PSO, which are used for getting better Weights and Bias values to provide as possible as minimum error for FNN in the training phase.

Table (4.6): Second Model- PSO Optimizer Parameters

NoP	Dim	IW	Wmax	Wmin	AC	Min Position	Max Position	Max_Itr	Momentum
98	2169	2	0.7	0.5	2	-25.9704	12.4725	100	0.8

Table (4.7) presented the confusion matrix for the second model in the training and testing phase. The table consisted of two parts A and B, where part A presents the result of the confusion matrix in the training phase with training dataset, and part B presents the confusion matrix in the testing phase with four different testing datasets with different number of instances.

Table (4.7): Second Model: Confusion Matrix of Train and Test Phase

Classified As	Yes	No
Yes	441	4
No	11	344

Part A: Confusion Matrix of Train Phase

Classified As	350 Test Dataset		400 Test Dataset		500 Test Dataset		600 Test Dataset	
	Yes	No	Yes	No	Yes	No	Yes	No
Yes	204	22	47	5	205	38	174	11
No	13	111	28	320	19	238	14	401

Part B: Confusion Matrix of Test Phase

By comparing Table (4.7) in the second model with the Table (4.3), obviously the difference between the results can be determined. In the second model, there is lower misclassification cases in comparison with the misclassifying cases in the first model. As it is clear in this table that correctly classified instances is larger and incorrectly classified instances is lower from the first model.

Table (4.8) presents the evaluation results of FNNPSO for the second model in both training and testing phases. In this table, the decline in the error rate is observed according to tested results, by comparing it with the results of the first model. It is clear there are increasing with the accuracy of the system and the percentage of correctly classified instances, with the decreasing in the percentage of incorrectly classified instances according to the comparison between these two models. This made the second model to deal with the test cases better than the first

model and to have the ability of classifying instances with less errors. As it was mentioned above with the Table (4.4) in the first model there are relation between the accuracy of the system and the error rate, here in this model can be noticed too.

Table (4.8): Second Model- Evaluation Results of FNNPSO (Training and Testing)

Parameter	Training	Testing			
	800	350	400	500	600
Accuracy	0.98125	0.90000	0.91750	0.88600	0.95833
CCI	785	315	367	442	575
CCI %	98.12%	90.00%	91.75 %	88.40 %	95.83 %
ICI	15	35	33	58	25
ICI %	01.87 %	10.00%	08.25 %	11.60 %	04.16 %
Sensitivity	0.99101	0.90265	0.90385	0.84362	0.94054
Fall-out	0.03098	0.10484	0.08046	0.07393	0.03373
Specificity	0.96901	0.89516	0.91954	0.92607	0.96627
Miss-rate	0.00898	0.09734	0.09615	0.15638	0.05945
Precision	0.97566	0.94009	0.62667	0.91518	0.92553
Recall	0.99101	0.90265	0.90385	0.84362	0.94054
F-measure	0.98327	0.92099	0.74016	0.87794	0.93298
MSE	0.03719	0.47226	0.43949	0.46592	0.32080
Elapsed Time	0.012772s	0.00044071s	0.00040871s	0.00051013s	0.00039302s

FNNPSO at the training phase in the learning of FNN obtained different values of weights and biases in the second model. These obtained values are illustrated in Figure (4.3) and Figure (4.4). In these figures larger domains are observed for the weights and biases than the domain of the figures (4.1) and (4.2) in the first model. This made the system to deal with the test datasets more softly that can deal with noises with getting good results.

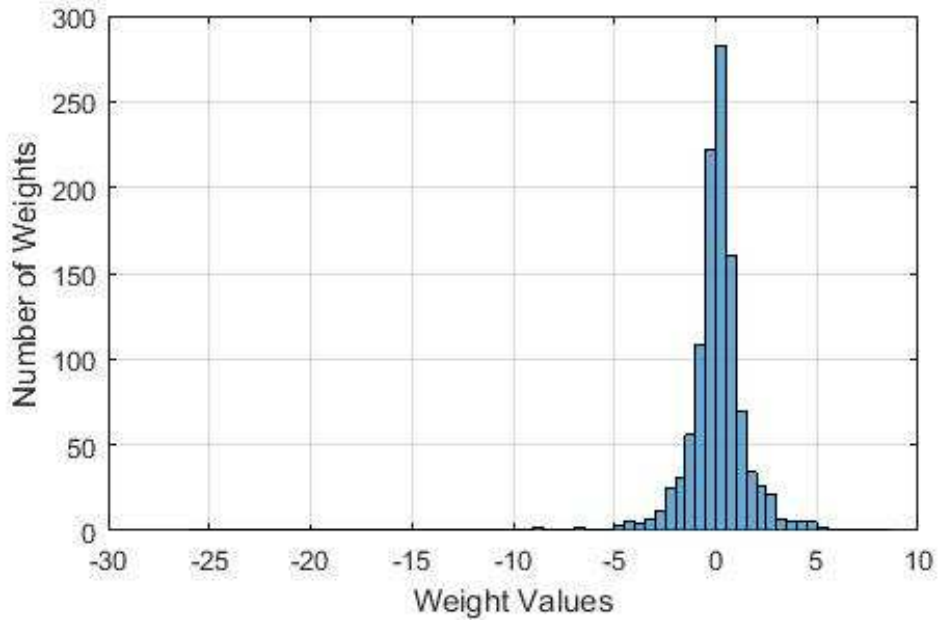


Figure (4.3): Second Model - Obtained Weights with FNNPSO

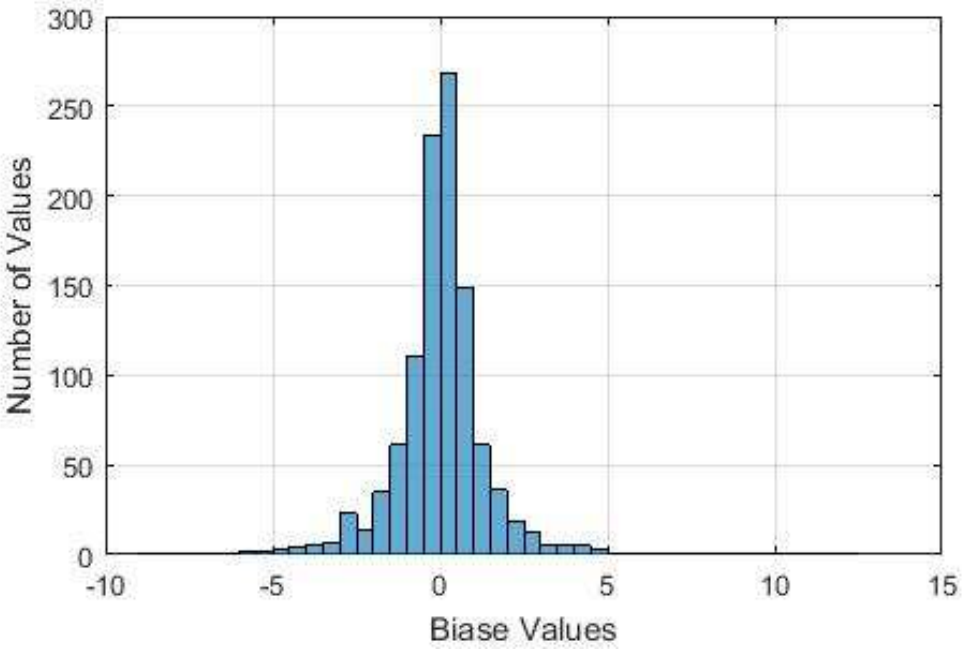


Figure (4.4): Second Model - Obtained Biases with FNNPSO

4.4 Experiment 2: Optimizing Forward Neural Network with PSO using Euclidian Distance

In this stage, a proposed system is implemented by using ED for improving PSO to obtain better weights and biases for FNN. The improved PSO attempts to evaluate the system as with reducing the least error rate as possible. For this purpose the training dataset with 800 instances is used in the training phase by depending on the 29 input features to train the system for determining the decision about the employee behavior and deciding if it deserves a recommendation for promotion or not as it is used in the first experiment.

The performance of the proposed system must be determined, for this reason it must be tested. So for testing, the improved system with four types of datasets are used as in the first experiment (350, 400, 500, and 600 instances). The results of the training and testing phases are presented as follows:-

Table (4.9), presents parameters and their values with FNN for implementing the proposed system.

Table (4.9): Proposed FNN Classifier Parameters

Input Features	Hidden Neurons	Output Class	Training NO.
29	37	2	800

Table (4.10), shows the used parameters for the improved PSO in the proposed system FNNPSOED.

Table (4.10): PSO Optimizer Parameters

NoP	Dim	IW	Wmax	Wmin	AC	Min Position	Max Position	Max_Itr	Momentum
98	2112	2	0.7	0.5	2	-0.6451	0.8259	100	0.8

Confusion matrix for the proposed model in the training and testing phases is presented in Table (4.11), which explains the performance of the system for classifying each case within the datasets. As within the first experiment this table consisted of two parts A and B too. Part A presents the results of the confusion matrix in the training phase with training dataset, and part B presents the confusion matrix in the testing phase. This proposed system provides the best result as it is clearly shown in the following table. Higher rate of correctly classified instances and the lower rate of incorrectly classified instances are noticed with this proposed system. The following tables can present these results.

Table (4.11): Confusion Matrix of Train and Test Phase

Classified As	Yes	No
Yes	349	9
No	4	438

Part A: Confusion Matrix of Train Phase

Classified As	350 Test Dataset		400 Test Dataset		500 Test Dataset		600 Test Dataset	
	Yes	No	Yes	No	Yes	No	Yes	No
Yes	176	0	252	0	218	1	511	1
No	0	174	2	146	4	277	6	82

Part B: Confusion Matrix of Test Phase

Table (4.12) presents the evaluation results of the proposed FNNPSOED in both training and testing phases. The lowest values of the error rate, fall-out, and miss-rate are noticed with the highest accuracy value. Subsequently, there are few

misclassification cases by comparing with the two models in experiment one which have higher error rates in each test cases. By comparing the elapsed time in these cases it can be notice that FNNPSOED finished with fewer time for one iteration as explained follow:

Table (4.12): Evaluation Results of FNNPSOED (Training and Testing)

Parameter	Training	Testing			
	800	350	400	500	600
Accuracy	0.98125	1.00000	0.99500	0.99000	0.98833
CCI	787	350	398	495	593
CCI %	98.37 %	100.0 %	99.50 %	99.00 %	98.83 %
ICI	13	0	2	5	7
ICI %	01.62 %	0.000 %	0.500 %	01.00 %	01.16 %
Sensitivity	0.99101	1.00000	1.00000	0.99543	0.99805
Fall-out	0.03098	0.00000	0.01351	0.01423	0.06818
Specificity	0.96901	1.00000	0.98649	0.98577	0.93182
Miss-rate	0.00898	0.00000	0.00000	0.00456	0.00195
Precision	0.97566	1.00000	0.99213	0.98198	0.98839
Recall	0.99101	1.00000	1.00000	0.99543	0.99805
F-measure	0.98327	1.00000	0.99605	0.98866	0.99320
MSE	0.03719	0.29554	0.27841	0.25488	0.26962
Elapsed Time	0.010586s	0.00041777s	0.00040147s	0.00041414s	0.00038517s

FNNPSOED at the training phase in the learning of FNN obtained different values of weights and biases. These obtained values are illustrated in Figure (4.5) and Figure (4.6). Here weights and biases are distributed in parallel and values are not grouped in one place, so dealing with noisy data was more softly with higher

results. This made the system to be robust and steady with any cases of the test data.

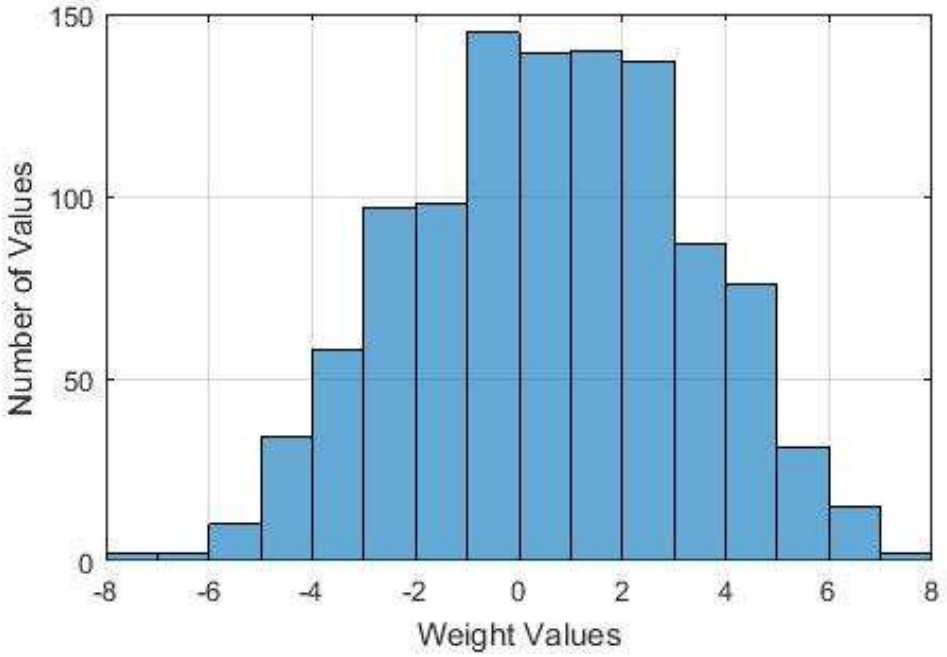


Figure (4.5): Obtained Weights with FNNPSOED

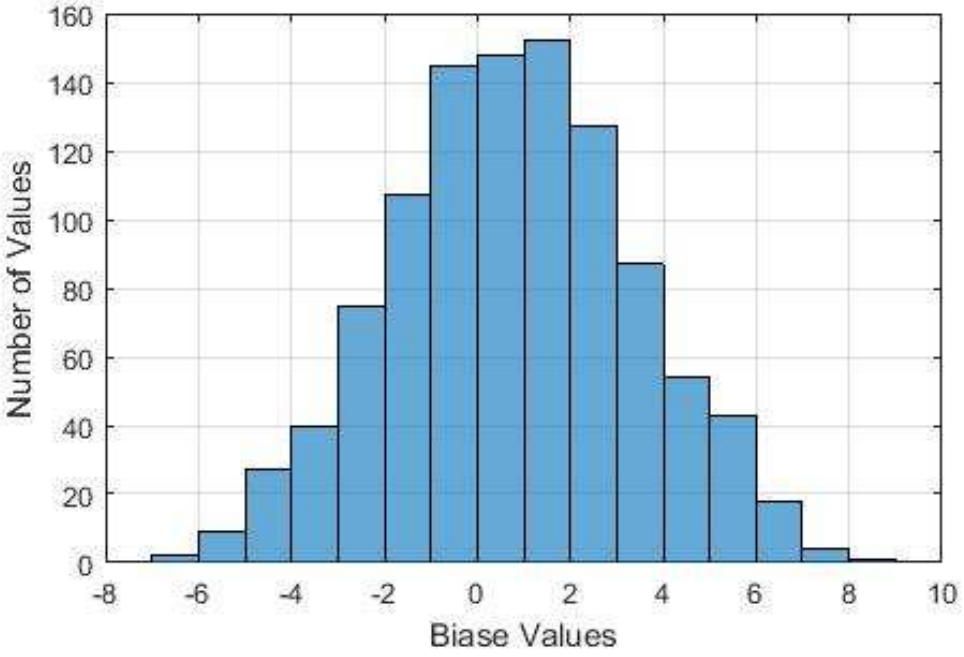


Figure (4.6): Obtained Biases with FNNPSOED

In each two experiment cases, error rates are decreased within each iteration that will help PSO with locating the best particle position that cause better results and reaching the goal for optimizing FNN Weights and Biases. From the Figure (4.7) noticed that the proposed system in the second experiment has the lowest error rate, this started with (0.4) and reached its minimum value with each iteration, where model1 and model 2 started with the higher points. Another important point that must be mentioned is that the proposed system reaches its lowest error rate value with lowest number of iterations unlike the standard two models. Here it can be said that the proposed system needs minimum iterations to reach the goal. The following figure illustrated a comparison of decreasing the error rate between each of the two experiment cases as follow:

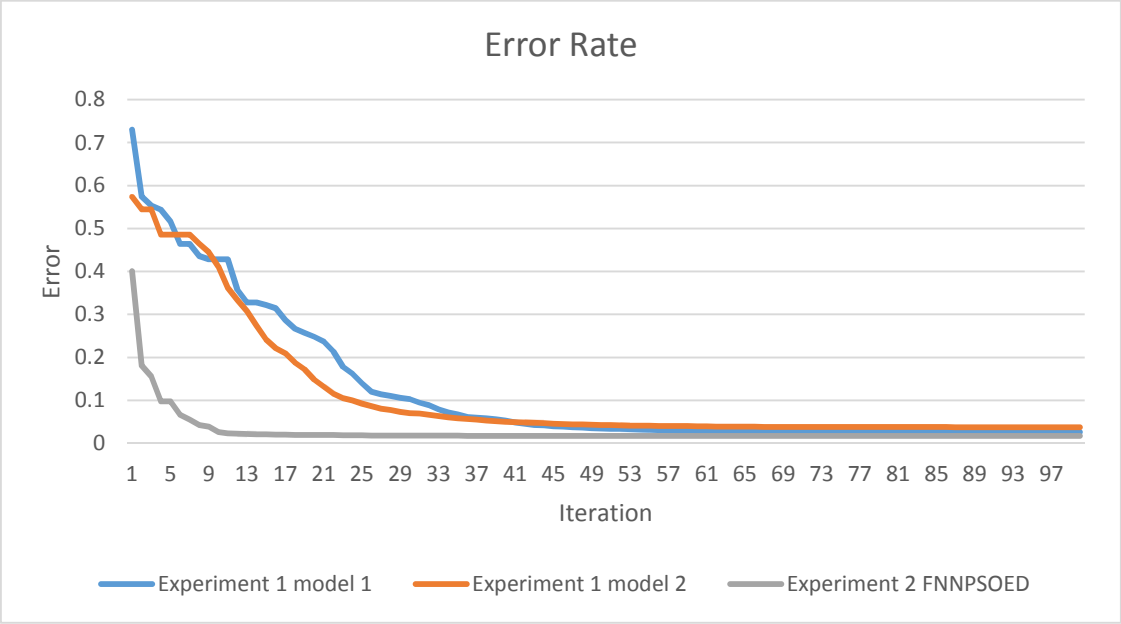


Figure (4.7): Error Rate of Three Case

4.5 Experiment 3: Classification using FRNN, NB, and DT Classifiers

In this experiment, three techniques of classification are used namely are FRNN, NB, and DT. The same datasets used with these techniques are those which were mentioned for the earlier experiments (1 and 2). Table (4.13) represents the used FRNN parameters which are input features, output (class), KNN, represents the number of the used neighbors for each point in the dataset, T-norm is used for the measures that are used in the upper approximation value (T(x,y) = min(x,y)), Implicator is used for the measures that are used in the lower approximation value (I(x,y) = max(1 - x, y)), and Similarity which is used to compose multiple relations (1 - abs(a(x) - a(y)) / abs(a_max - a_min)), where Tnorm is used for this purpose.

Table (4.13): FRNN Classifier Parameters

Input Features	Output	KNN	Tnorm	Implicator	Similarity
29	2	10	TnormKD	ImplicatorKD	Similarity1

Naïve Bayes classifier is used in this experiment as another classification technique for deciding on the employee behavior in the companies. Like other techniques a dataset with 800 instances is used with 29 input features and 2 output classes. The Naïve Bayes classifier evaluation results are presented in the following tables.

The last classifier technique in this experiment was the Decision Tree classifier. DT classifier parameters are presented in Table (4.14), like input features, output (class), confidence Factor (CF). The confidence factor is used for pruning (smaller values incur more pruning), minNO (minNumObj) represents the minimum number of instances per leaf, numFolds, determines the amount of data used for the reduced error pruning, one fold is used for pruning, the rest for growing the

tree, and Seed is used for randomizing the data, when the reduced error pruning is used.

Table (4.14): DT Classifier Parameters

Input Features	Output	CF	minNO	numFolds	Seed
29	2	0.25	2	3	1

The Confusion Matrices for the above mentioned classification techniques in this experiment are presented in Table (4.15) for each technique, there is a number of misclassification cases here in this experiment as follow:

Table (4.15): Confusion Matrices of FRNN, NB, and DT

Classified As	FRNN		NB		DT	
	Yes	**No**	**Yes**	**No**	**Yes**	**No**
Yes	301	0	302	0	293	9
No	11	488	105	393	10	488

The evaluation results for the three used techniques in this experiment are given in Table (4.16). This table represents the accuracy and the ability of the used techniques in classifying the collected dataset. Results was good with high accuracy and low misclassification with these three techniques. Beside of good classification results error rates can be noticed with each classification techniques in this experiment. These results are represented in the following table.

Table (4.16): Evaluation Results for FRNN, NB, and DT.

Parameter	FRNN	NB	DT
Accuracy	0.9862	0.8687	0.9750
CCI	790	696	781
CCI %	98.75%	87.00%	97.62%
ICI	10	104	21
ICI %	01.25%	13.00%	2.625%
Sensitivity	0.9850	0.8680	0.9740
Fall-out	0.0100	0.0800	0.0290
Specificity	0.9770	0.7900	0.9790
Miss-rate	0.0220	0.2090	0.0200
Precision	0.9850	0.9020	0.9740
Recall	0.9850	0.8680	0.9740
F-measure	0.9850	0.8700	0.9740
MAE	0.1024	0.1678	0.0309
RMSE	0.2157	0.3469	0.1531

4.6 Evaluation of Experimental Results

Eventually, in this section the classification rate of each case that is used in this thesis must be discussed and represented. The experimental results are represented in Table (4.17) that contained the classification rate for each case in the training and testing phases. In this table can be noticed that the accuracy in the Experiment 2 that FNNPSOED is used the highest accuracy rate in reached by comparing with other experiments, as follows:

Table (4.17): Classification Rate for Each Experimental Case

		Experiment 1		Experiment 2	Experiment 3		
	Data Sets	Model 1	Model 2	FNNPSOED	FRNN	NB	DT
Training	800 Train Dataset	**97.750 %**	**98.125 %**	**98.375 %**	**98.484 %**	**86.760 %**	**97.382 %**
Testing	350 Test Dataset	**88.285 %**	**90.000 %**	**100.00 %**			
	400 Test Dataset	**88.250 %**	**91.750 %**	**99.500 %**			
	500 Test Dataset	**84.000 %**	**88.600 %**	**99.000 %**			
	600 Test Dataset	**95.500 %**	**95.833 %**	**98.833 %**			

Figure (4.8) illustrated the accuracy and the performance of the proposed system in both training and testing phases for three experimental evaluation results as shown.

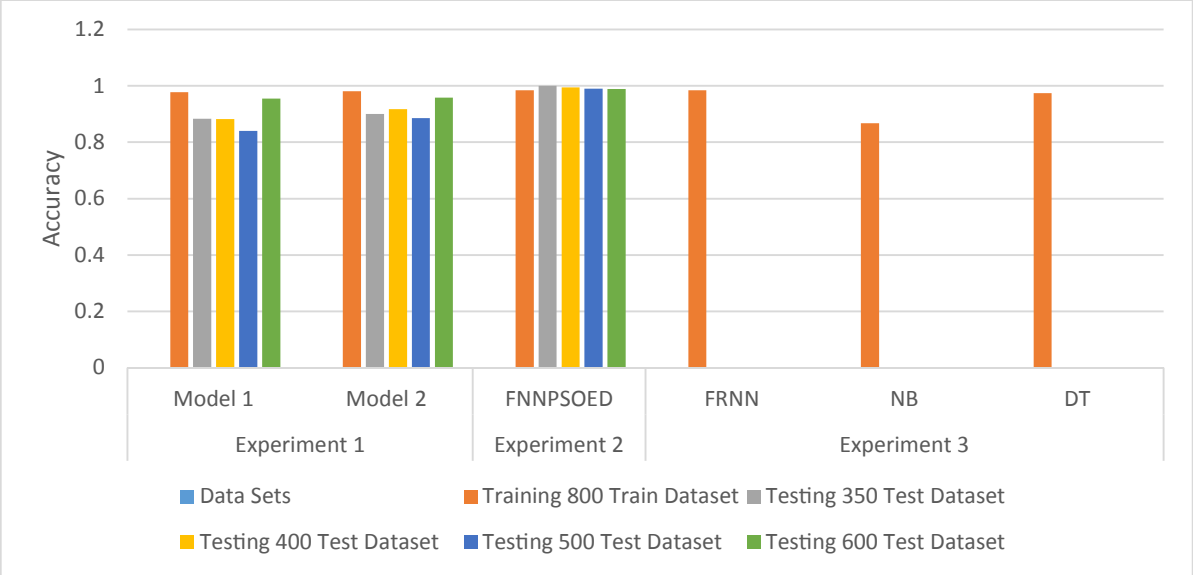


Figure (4.8): Classification Accuracy of Proposed System (Training and Testing)

In Figure (4.8) accuracy of each case and model visualized and it is clear that there is the difference in each case. Using ED has its effect on the system and its performance.

Chapter Five

Conclusion and Future Recommendation

Chapter Five

Conclusions and Future Recommendation

5.1 Conclusions

This thesis attempts to build a system for making decisions using intelligent techniques in data mining to replace the traditional ways in managing companies by the management and deciding for them to promote the employee or not, accept him\her or not, and may be used for turning over the employee or no. For the mentioned reason, the appropriate attributes for employees in the private and public sectors (companies) in Kurdistan Region were identified and depending on these attributes, the system is enabled to take the right decision about the employees in a proper manner, and to improve the quality and increase the income of the company.

In the previous chapters of this thesis, the proposed system for generating random number for PSO with FNN was presented and built. The effect of the generated random number and all the involved parameters of PSO and FNN were illustrated too. Several conclusion points for building this system to decide on human talents in private sectors have been concluded considering the obtained results from the proposed system, used techniques, and the collected dataset. These points are performed based on a series of classification experiments. Some of these conclusions are summarized as follows:-

1. Attribute value types (numeric, nominal, discrete … etc.) in the collected dataset have their effect on the accuracy result. Nominal and numeric were the appropriate types that are considered and used in this thesis.

2. Coding instance values in the dataset into float point number format and the class label coding to the format that was suitable with the feature values in both training and testing phases have increased the accuracy results with the used classification algorithms.
3. Using Natural Inspired Algorithm such as PSO with FNN that has one hidden layer is the main point for obtaining the best weights and biases for ANN by determining the best direction and the best position among particles according to the results (Table 4.4 and Table 4.8) in Chapter four.
4. Euclidian Probability Distribution is an important concluding point for implementing the proposed system to increase the accuracy of the system and decreasing the error rate. Table 4.12 in Chapter four represents the effect of using this algorithm with PSO.
5. The highest classification accuracy rate was with using (ED) algorithm for generating the random number within PSO which is used for learning NN and gaining weights and biases. Table 4.17 and Figure 4.10 represent the best accuracy result that is obtained with FNNPSOED.
6. It can be concluded from using the FNNPSOED, that the distribution of weights and biases were more uniform and steady compared to the obtained weights and biases with the using standard FNNPSOED (Figures 4.5 and 4.6). This made the proposed system to be more robust and have fixed error rate starting with the lower error value and reaching the minimal error value as is explained in the Figure 4.7 in chapter Four.

5.2 Future Recommendation

This study work can be extended into ways. Some points of ideas can be explained as follows:-

1. Using other multiclass applications and datasets instead of using only two classes in this thesis.
2. Using feature selection algorithms for selecting the best features may increase the accuracy and quality of the system.
3. Other techniques of natural inspired algorithms such as Grey Wolf, Cuckoo Search and Artificial Bee Colony can be used instead of PSO.
4. Other Probability Distribution techniques such as Manhattan and Gaussian can be used for generating random numbers with PSO.
5. Other types of ANNs such as Recurrent Neural Networks, Spiking Neural Networks and Deep Learning Neural Networks can be used instead of the used FNN which has one hidden layer. In other words, the number of the hidden layers can be changed and the memory of a network can be increased via adding more context layers as in Recurrent Neural networks.

Appendices

Appendix A

A1: FNNPSOED Training Phase Pseudo-Code

Input: Training Samples; Number of Input, Hidden and Output Neurons; Number of Particles; Max Iteration; Weights; Biases.
Output: Best Weights, Biases, and Accuracy of Train Samples
Part A

1. Read train samples with 800X30 matrix
2. Initialize FNN and PSO parameters
3. Create Position and Velocity vectors
4. **For each** Particles in the Swarm
5. Randomly initialize Velocities and Positions created in (step 3)
6. **End for**
7. Initialize ED counters
8. **For each** instance
9. Get the distance between each two attributes using (Eq. 3.1)
10. Standardize the distance obtained from (step 8) using (Eq. 3.3)
11. **End for**
12. **For each Iteration**
13. Initialize Weights with Position values from 1 to 1073 (step 5)
14. Initialize Biases with Position values from 1074 to end (step 5)
15. **For each train**
16. Calculate FNN with the random Weights and Biases from step (13, 14)
17. Calculate fitness function error= Mean Square Error
18. **If** pBest > fitness
19. Assign that Position to the pBest
20. **End if**
21. **If** gBest > fitness
22. Assign that Position to the gBest
23. **End if**
24. **If gBestScore==1 then Break;**
25. **End for**
26. **For each Particle**
27. Calculate new Velocities for each Particle using (Eq. 3.2)

28. Calculate new Positions for each Particle using new Velocities using (Eq. 2.16)
29. Iteration ++;
30. **End for**

Part B:

31. Assign Weights with the Particle best global Positions
32. Assign Biases with the Particle best global Positions
33. **For each training case**
34. Calculate FNN with new Weights and Biases
35. Determine number of correctly and incorrectly classified instances
36. Calculate classification rate and accuracy of the system
37. Calculate Confusion Matrix
38. **End for**

End of Pseudo Code

A2: Modified PSO Pseudo – Code

1. Read train samples with 800X30 matrix
2. Initialize PSO parameters
3. Create Position and Velocity vectors
4. **For each** Particles in the Swarm
5. Randomly initialize Velocities and Positions created in (step 3)
6. **End for**
7. Initialize ED counters
8. **For each** instance
9. Get the distance between each two attributes using (Eq. 3.1)
10. Standardize the distance obtained from (step 8) using (Eq. 3.3)
11. **End for**
12. Calculate fitness function error= Mean Square Error
13. **If** pBest > fitness
14. Assign that Position to the pBest
15. **End if**
16. **If** gBest > fitness
17. Assign that Position to the gBest
18. **End if**
19. **If gBestScore==1 then Break;**
20. **End for**
21. **For each Particle**

22. Calculate new Velocities for each Particle using (Eq. 3.2)
23. Calculate new Positions for each Particle using new Velocities using (Eq. 2.16)
24. Iteration ++;
25. **End for**

End of Pseudo Code

Appendix B

B1: Pseudo Code of Replace Missing Value Algorithm

Input: Training Samples
Output: Handling Missing Values

1. **For c = 1 to M**
2. Find mean value "Am" of all the attributes of the column „c"
3. Am(c) = (sum of all the elements of column c of d)/n
4. **End for**
5. **For r=1 to N**
6. **For c = 1 to M**
7. If D(N,M) is not a Number (missing value), then
8. Substitute Am(c) to D(N,M)
9. **End for**
10. **End for**

B2: Pseudo Code of SMOTE Algorithm

Input: Number of minority class samples T; Amount of SMOTE N%; Number of nearest neighbors k
Output: (N/100)* T synthetic minority class samples
1. (∗ If N is less than 100%, randomize the minority class samples as only a random percent of them will be SMOTEd. ∗)
2. **if** N < 100
3. **then** Randomize the T minority class samples
4. T = (N/100) ∗ T
5. N = 100
6. **endif**
7. N = (int)(N/100) (∗ The amount of SMOTE is assumed to be in integral multiples of 100. ∗)
8. k = Number of nearest neighbors
9. numattrs = Number of attributes
10. Sample[ ][ ]: array for original minority class samples
11. newindex: keeps a count of number of synthetic samples generated, initialized to 0.
12. Synthetic[ ][ ]: array for synthetic samples (∗ Compute k nearest neighbors for each minority class sample only. ∗)

13. **for** i ← 1 **to** T
14. Compute k nearest neighbors for i, and save the indices in the nnarray
15. Populate (N, i, nnarray)
16. **endfor** Populate(N, i, nnarray) (∗ Function to generate the synthetic samples∗)
17. **while** N 6≠0
18. Choose a random number between 1 and k, call it nn. This step chooses one of the k nearest neighbors of i.
19. **for** attr ← 1 **to** numattrs
20. Compute: dif = Sample[nnarray[nn]][attr] − Sample[i][attr]
21. Compute: gap = random number between 0 and 1
22. Synthetic[newindex][attr] = Sample[i][attr] + gap ∗ dif
23. **endfor**
24. newindex++
25. N = N − 1
26. **endwhile**
27. **return** (∗ End of Populate. ∗)
End of Pseudo-Code.

B3: Fuzzy Rough Nearest Neighbor Pseudo Code

Input: X, the training data; C, the set of decision classes; y, the object to be classified
Output: Classification for y

1. **Begin**
2. N⟵ getNearestNeighbours(y,K)
3. $\mu 1 \longleftarrow 0$, $\mu 2 \longleftarrow 0$, Class ⟵ ø
4. ∀C ∈ C do
a. if ((R↓C)(y) ≥ μ1(y) && (R↑C)(y) ≥ μ2(y)) then
5. Class⟵C
6. $\mu 1(y) \longleftarrow (R\downarrow C)(y)$, $\mu 2(y) \longleftarrow (R\uparrow C)(y)$
a. end
7. output Class
8. end

B4: Decision Tree Pseudo Code

INPUT: Training data
OUTPUT Decision tree
DTBUILD (*D)

```
1.  Begin
2.      T=φ;
3.      T= Create root node and label with splitting attribute;
4.      T= Add arc to root node for each split predicate and label;
5.      For each arc do
6.          D= Database created by applying splitting predicate to D;
7.          If stopping point reached for this path, then
8.          T'= create leaf node and label with appropriate class;
9.      Else
10.     T'= DTBUILD(D);
11.     T= add T' to arc;
12. End
```

References

[1] Lockwood N. R., SPHR G. M.A., (2006), "*Talent Management: Driver for Organizational Success*", SHRM® Research Quarterly, published by the Society for Human Resource Management, ISBN 1-932132-42-2, Vol. 51, No. 6, pp.1-11.

[2] Bloom N., & Van Reenen J., (2011), "*HUMAN RESOURCE MANAGEMENT AND PRODUCTIVITY*", NATIONAL BUREAU OF ECONOMIC RESEARCH, NBER Working Paper No. 16019, 4, pp.1697-1767.

[3] Chen Y. C., Wang W. C., & Chu Y. C., (2010), "*Structural Investigation of the Relationship between Working Satisfaction and Employee Turnover*", the Journal of Human Resource and Adult Learning Vol. 6, No. 1, p.41.

[4] Ivancevich J. M., Matteson M. T., & Konopaske R., (1990), "*Introduction to Organizational Behavior*", Bpi/Irwin, URL:https://www.wiziq.com/tutorial/136343-organizational-behavior Accessed: 22/3/2016.

[5] Waheed S., Zaim A., & Zaim H., (2012), "*TALENT MANAGEMENT IN FOUR STAGES*", the USV Annals of Economics and Public Administration, Vol. 12, Issue 1(15), pp.130-137.

[6] Jantan H. R., Hamdan A. A., & Othman Z., (2010), "*human talent Forecasting using data Mining Classification Techniques*", International Journal of Technology Diffusion, Vol. 1, No. 4, pp. 29-41.

[7] Jantan H., Hamdan A. R., & Othman Z. A., (2011), "*Data Mining Classification Techniques for Human Talent Forecasting*", Knowledge-Oriented Applications in Data Mining, INTECH Open Access Publisher, ISBN: 978-953-307-154-1.

[8] Jantan H., Hamdan A. R., & Othman Z. A., (2009), "*Knowledge Discovery Techniques for Talent Forecasting in Human Resource Application*", International Scholarly and Scientific Research & Innovation International Science Index, Vol. 3, No. 2, pp.579-587.

[9] Sexton R. S., McMurtrey S., Michalopoulos J. O., & Smith A. M., (2004), "*Employee turnover: a neural network solution*", Computers & Operations Research, vol. 32, No. 10, PP: 1-17 (col.fig. Nil), Elsevier Ltd. All rights reserved.

[10] Chang H. Y., (2009), "*Employee Turnover: A Novel Prediction Solution with Effective Feature Selection*", WSEAS International Conference. Proceedings. Mathematics and Computers in Science and Engineering, Issue 3, Vol. 6, ISSN: 1790-0832, PP: 417-426.

[11] Jantan H., Hamdan A. R., & Othman Z. A., (2010), "*Human Talent Prediction in HRM using C4.5 Classification Algorithm*", International Journal on Computer Science and Engineering, Vol. 02, No. 08, ISSN : 0975-3397, pp. 2526-2534.

[12] Jantan H., Hamdan A. R., & Othman Z. A., (2011), "*Towards applying Data Mining Techniques for Talent Mangement*", International Conference on Computer Engineering and Applications, IPCSIT vol.2, pp: 476 - 481.

[13] Al-Radaideh Q. A., & Al Nagi E., (2012), "*Using Data Mining Techniques to Build a Classification Model for Predicting Employees Performance*" International Journal of Advanced Computer Science and Applications, Vol. 3, No. 2, pp: 144-151.

[14] Florence. T. M. & Savithri R., (2013), "*TALENT KNOWLEDGE ACQUISITION USING C4.5 CLASSIFICATION ALGORITHM*", International Journal of Emerging Technologies in Computational and Applied Sciences, Vol. 4, No. 4, pp. 406-410.

[15] Jantawan B., & Tsai C. F., (2013), "*The Application of Data Mining to Build Classification Model for Predicting Graduate Employment*", International Journal of Computer Science and Information Security, Vol. 11, No. 10.

[16] Tamizharasi K., & Rani R. U., (2014), "*Employee Turnover Analysis with Application of Data Mining Methods*", International Journal of Computer Science and Information Technologies (IJCSIT), Vol. 5 (1), ISSN: 0975-9646, pp: 562-566.

[17] Seidu Y., (2011) "*Human resource management and Organizational performance*", PhD Thesis, Aston University, Birmingham.

[18] Allen M. R., (2006), "*STRATEGIC HUMAN RESOURCE MANAGEMENT AND FIRM*", PhD Thesis, Faculty of the Graduate School, Cornell University.

[19] Cheng Z., & Chen Y., (2012), "*Data Mining Applications in Human Resources Management System*", Journal of Convergence Information Technology (JCIT), Vol. 7, No. 8, issue 8.30, pp: 262 – 271.

[20] Azar A., Sebt M. V., Ahmadi P., & Rajaeian A., (2013), "*A model for personnel selection with a data mining approach: A case study in a commercial bank*", SA Journal of Human Resource Management, Vol.11, No.1, pp:10-pages.

[21] Long L. K., & Troutt M. D., (2003), "*Data Mining for Human Resource Information Systems*", "*Data Mining: Opportunities and Challenges",* edited by John Wang.

[22] Laudon K. C., & Laudon J. P., (2009), "*Management Information Systems*", Tenth Edition, New Delhi-110001.

[23] Aggarwal N., & Kapoor M., (2012), "*Human Resource Information Systems (HRIS) - Its role and importance in Business Competitiveness*", Gian Jyoti E-Journal, Vol. 1, Issue 2, ISSN 2250-348X.

[24] Arora K., (2013), "*Importance of HRIS: A Critical Study on Service Sector*", Global Journal of Management and Business Studies, ISSN 2248-9878, Vol. 3, No. 9, pp. 971-976.

[25] Shaikh M. S., (2012), **"***HUMAN RESOURCE INFORMATION SYSTEM (HRIS) DESIGNING NEEDS FOR BUSINESS APPLICATION*", ZENITH International Journal of Business Economics & Management Research, ISSN 2249-8826, Vol.2, Issue 1.

[26] Zaïane O. R., (1999), "*Principles of Knowledge Discovery in Databases*", Department of Computing Science, University of Alberta.

[27] Maimon O., & Rokach L. (Eds.), (2010), "*Data Mining and Knowledge Discovery Handbook*", Springer New York Dordrecht Heidelberg London, ISBN 978-0-387-09822-7, e-ISBN 978-0-387-09823-4, Vol. 2, New York: Springer.

[28] Cios K. J., Swiniarski R. W., Pedrycz W., & Kurgan L. A., (2007), "*A Knowledge Discovery Approach*", In Data Mining, ISBN: 978-0-378-33333-5, pp. 9-24, Springer US.

[29] Zaki M. J., & Wong L., (2003), "*DATA MINING TECHNIQUES*", WSPC/Lecture Notes Series: 9in x 6in, p.2.

[30] Han J., & Kamber M., (2006), "*Data Mining: Concepts and Techniques, 2nd edition*", ISBN 1-55860-901-6.

[31] Rygielski C., Wang J. C., & Yen D. C., (2002), "*Data mining techniques for customer relationship management*", Technology in society Vol. 24, No. 4, pp.483-502.

[32] Tseng S. M., Wang K. H., & Lee C. I., (2003), "*A pre-processing method to deal with missing values by integrating clustering and regression techniques*", Applied Artificial Intelligence Vol. 17, No. 5-6, pp:535-544.

[33] Kaiser J., (2014), "*Dealing with Missing Values in Data*", Journal of Systems Integration Vol. 5, No. 1, pp: 42-51.

[34] Poolsawad N., Kambhampati C., & Cleland J. G. F., (2014), "*Balancing Class for Performance of Classification with a Clinical Dataset*", Proceedings of the World Congress on Engineering, London, U.K, Vol. 1.

[35] Bhardwaj B. K., & Pal S., (2011),"*Data Mining: A prediction for performance improvement using classification*", (IJCSIS) International Journal of Computer Science and Information Security, Vol. 9, No. 4.

[36] Macro L., (2007), "*Data Mining Techniques for Prediction and Classification in Discrete Data Applications*", Ph.D. Thesis, Department of Operations Research, University of Colorado.

[37] Singh Y., & Chauhan A. S., (2009), "*Neural networks in data mining*", Journal of Theoretical and Applied Information Technology, Vol. *5*, No. 6, pp.36-42.

[38] Turban E., Sharda R., Aronson J. E., & King D., (2008) "*Business intelligence: A managerial approach"*. Upper Saddle River: Pearson Prentice Hall. ISBN 9780132347617 – 013234761x.

[39] Tewary G., (2015), "*EFFECTIVE DATA MINING FOR PROPER MINING CLASSIFICATION USING NEURAL NETWORKS*", International Journal of Data Mining & Knowledge Management Process (IJDKP), Vol.5, No.2.

[40] Rashid T., (2006), "*A Novel Recurrent Neural Network Model: A Case Study in Energy Load Forecasting"*, PhD. Thesis, College of Engineering, Mathematical and Physical Sciences. National University of Ireland, Dublin.

[41] Verbiest N., (2014), "*Fuzzy rough and evolutionary approaches to instance selection*", PhD thesis, Ghent University.

[42] Jabar A. L., & Rashid T. A., (2015),"*Combining Fuzzy Rough Set with Salient Features for HRM Classification*", Computer and Information Technology; Ubiquitous Computing and Communications; Dependable, Autonomic and Secure Computing; Pervasive Intelligence and Computing (CIT/IUCC/DASC/PICOM), pp. 244-251.

[43] Jensen R., & Cornelis C., (2011), "*Fuzzy-rough nearest neighbour classification and prediction*", Theoretical Computer Science, Vol. 412, No.42, pp.5871-5884.

[44] Patil T. R., & Sherekar S. S., (2013), "*Performance analysis of Naive Bayes and J48 classification algorithm for data classification*", International Journal of Computer Science and Applications; Vol. 6, No. 2, pp:256-261.

[45] Novaković J., Štrbac P., & Bulatović D., (2011), "*Toward optimal feature selection using ranking methods and classification algorithms*", Yugoslav Journal of Operations Research ISSN: 0354-0243 EISSN: 2334-6043, Vol. 21, No. 1.

[46] Somasundaram P., & Muthuselvan N. B., (2010), "*A Modified Particle Swarm Optimization Technique for Solving Transient Stability Constrained Optimal Power Flow*" Journal of Theoretical and Applied Information Technology, Vol.19, No. 8, pp. 970-989.

[47] Fan H., (2002), "*A modification to particle swarm optimization algorithm*", *Engineering Computations*, Vol. *19, No.* 8, pp: 970-989.

[48] Li J., Ding L., & Li B., (2014), "*A Novel Naive Bayes Classification Algorithm Based on Particle Swarm Optimization*" Open Automation and Control Systems Journal, Vol. 6, pp: 747-753.

[49] Mu A. Q., Cao D. X., & Wang X. H., (2009), "*A modified particle swarm optimization algorithm*" Natural Science, Vol.1, No.2, pp: 151-155.

[50] Leskovec J., Rajaraman A., & Ullman J. D., (2014), "*Mining of massive datasets"*. Cambridge University Press.

[51] Fawcett, T. (2006). “An introduction to ROC analysis”, *Pattern recognition letters*, *27*(8), 861-874.

[52] Chai, T., & Draxler, R. R. (2014). “Root mean square error (RMSE) or mean absolute error (MAE)?–Arguments against avoiding RMSE in the literature”, *Geoscientific Model Development*, *7*(3), 1247-1250.

[53] Shcherbakov, M. V., Brebels, A., Shcherbakova, N. L., Tyukov, A. P., Janovsky, T. A., & Kamaev, V. A. E. (2013). “A survey of forecast error measures”, *World Appl Sci J*, *24*, 171-176.

[54] Varshavsky, V., Marakhovsky, V., & Levin, I. (2005). “CMOS fuzzification circuits for linear membership functions”. *WSEAS Transactions on Systems*, *4*(4), 238-243.

List of Publication

1. Asia L. Jabar and Tarik A. Rashid, “Combining Fuzzy Rough Set with Silent Features for HRM Classification”, Computer and Information Technology; Ubiquitous Computing and Communications; Dependable, Autonomic and Secure Computing; Pervasive Intelligence and Computing (CIT/IUCC/DASC/PICOM), 2015 IEEE International Conference on, INSPEC. Accession Number: 15681346, IEEE, IEEE Xplore: 28 December 2015.
2. Tarik A. Rashid, Asia L. Jabar, “Improvement on predicting employee “behaviour through intelligent techniques”. Source: IET Networks, pp. 7. Dol: 10.1049/iet-net. 2015-0106, Online ISSN 2047-4962 Available online: indexed by Elsevier (Scopus) SJR, IF=0.16, 2016.
3. Tarik A. Rashid and Asia L. Jabar, “A new modified Particle Swarm Optimization with neural network for classifying Employee Behaviour”. Submitted to Neural Computing and Applications, indexed by ISI, IF=1.5, in the process.

الخلاصة

في وضع التحول الاقتصادي اليوم حول العالم، ظهرت العديد من التحديات و القضايا مع ظهور هذا التحول. احدى التحديات التي ظهرت مع التزايد في استراتيجية المنظمات في سوق العمل كان في مجال ادارة الموارد البشرية. حيث يعتبر ادارة الموارد البشرية من احد اهم الاجزاء في تكوين المنظمات.

وهناك العديد من التحديات و القضايا في هذا المجال مثل المناولة اليدوية لمعلومات الموظفين و اتخاذ القرارات على السلوك البشري، عملية تحديد الموهبة الموجودة بين جميع العاملين في المؤسسة، تأييد الموظف للترقية التي تستحقها، قبول موظف جديد اذا لزم الامر، منع موظف ممتاز من مغادرة المنظمة التي قد يؤثر على اداءها في سوق العمل، استخدام الاوراق اثناء العمل ايضا يعتبر من احدى القضايا المعقدة في الشركة. كل هذه المشاكل تؤدي الى خطأ بشري و اهدار الوقت للمنظمة. لذلك، يوجد اهتمام متزايد في ادارة الموارد البشرية للشركات و ذلك بسبب تأثيرها على واردات هذه الشركات.

في هذه الاطروحة، تم استخدام منهج المسح لجمع البيانات من شركات الاتصالات المختلفة، و شركات البناء في اقليم كوردستان. تم معالجة هذه البيانات من خلال اتباع عدد من الخطوات و استخدام التقنيات المختلفة للتعامل مع القيم المفقودة في البيانات التي تم جمعها، و من ثم تحقيق التوازن بين عينات الموجودة في مجموعة البيانات التي تم جمعها. ثم، في مرحلة التصنيف، استخدمت تقنيات فعالة لتنفيذ النظام المقترح. و استخدمت Forward Neural Network ، Fuzzy Rough Nearest Neighbor، Naïve Bayes ، Decision Tree. وبالاضافة الى ذلك، تم تطوير نموذج جديد يسمى FNNPSOED عن طريق استخدام تقنية الامثل المفيدة مثل Particle Swarm Optimization. يتم استخدام Particle Swarm Optimization لتحقيق الاستفادة المثلى من الأوزان و الانحيازات في Forward Neural Network. و ايضا استخدمت طريقة Euclidean Distance، لتحسين Particle Swarm Optimization. أنتج النموذج الجديد نتائج عالية الجودة من حيث الدقة و السرعة و الخطأ. أنتجت FNNPSOED أفضل النتائج مع استخدام أنواع من بيانات الأختبار و الذي يحتوي على (350، 400، 500، 600 حالات)، و كانت النتائج ٪100.00، ٪99.500، ٪99.00، ٪98.833 على التوالي.

تطوير تقنية شبكات العصبية باستخدام الجسيمات السرب الأمثل

رسالة

مقدمة الى مجلس كلية العلوم

في جامعة السليمانية كجزء من متطلبات

نيل شهادة ماجستير في علوم

الكومبيوتر

من قبل

أسـيا لطيف جبار

بكالوريوس علوم الكومبيوتر (2010)، جامعة كركوك

باشراف

أ.د.طارق احمد رشيد

نوفمبر 2016 **تشرين الثاني 1438**

پوختە

لە ئەنجامی گۆڕانکاری بەرەو پێش چوونی ئابووری لە بازاری کار لە ئەمڕۆی جیهاندا،زۆرێك لە ئاستەنگ و کێشە سەریان هەڵداوە لەم ماوەی بەرەوپێش چوون گۆڕانکاریەدا. هاوکات لەگەڵ بەرەوپێش چوون و پەرەسەندنی ستراتیجی دامەزراوەکان لەبازاری کاردا، یەکێك لەو ئاستەنگانەی کە پەیدابوو لە بەشی بەڕێوە بردنی سەرچاوە مرۆییەکان بوو. لەکاتێکدا بەڕێوە بردن و پەرەپێدانی سەرچاوە مرۆییەکان بە یەکێك لە گرنگترین بەشەکانی دامەزراوەکان دادەنرێت

ژمارەیەك کێشە و ئاستەنگ لەبەشی بەڕێوە بردنی سەرچاوە مرۆییەکان بەدی دەکرێت، لەوانەش تۆمارکردنی زانیاری کارمەندەکانی کۆمپانیا لە ڕێگەی دەست، بڕیاردان لەسەر ڕەفتاری مرۆیی کارمەندەکان، پڕۆسەی دەست نیشان کردنی بەهرە تایبەتمەندەکانی نێوان کارمەندانی کۆمپانیا، پێشنیاز کردنی خەڵات بۆ کارمەندێك کە شایستەی بێت، پەسەندکردنی کارمەندی نوێ بۆ کۆمپانیا ئەگەر پێویست بێت، ڕێگە گرتن لە کارمەندی بەهرەدار لە بەجێهێشتنی کۆمپانیا کە کاریگەری لەسەر کارا بوونی کۆمپانیا دەبێت لە بازاڕی کاردا، هەروەها هەڵگرتنی زانیاری لەسەر شێوەی لاپەڕە و کاغەز کارێکی سەخت و جەنجالە. بۆیە پێویستە گرنگی زیاتر بدرێت بە بواری بەڕێوەبردنی سەرچاەی مرۆییەکانی کۆمپانیا و کاریگەرییەکانی لەسەر داهاتی ئابووری کۆمپانیا.

بۆ ئەم مەبەستەش بڕیار درا بە ئەنجام دانی ئەم تویژینەوەیە، وە ڕێگەی خەملاندن بەکارهێنرا بۆ مەبەستی کۆکردنەوەی داتا لە ژمارەیەك کۆمپانیا لە هەرێمی کوردستان. دواتر ئەم داتا کۆکراوەیە ئامادەکرا لەڕێگەی بەکارهێنانی چەند تەکنیکێك بۆ نەهێشتنی داتا ون بووەکان و باڵانس کردنی داتا بۆ بەکارهێنانی وەك سەرچاوەیەکی سەرەکی بۆ باشتر کردنی پۆلێن کردنی کەیسەکان بە بەکارهێنانی ئەو تەکنیکانەی کەلەژێر چاودێری و ڕێگە پێشاندراون بۆ بنیات نانی سیستمی نوێ. بۆ ئەمەش Forward Neural Network, Fuzzy Rough Nearest Neighbor, Decision Tree, Naïve Bayes بەکارهێنرا. وە بۆ دروست کردنی FNNPSOED وەکو مۆدێلێکی نوێ بەبەکارهێنانی تەکنیکی Particle Swarm Optimization بەسود وەرگرتن لە یاسای Euclidean Distance. لە پاش ئامادەکردن و بنیات نانی سیستمی نوێ ئەنجامی باشتر بەدەست هات بە ڕێژەی باشتر لەکاتی تاقی کردنەوە بە بەکارهێنانی چوار داتای جیاواز بەم شێوەیە (350، 400، 500، 600) حاڵەت، لە ئەنجام دا بەدەست هێنانی ئەم ئەنجامانە (100%، 99.5%، 99%، 98.83%).

بەرەو پێش بردنی تەکنیکی نەرمەبژێر بە بەکارهێنانی ئاپۆرەی تەنۆچکەکان

نامەیەك پێشکەش کراوە

بە ئەنجومەنی کۆلێجی زانست لە زانکۆی سلێمانی

وەك بەشێك لە پێداویستیەکانی بەدەست هێنانی بڕوانامەی

ماستەر لە زانستی کۆمپیوتەر

لەلایەن

أســـیا لطیـــف جـــبار

بەکالۆرێۆس لە زانستی کۆمپیوتەر (2010)، زانکۆی کەرکوك

بە سەرپەرشتی

پ.د. طارق احمد رشید

سەرماوەز 2716 **تشرینی دووەم 2016**