

ANKARA YILDIRIM BEYAZIT UNIVERSITY

GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES



**A NEW FRAMEWORK BY USING DEEP LEARNING
TECHNIQUES FOR DATA PROCESSING**

Ph.D Thesis By

Ahmad Mozaffer Karim

Department of Computer Engineering

December 2018

ANKARA

A NEW FRAMEWORK BY USING DEEP LEARNING TECHNIQUES FOR DATA PROCESSING

A Thesis Submitted to

The Graduate School of Natural and Applied Sciences of

Ankara Yıldırım Beyazıt University

**In Partial Fulfillment of the Requirements for the Degree Doctor of Philosophy
in Computer Engineering, Department of Computer Engineering**

By

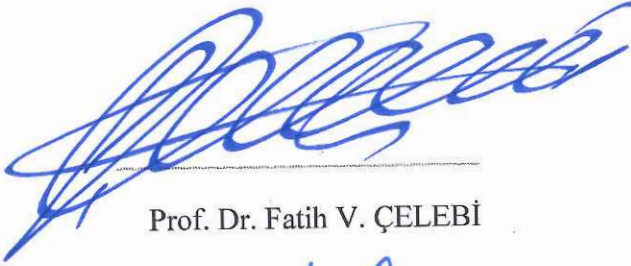
AHMAD MOZAFFER KARIM

December 2018

ANKARA

PH.D. THESIS EXAMINATION RESULT FORM

We have read the thesis entitled “A New Framework by Using Deep Learning Techniques for Data Processing” completed by **Ahmad Mozaffer Karim** under the supervision of **Prof. Dr. Fatih V. ÇELEBİ** and we certify that in our opinion it is fully adequate, in scope and in quality, as a thesis for the degree of Doctor of Philosophy.



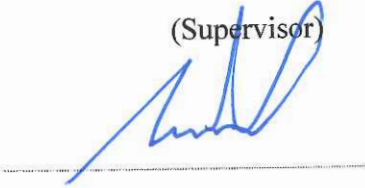
Prof. Dr. Fatih V. ÇELEBİ

(Supervisor)



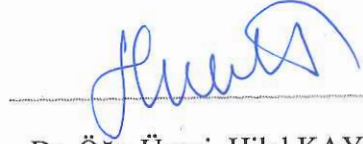
Prof. Dr. Mehmet R. TOLUN

(Co-Supervisor)



Doç. Dr. Mehmet S. GÜZEL

(Jury member)



Dr. Öğr. Üyesi. Hilal KAYA

(Jury member)



Dr. Öğr. Üyesi. Özkan KILIÇ

(Jury member)



Doç. Dr. Gazi E. BOSTANCI

(Jury member)

Prof. Dr. Ergün ERASLAN

Director

Graduate School of Natural and Applied Sciences

ETHICAL DECLARATION

I hereby declare that, in this thesis which has been prepared in accordance with the Thesis Writing Manual of Graduate School of Natural and Applied Sciences,

- All data, information and documents are obtained in the framework of academic and ethical rules,
- All information, documents and assessments are presented in accordance with scientific ethics and morals,
- All the materials that have been utilized are fully cited and referenced,
- No change has been made on the utilized materials,
- All the works presented are original,

and in any contrary case of above statements, I accept to renounce all my legal rights.

Date:10/12/2018

Signature:.....

Name & Surname: Ahmad Mozaffer Karim

ACKNOWLEDGEMENTS

I would like to gratefully acknowledge various people who have contributed in some way to accomplish this thesis.

First and foremost, there are no proper words to convey my deep gratitude and respect for my supervisor and the great mentor, Prof. Dr. Fatih Çelebi. The door to Prof. Çelebi's office was always open whenever I ran into a trouble spot or had a question about my research or writing. He consistently allowed this paper to be my own work, but he steered me in the right direction whenever he thought I needed it.

My sincere thanks must also go to the members of my thesis advisory and exam committee: Prof. Dr. Mehmet Tolun, Dr. Hilal Kaya and Dr. Mustafa Yeniad. They generously gave their time to offer me valuable comments toward improving my work, especially Prof. Dr. Şahin Emrah, who has inspired me to become an independent researcher. Without him, this thesis would not have been completed or written.

I would like to express my very great appreciation for all people in the Graduate School of Natural Sciences at Ankara Yıldırım Beyazıt Üniversitesi for their support, especially for my instructors throughout the academic courses which were a part of my Ph.D. dissertation, namely Prof. Dr. Abdullah Çavuşoğlu, Prof. Dr. Ünal Çamdali and Dr. Öğr. Üyesi Lami Kaya.

I owe my deepest gratitude to my father, Mr. Modahfar Mohammed and my mother Sumia Omar, for their wonderful services during my study years.

I would like to thank all my friends, those who expressed their good wishes and encouragement throughout my studies.

2018, 10 December

Ahmad Mozaffer Karim

A NEW FRAMEWORK BY USING DEEP LEARNING TECHNIQUES FOR DATA PROCESSING

ABSTRACT

Deep auto-encoder neural networks have been widely used in several image classification and recognition problems, including handwriting recognition, medical imaging, face recognition, etc. The overall performance of deep auto-encoder neural networks mainly depends on the number of parameters used, structure of neural networks and the compatibility of the transfer functions. However, an inappropriate structure design can cause a reduction in the performance of deep auto-encoder neural networks.

Four frameworks are proposed to evaluate the performance of the auto-encoder which is one of the common used deep learning techniques. In the first framework, the parameters of each auto-encoders were optimized by using Taguchi method. The proposed framework was validated by using four datasets; DDOS detection, IDS recognition, Epileptic seizure recognition and Digit classification datasets. In the second framework, pre-processing technique of energy spectral density was used to extract important features from input data. The proposed framework was tested by using three medical datasets. In the third framework, deep auto-encoder was combined with Discrete Wavelet Transform (DWT) to enhance its performance. Then, framework produced satisfactory results when compared to well-known studies in this field.

Finally, a linear model was proposed as a post-processing technique to enhance the output of deep auto-encoder which its parameters were estimated by using Particle Swarm Optimization Algorithm (PSO). The experimental results show that the proposed method presented high accuracy when compared with previous studies.

Keywords: Deep learning; energy spectral density, taguchi method; auto-encoder, particle swarm optimization algorithm; linear model.

A NEW FRAMEWORK BY USING DEEP LEARNING TECHNIQUES FOR DATA PROCESSING

ÖZ

Derin otomatik kodlayıcı sinir ağları el yazısı tanıma, medikal görüntüleme, yüz tanıma vb. de dahil olmak üzere çeşitli sınıflandırma ve tanıma sorunlarında yaygın olarak kullanılmaktadır. Derin otomatik kodlayıcı sinir ağlarının genel olarak performansı kullanılan parametrelerin sayısına, sinir ağlarının yapısına ve transfer fonksiyonlarının uyumluluğuna bağlıdır. Bununla birlikte, uygun olmayan yapı tasarımı, derin otomatik kodlayıcı sinir ağlarının performansında bir düşüşe neden olabilmektedir.

Yaygın olarak kullanılan derin öğrenme tekniklerinden birisi olan otomatik kodlayıcı performansını değerlendirmek için dört çerçeve önerilmiştir. Birinci çerçevede, her bir otomatik kodlayıcıya ait parametre Taguchi yöntemi kullanılarak optimize edilmiştir. Önerilen çerçeve DDOS (Dağıtık Hizmet Aksatma) tespiti, IDS (Kimlik) tanıma, Epileptik nöbet tanıma ve basamak sınıflandırma veri seti olmak üzere dört veri seti kullanılarak doğrulanmıştır. İkinci çerçevede, girdi verilerinden önemli özellikleri çıkarmak için enerji spektral yoğunluğunun ön işleme tekniği kullanılmıştır. Önerilen çerçeve üç medikal veri seti kullanılarak test edilmiştir. Üçüncü çerçevede, derin otomatik kodlayıcı performansını artırmak için Kesikli Dalgacık Dönüşümü (DWT) ile birleştirilmiştir. Daha sonra, çerçeve bu alanda iyi bilinen çalışmalara kıyaslanmış ve tatmin edici sonuçlar ürettiği görülmüştür.

Son olarak, derin otomatik kodlayıcının çıktısını artırmak için işlem sonrası teknik olarak doğrusal (lineer) bir model önerilmiş ve parametreleri Parçacık Sürü Optimizasyonu (PSO) Algoritması kullanılarak tahmin edilmiştir. Deneysel sonuçlar, önerilen yöntemin önceki çalışmalarla karşılaştırıldığında yüksek doğruluk sağladığını göstermiştir.

Anahtar Kelimeler: Derin öğrenme; enerji spektral yoğunluğu, taguchi yöntemi; otomatik kodlayıcı, parçacık sürü optimizasyonu; doğrusal model.

CONTENTS

PH.D. THESIS EXAMINATION RESULT FORM	ii
ETHICAL DECLARATION	iii
ACKNOWLEDGEMENTS	iv
ABSTRACT.....	v
ÖZ.....	vi
CONTENTS.....	vii
NOMENCLATURE.....	x
LIST OF TABLES	xi
LIST OF FIGURES	xiv
1 INTRODUCTION	1
1.1 Problem Statement	2
1.2 The Goal of This Study	3
1.3 Datasets	3
1.4 Thesis Organization.....	4
2 BACKGROUND AND RELATED WORKS.....	5
2.1 Unsupervised Learning.....	5
2.1.1 Factor analysis.....	6
2.1.2 Principal Components Analysis (PCA).....	7
2.1.3 The K-means Clustering Algorithm.....	7
2.2 Supervised Learning.....	8
2.2.1 Neural Network.....	9

2.2.2	Support Vector Machine	10
2.2.3	Random Forest (RF).....	12
2.3	Taguchi Method	15
2.4	Particle Swarm Optimization Algorithm (PSO).....	18
2.5	Literature Review	20
3	MATERIALS AND METHODS.....	31
3.1	Deep Learning	31
3.1.1	Stacked Sparse Auto-encoder	33
3.1.2	Convolutional Neural Network.....	33
3.1.3	Recurrent neural network (RNN).....	38
3.2	Proposed Frameworks	41
3.2.1	Deep Auto-encoder Based on Taguchi Method.....	43
3.2.2	Deep Auto-encoder based Energy Spectral Density	46
3.2.3	Deep Auto-encoder Based on Discrete Wavelet Transform.....	49
3.2.4	Deep Auto-encoder Optimized with Linear Model Based on PSO Discrete Wavelet Transform.....	52
4	EXPERIMENTAL RESULTS AND DISCUSSIONS	56
4.1	Experimental Results by Using Sparse Auto-encoder and Taguchi Method Framework.....	57
4.1.1	DDOS Detection using the proposed framework	57
4.1.2	IDS Attack.....	65
4.1.3	Epileptic Seizure Recognition.....	68

4.1.4	Handwritten Digit Classification.....	71
4.2	Experimental Results by Using Using Deep Auto-encoder and Energy Spectral Density	74
4.2.1	Epileptic Seizure Detection.....	74
4.2.2	SPECTF Classification	80
4.2.3	Diagnosis of Cardiac Arrhythmias.....	86
4.3	Experimental Results by Using Deep Auto-encoder and DWT.....	93
4.4	Experimental Results Using Novel Framework Combining the Deep Auto-Encoder and a Linear Model Based on PSO	95
4.4.1	Epileptic Seizure Detection.....	95
4.4.2	SPECTF Classification	101
4.4.3	Diagnosis of Cardiac Arrhythmias.....	108
4.4.4	Digit Classification	116
5	CONCLUSION AND FUTURE WORK	119
5.1	Conclusion.....	119
5.2	Future Work	121
	REFERENCES.....	122
	CURRICULUM VITAE.....	139

NOMENCLATURE

Acronyms

ML	Machine learning
DL	Deep Learning
NN	Neural Network
ANN	Artificial Neural Network
SVM	Support Vector Machine
PCA	Principal Components Analysis
RF	Random Forest
PSO	Particle Swarm Optimization
SSAEs	Stacked Sparse Auto-encoders
CNN	Convolutional Neural Network
RNN	Recurrent Neural Network
LM	Linear Model
AI	Artificial Intelligence
DDoS	Distributed Denial-Of-Service
IDS	Intrusion Detection System

LIST OF TABLES

Table 2.1: Orthogonal array selection table.	15
Table 4.1: Auto-encoder 1 upper and lower level values of factors for ddos detection.	58
Table 4.2: Auto-encoder (1) upper and lower level values of factors for ddos attack detection.	59
Table 4.3: Auto-encoder 1 parameters values obtained by using taguchi method for ddos attack detection.	59
Table 4.4: Auto-encoder 1 s/n ratios obtained in the taguchi experimental design for ddos attack detection.	60
Table 4.5: Auto-encoder 1 parameters set obtained through optimization for ddos detection.	61
Table 4.6: Auto-encoder 2 upper and lower level values of factors for ddos detection.	61
Table 4.7: Auto-encoder 2 upper and lower level values of factors for ddos detection.	62
Table 4.8: Auto-encoder 2 parameters values obtained by using taguchi method for ddos attack detection.	62
Table 4.9: Auto-encoder 2S/N ratios obtained in the taguchi experimental design for ddos attack detection	63
Table 4.10: Auto-encoder 2 parameters set obtained through optimization for ddos detection	64
Table 4.11: Ddos detection methods results comparison.	64
Table 4.12: Auto-encoder 1 parameters set obtained through optimization for ids detection.	66

Table 4.13: Auto-encoder 2 parameters set obtained through optimization for ids detection.	67
Table 4.14: A comparison amongst ids detection methods results.	67
Table 4.15: Auto encoder 1 parameters set obtained through optimization for epileptic seizure recognition.	69
Table 4.16: Auto-encoder 2 parameters set obtained through optimization for epileptic seizure recognition.	70
Table 4.17: A Comparison of epileptic seizure recognition results.	70
Table 4.18: Auto-encoder 1 parameters set obtained through optimization for handwritten digit classification.	72
Table 4.19: Auto-encoder 2 parameters set obtained through optimization for handwritten digit classification.	72
Table 4.20: A comparison of handwritten digit classification results.	73
Table 4.21: Auto-encoders factors values for epileptic seizure detection.	77
Table 4.22: Epilepsy seizure detection results.	78
Table 4.23: Comparisons of epilepsy detection results.	79
Table 4.24: Auto-encoders factors values for spectf classification.	83
Table 4.25: Spectf classification results.	84
Table 4.26: Comparisons of spectf classification results.	85
Table 4.27: Auto-encoders factors values for diagnosis of cardiac arrhythmias.	89
Table 4.28: Diagnosis of cardiac arrhythmias results.	90
Table 4.29: Comparison of diagnosis of cardiac arrhythmias results with previous studies.	91
Table 4.30: Auto-encoders parameters of deep auto-encoder and a linear model based on pso.	96
Table 4.31: Pso parameters for epilepsy seizure.	97
Table 4.32: Epilepsy seizure detection results using post processing.	99

Table 4.33: Comparison proposed framework with previous studies.....	100
Table 4.34: Auto-encoders factors values for spectf classification using post processing.....	104
Table 4.35: Pso parameters for spectf.	105
Table 4.36: Spectf classification results using post processing.	106
Table 4.37: Comparisons of spectf classification results using post processing.....	107
Table 4.38: Auto-encoders factors values for diagnosis of cardiac arrhythmias using post processing.	111
Table 4.39: Pso parameters for cardiac arrhythmias.	112
Table 4.40: Diagnosis of cardiac arrhythmias results.	113
Table 4.41: Comparison of diagnosis of cardiac arrhythmias results.	114
Table 4.42: Digit classification using post processing.	117
Table 4.43: A comparison of handwritten digit classification results.....	118

LIST OF FIGURES

Figure 2.1: Unsupervised learning block diagram	6
Figure 2.2: 3D to 2D using PCA [9].	7
Figure 2.3: Finding three clusters using K-means algorithm [11].	8
Figure 2.4: Supervised learning block diagram	9
Figure 2.5: Simple neural network.....	10
Figure 2.6: SVM classifier [16].	11
Figure 2.7: SVM regression [16].	11
Figure 2.8: Random forest [17].....	13
Figure 2.9: Machine learning structure.	14
Figure 2.10: Taguchi method flowchart.....	17
Figure 2.11: Simple pso flowchart [29].	19
Figure 3.1: Deep learning structure [91].....	32
Figure 3.2: Convolutional neural network [95].	34
Figure 3.3: Convolution layer	35
Figure 3.4: Features map of convolution layer	36
Figure 3.5: Max pooling.....	37
Figure 3.6: Average pooling	37
Figure 3.7: Fully connected layer	38
Figure 3.8: Recurrent neural network structures [105].	40
Figure 3.9: LSTM [106].....	41
Figure 3.10: Deep learning framework combining sparse auto-encoder and taguchi method.....	45
Figure 3.11: The proposed framework using SSAE architecture for training and ESD for feature estimation	48

Figure 3.12: DWT process.....	50
Figure 3.13: Deep auto-encoder based on DWT.....	51
Figure 3.14: The deep learning framework based on a linear model and pso.	52
Figure 3.15: The deep learning framework based on a linear model and pso training and testing flowchart.	55
Figure 4.1: Auto-encoder 1 main effect of experimental parameters on the S/N ratio for DDOS	60
Figure 4.2: The main effect of experimental parameters on the S/N ratio for auto-encoder 2 for ddos.....	63
Figure 4.3: Confusion matrix for ddos detection results.....	65
Figure 4.4: Confusion matrix for ids detection results.....	68
Figure 4.5: Confusion matrix for epileptic seizure recognition results.....	71
Figure 4.6: Confusion matrix for handwritten digit classification.....	73
Figure 4.7: An instance (patient data) from epilepsy dataset.....	75
Figure 4.8: Features extracted by using energy spectral density from an ‘instance of epilepsy dataset.	76
Figure 4.9: An instance (patient data) from spectf dataset.	81
Figure 4.10: Features extracted via esd from spectf dataset.	82
Figure 4.11: An instance (patient data) from cardiac arrhythmias dataset.	87
Figure 4.12: Features extracted via esd from cardiac arrhythmias dataset	88
Figure 4.13: Confussion matrix of deep auto-encoder based dwt.....	94
Figure 4.14: Datasets for normal and abnormal cases.	96
Figure 4.15: The mse for the linear system.....	98
Figure 4.16: An instance (patient data) from spectf dataset.....	101
Figure 4.17: Spectf dataset features in auto-encoder 1.	102
Figure 4.18: Spectf dataset features in auto-encoder 2.	103

Figure 4.19: An instance (patient data) from cardiac arrhythmias dataset.	109
Figure 4.20: Cardiac Arrhythmias Data in Auto-encoder 1.....	110
Figure 4.21: Cardiac arrhythmias data in auto-encoder 2.....	110

CHAPTER 1

INTRODUCTION

Machine learning (ML) is a popular branch of artificial intelligence (AI) that does not need to be explicitly programmed but allows machines to obtain new skills and predict results with high accuracy. Deep learning (DL) is a new version of ML which has recently been applied in many fields from computer vision to high dimensional data processing. Essentially, DL achieves great improvement in solving problems that has resisted the trials of the AI society for more than three decades and also achieved many state-of-the-art results [1,2]. It should be noted that DL can predict comprehensive outcomes by requiring little engineering, which cannot be compared by the conventional AI based approaches. DL will be applied to different fields in the near future due to its flexible and generic structure. Development of innovative learning algorithms and new structures for deep neural networks will merely speed up this progress [3]. Recently, Deep Auto-encoders have shown state-of-the-art achievement on different machine learning tasks which relies on unsupervised learning algorithms [4]. Deep auto-encoders have been widely used in different fields from image recognition to computer network etc. DL is presently earning a lot of heed for its use with big healthcare data. Even though ANN was introduced in 1950, there were severe limitations in its application to solve real dilemmas, due to vanishing gradient and overfitting problems, which hindered the training in deep architecture, lack of computing power, and primarily the absence of sufficient data to train the computer system. Deep learning has number of techniques which used in different medical application such as convolutional neural network, deep auto-encoder and deep belief neural network etc. These techniques presented satisfactory results when compared to previous traditional methods because of having ability to learn high level features from input data. Furthermore, deep learning techniques provide speed up processing when they deal with huge data when compared to traditional techniques including support vector machine, neural network and Naive Bayes classifier.

In this thesis, three frameworks were proposed by using deep learning techniques for data classification. In the first framework, the auto-encoders were optimized by using Taguchi method. The Taguchi method was used to find the best combination of parameters and obtained the best accuracy. The proposed framework was validated by using four datasets of digit recognition, epilepsy detection, DDOS detection and IDS detection. The experimental results compared to best proposed methods in this fields and showed that our proposed method gives better results than previous studies.

In the second framework, an effective framework which can automatically classify different types of data by using deep learning techniques based on energy spectral density (ESD) was introduced. ESD is used in the feature extraction level. In the next level, two sparse auto-encoders were used to reduce the dimension of extracted features by ESD and learn sensitive features. The acquired high level and sensitive features were classified by using SoftMax classifier.

In the third framework, deep auto-encoder was combined with DWT to enhance its performance. The A5 was calculated to reduce the dimension of the features and extracted important features. Then, framework produced satisfactory results when compared to well-known studies in this field.

Finally, the last framework consisted of two stages as deep auto-encoder and post processing step. In the first stage, the Deep auto-encoder was used to classify input data and the obtained accuracy was optimized by using a linear model as post processing model. Furthermore, the parameters of models were estimated by using Particle Swarm Optimization (PSO) algorithms. Moreover, the proposed framework was validated by using epileptic seizure datasets and presented satisfactory results when compared with previous studies.

1.1 Problem Statement

In this thesis, the problem is generally focused on developing frameworks using deep learning techniques with different other methods such as: ESD, Taguchi method and post processing techniques to optimize the performance of deep auto-encoder in data classification. The main problem of deep learning techniques is that they are not fast

in numbers of problems or do not present high accuracy. Therefore, the deep auto-encoder is combined with different techniques to improve its performance. Generally, the problem can be studied in number of points:

- How to optimize the structure of deep auto-encoder that is used in data classification?
- How to speed up and increase the performance of deep auto-encoder?
- How to increase the performance of deep auto-encoder by using pre and post processing techniques?

1.2 The Goal of This Study

- To propose a novel framework for optimizing the structure of auto-encoder by using the Taguchi Method.
- To use ESD with deep auto-encoder to improve deep auto-encoder performance and validated the framework with number of medical datasets.
- To use deep auto-encoder based on Discrete wavelet transform to decrease computational time and increase performance of the system.
- Propose a novel framework by using post processing with deep auto-encoder to improve the performance of deep auto-encoder.

1.3 Datasets

Information retrieval is one of the most common problems in computer science. Generally, dataset is an important part of any research study and exactly in our study because number of frameworks were proposed and these frameworks must be validated by using common used classification datasets in different fields.

In the first framework, four datasets were used digit recognition, epilepsy detection, DDOS detection and IDS detection.

In the second and second frameworks, there datasets were used for epilepsy recognition, Heart disease classification and arrhythmia classification.

In the last framework, epilepsy seizure dataset that proposed by Bonn university was used to validate the proposed method.

1.4 Thesis Organization

This thesis involves five chapters as follows:

- **Chapter One:** Introduction: Problem statement, The Goal of the thesis, dataset.
- **Chapter Two:** Literature review: this chapter provides an overview of the related works in deep learning studies.
- **Chapter Three:** Methodology: this chapter provides an outline of the research methodology which used in this thesis. Deep learning, Deep Auto-encoder.
- **Chapter Four:** The implementation details of experiment and the results that were obtained for all the proposed scenarios and comparison of the results.
- **Chapter Five:** Conclusion and future works.

CHAPTER 2

BACKGROUND AND RELATED WORKS

Generally machine learning consists of three different techniques as unsupervised, semi-supervised and supervised learning methods. In this chapter, supervised and unsupervised learning methods are explained with its types and examples. Furthermore, several studies related to deep learning, DDOS detection, IDS detection, Epilepsy recognition, Digit classification, SPECTF Classification and Cardiac Arrhythmias are presented in details.

2.1 Unsupervised Learning

Assume a model which receives input of sequence features x_1, x_2, x_3, \dots , where x_t is the sensitive input at time t . This input features, commonly known as data, match with an image on the retina, the pixels in a camera, or a video waveform. Furthermore, in unsupervised learning there is only feature (X) as input and no corresponding label variables' information. The output of the unsupervised model is also features with low dimension that were extracted from input features.

Hence, this technique known as unsupervised because of there is not educator and target output. The main advantage of unsupervised learning is reducing the input data size and reconstruct it in the form which increases the classification accuracy [5, 6]. Generally the unsupervised learning is used to reduce features dimension and extracted, selected effective features as shown in Figure 2.1.

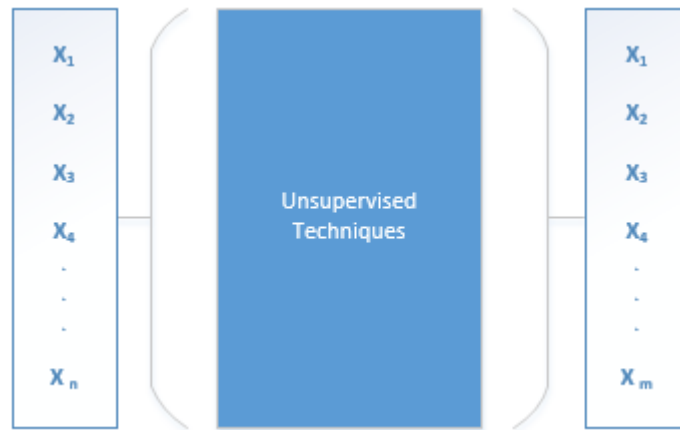


Figure 2.1: Unsupervised learning block diagram

Where both X_n and X_m represented features in different values and dimensions. The unsupervised techniques are usually used before supervised learning techniques to reduce the computational time by reducing the dimension of features and optimize the outcome of classification by performing new clusterings.

In this section, numbers of unsupervised learning methods are presented:

2.1.1 Factor analysis

D represents the dataset which consists of D -dimension real values $D = \{y_1, y_2, y_3, \dots\}$. The input data is created by using the Equation.

$$y = \Lambda x + \varepsilon \quad (2.1)$$

where x is a k -dimensional zero-mean Gaussian vector with features match to hidden factors, Λ is matrix of parameters in the form of $D \times K$. The details of the mathematical model are presented in [7].

2.1.2 Principal Components Analysis (PCA)

Principal Component Analysis (PCA) is perhaps the most common multivariate statistical method and is used as unsupervised learning method in different studies by researchers. Moreover, it is also the oldest multivariate method [8]. PCA is generally presented in problems which have big multi-dimensional data. The high dimension and number of features, lead to increase the computation time and more system memory is required. However, PCA tries to decrease the dimension of features and keeping the specific variance amount between features, while a practical reply is still possible.

In PCA, the basic idea concerns on choosing L dimension features from M dimension features which is the input data as ($L < M$). Which means transferring M dimensional features to the L dimension. M chose in the form that M features become optimum descriptive to the original input data M . as shown in Figure 2.2 the translation of the data from 3D to 2D is presented [9].

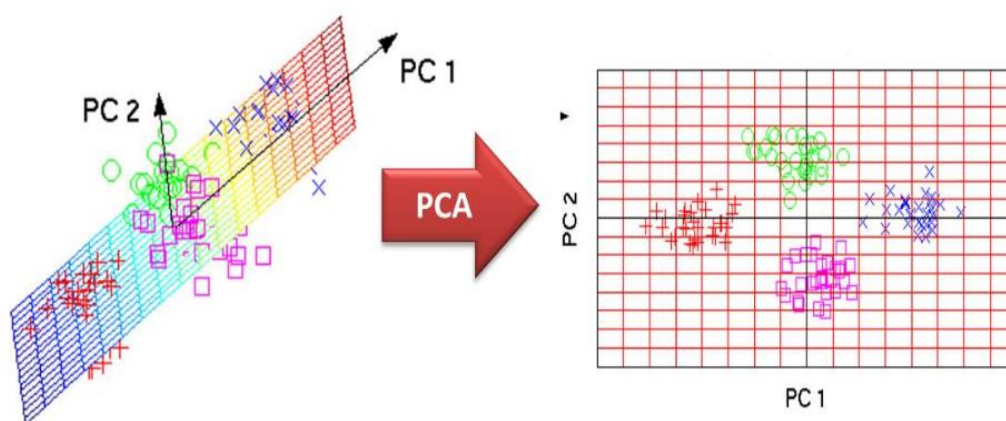


Figure 2.2: 3D to 2D using PCA [9].

2.1.3 The K-means Clustering Algorithm

K-means is a method of clustering features into an exact number of separate clusters. The "K" represented the number of the clusters determined. Several distance measures occur to select which feature will be added to which cluster. The purpose of the

algorithm is to minimize the measure between the center of the cluster and the given feature by iteratively adding a feature to any cluster and dismiss when the lowermost distance measure is realized. Therefore, the sample data is originally separated into K clusters and the observations are arbitrarily allotted to the clusters. Briefly, the technique can be represented in the pseudo code as shown below [10, 11].

Simple K-mean algorithm

- 1: Select K point as initial centroids.
 - 2: **repeat**
 - 3: From K clusters by assigning each point to its closest centroid.
 - 4: Re-compute the centroid of each cluster
 - 5: **Until** Centroid do not change
-

When above simple K-mean algorithm executed the input data will be arranged as Figure 2.3 depended on the number of iterations.

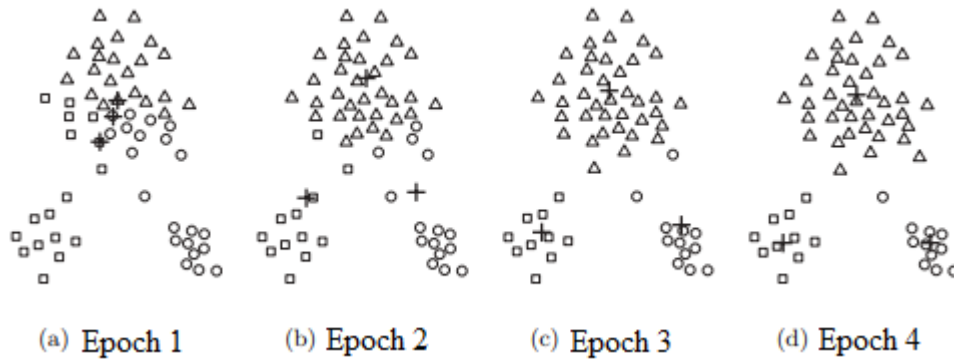


Figure 2.3: Finding three clusters using K-means algorithm [11].

2.2 Supervised Learning

Supervised technique is an automatic learning method which tries to yield rules from input features. Therefore, the aim of this method is to learn how to classify features to the labels that are prepared by experts by using the equation in (2.2) [12]. The purpose

is to use this equation to determine a compact representation of the function of prediction, that a new input x associates an output $g(x)$.

$$g(x) = f(x) \quad (2.2)$$

The input features were divided into two sets as training and testing sets. In training stage, data tries to learn rules and differences between labels. Then, in the testing stage, the learned model is used to predicate the output using new data. The block diagram of Unsupervised learning presented in Figure 2.4.



Figure 2.4: Supervised learning block diagram

Where X_n represents the input features that is classified by supervised learning techniques to the labels X_m and the m, n refer to the dimension of the features and the labels. The supervised learning is presented in a number of techniques in this section:

2.2.1 Neural Network

Artificial Neural Network (ANN), commonly identified as a neural network (NN), is essentially a mathematical model inspired by human nervous systems comparable with brain process information. NN includes an interconnected sets of mannered neurons and it uses connectionist method to process data for computation. NN works such as an adaptive system, which modificates its organization in learning stage. NN can easily models the simple and also complex relationships. An ANN can be designed for a

particular application, such as data classification, recognition, estimation and regression [13, 14].

There are several types of neural network configurations. There are several neural network types such as: Single layer feed-forward network, Multilayer feed-forward network and Single node with its own feedback. Simple neural network structure is shown in Figure 2.5.

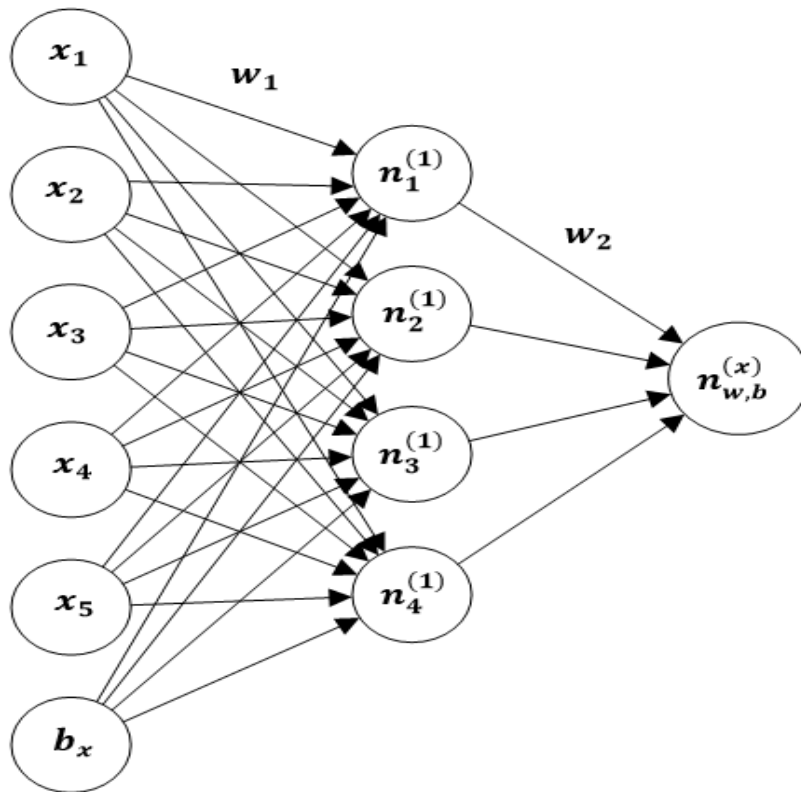


Figure 2.5: Simple neural network

2.2.2 Support Vector Machine

SVMs were initially proposed for data classification problems by Vapnik and co-workers, and they have been commonly applied to several fields for face recognition, signal processing, classification and regression [15]. The main feature of SVM is that they influence the organizational risk minimization standard to catch a decision function with a decent generalization capability. The exact subset of the training is feature points which are called support vectors controlled the answer to a particular problem. As illustration in Figure 2.6, the SVM builds a maximum margin hyperplane

which differs from linear classifiers that tries to find nearest hyperplane as shown in Figure 2.7. According to the results, SVM can produce satisfactory, unique and global optimal solution compared to various linear classifiers [16].

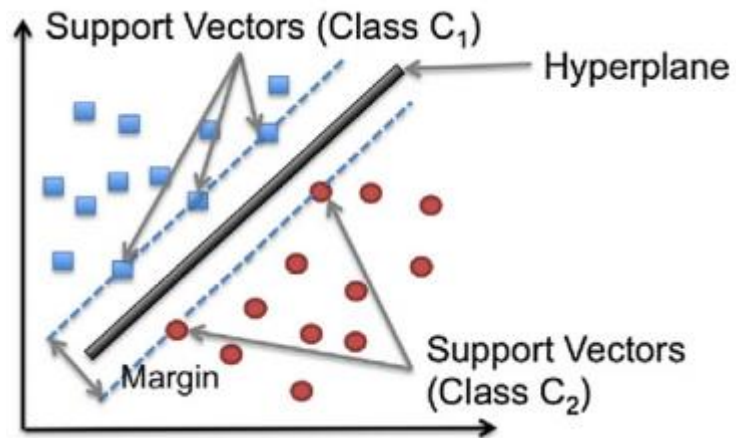


Figure 2.6: SVM classifier [16].

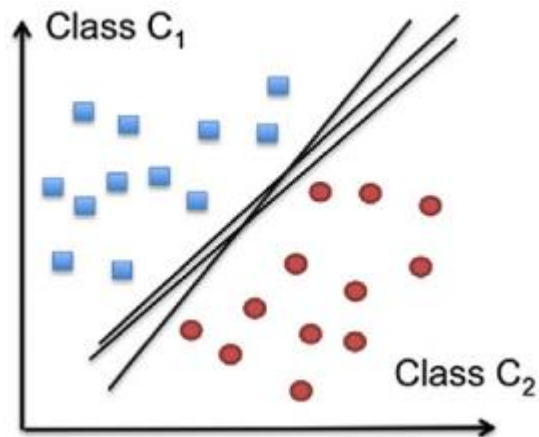


Figure 2.7: SVM regression [16].

2.2.3 Random Forest (RF)

RF is an influential learning algorithm, which mixes number of randomized decision trees and collections of their estimations by averaging. In recent years, the RF is one of the common successful overall purpose techniques.

In this part, we exist a label ranking technique depended on random forest system, that was indicated as Label ranking Random forest (LR-RF). The presented LR-RF works in two levels. In the first level, it builds several decision trees by using various training examples at building level, and then at the estimating level, query example permits over all trees, a two-step status combination approach is applied to collect the adjacent grades into a final estimated status. Figure 2.8 shows the entire procedure from a query instance x to finally get the estimated status $\hat{\pi}$ [17].

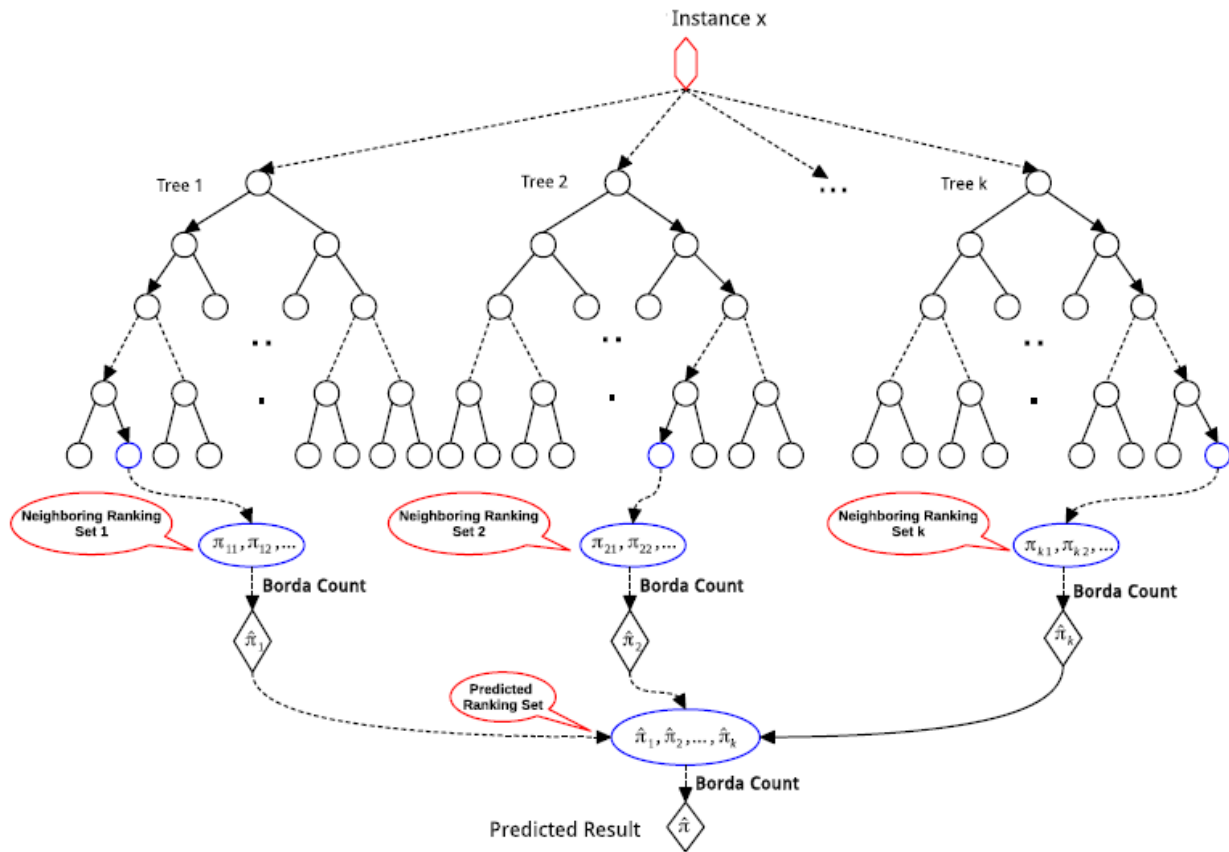


Figure 2.8: Random forest [17].

Finally, the machine learning represented important and well studied topic of artificial intelligence area which is applied in many interested fields such as: Computer vision, disease diagnosis, video classification etc.. In Figure 2.9 the race of machine learning are presented in detailed form.



Figure 2.9: Machine learning structure.

2.3 Taguchi Method

Taguchi method is a statistical robust design method that was first proposed by Genichi Taguchi to improve the quality of manufactured product and also more recently applied to a variety of fields from engineering to marketing [19, 20]. Three concepts were considered by the Taguchi concepts, namely Taguchi loss Function, Offline quality control and Orthogonal arrays for experimental design. Taguchi Method offers a methodology for designing the experiments. For instance, if an experiment is aimed on heating of wire by passing the electricity through it, then different control parameters from material type to diameter of wire are considered. Those parameters may have various values. DOE allows you to obtain the parameters and their values in an efficient manner. An example orthogonal selection table is illustrated in Table 2.1.

Table 2.1: Orthogonal array selection table.

		Number of Parameters (NoP)							
Number of Levels		2	3	4	5	6	7	8	9
	2	L4	L4	L8	L8	L12	L12	L12	L16
	3	L9	L9	L9	L18	L18	L18	L27	L27
	4	L16	L16	L16	L16	L32	L32	L32	L32
	5	L25	L25	L25	L25	L25	L50	L50	L50

Essentially those arrays tend to adopt a methodical way to permute and combine the collaboration among different parameters. Besides, unlike the full factorial experiment, there is no need to carry out each experiment respectively. To obtain the objective value or best accuracy, Taguchi method decreases the number of necessary experiments by using orthogonal arrays (OA). This reduces the number of experiments to be performed and also reduces the overall cost. These arrays are essentially predefined matrices, including control parameters and number of experiments. The purpose of the Taguchi method is to design an experiment that reduces the effect of the operator that cannot be controlled with a least amount of experiments [21, 22]. The

selection of an appropriate orthogonal array is mainly based on the number of control parameters and corresponding levels. Orthogonal arrays vary from L4 to L50 (see Table 2.1). More numbers of control parameters yield the higher values after “L”. Design of experiments are performed by employing the defined orthogonal array [21]. The iterations of experiments can be performed once the OA is carefully chosen. The number of iterations are then confirmed based on the complexity of the experiments. As aforementioned, the purpose of Taguchi method to design an experiment that reduces the effect of the operator that cannot be controlled with a least amount of experiments [21]. Taguchi method is a powerful technique for supplying the best set among different stages of various parameters. The measure used in Taguchi method is Signal-to-noise (S/N) ratio to measure and esteem the superiority features is the ratio of signal (S) to the operator of noise (N). Various S/N ratios were presented but, three of them are considered standard [23]. The first standard is “Smaller-is-better”, when the objective account of the quality variable η is zero. In this case, the S/N ratio can be defined as Eq. (2.3):

$$\eta = -10 \log \sum \frac{x^2}{k} \quad (2.3)$$

In equation 1, x is the account of the experimental control and k is the number of experiments. The second standard is “Larger-is-better” when, the zero account of the quality variable y is unlimited and in this case, the S/N ratio can be realized as Eq. (2.4):

$$\eta = -10 \log \sum (\frac{1}{x^2})/k \quad (2.4)$$

Here, x is experimental surveillance account and k is the number of experiments. The last standard is “Nominal-is-best”: In these styles of problems, the objective account of the quality variable x is specific. According to which, the S/N ratio can be realized as Eq. (2.5).

$$\eta = 10 \log \sum x^{-2}/\sigma \quad (2.5)$$

Here, \bar{x} is the average account for the experimental surveillance and σ is the criterion variation of the experimental surveillance [24, 25].

Overall, the average values of the Signal-to-noise (S/N) ratio for each level of each of the parameter are calculated. The maximum and minimum values of differences are presented that the appropriate S/N ratio is decided based on the experimental strategy. This principally has a great influence on assessing the experiments. The Taguchi method process shown in Figure 2.19.

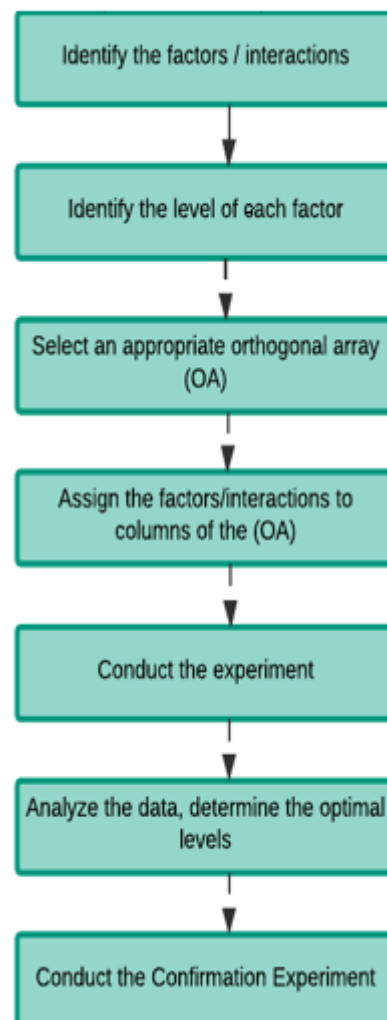


Figure 2.10: Taguchi method flowchart.

2.4 Particle Swarm Optimization Algorithm (PSO)

PSO algorithms are nature-inspired population-based metaheuristic algorithms proposed by Eberhart, Kennedy, and Shi [26, 27, 28]. These algorithms impersonator the social behavior of birds for problem solving.

The PSO algorithm is initialized with a set of random solutions which represent the particles and then searches for an optimal solution by updating the generations. In each iteration, every particle is updated by following the first two (best) values. The first solution that is achieved so far is the best solution and is stored, known as pbest. Then, the other best value that is followed by the PSO is the best solution, achieved so far by any particle in the population and this best solution is a global best and known as gbest. The particle updates the positions and velocity by using equation (2.6) and (2.7) after selecting the best two solutions.

$$X_{k+1}^i = X_k^i + V_{k+1}^i \quad (2.6)$$

$$V_{k+1}^i = V_k^i + c_1 r_1 (P_k^i - X_k^i) + c_2 r_2 (P_k^g - X_k^i) \quad (2.7)$$

Where X_k^i is represented Particle position, V_k^i represented Particle velocity, P_k^i represented Best "remembered" individual particle position (pbest), P_k^g represented Best swarm position (gbest), c_1 and c_2 Cognitive and social parameters and r_1, r_2 random parameters between (0,1).

The PSO have number of advantages when compared with other optimization algorithms, PSO is easy to execution, there are few parameters for tuning the simple flowchart of PSO presented in Figure 2.11. Therefore, PSO effectively applied in several fields: training neural network, functions optimization etc...

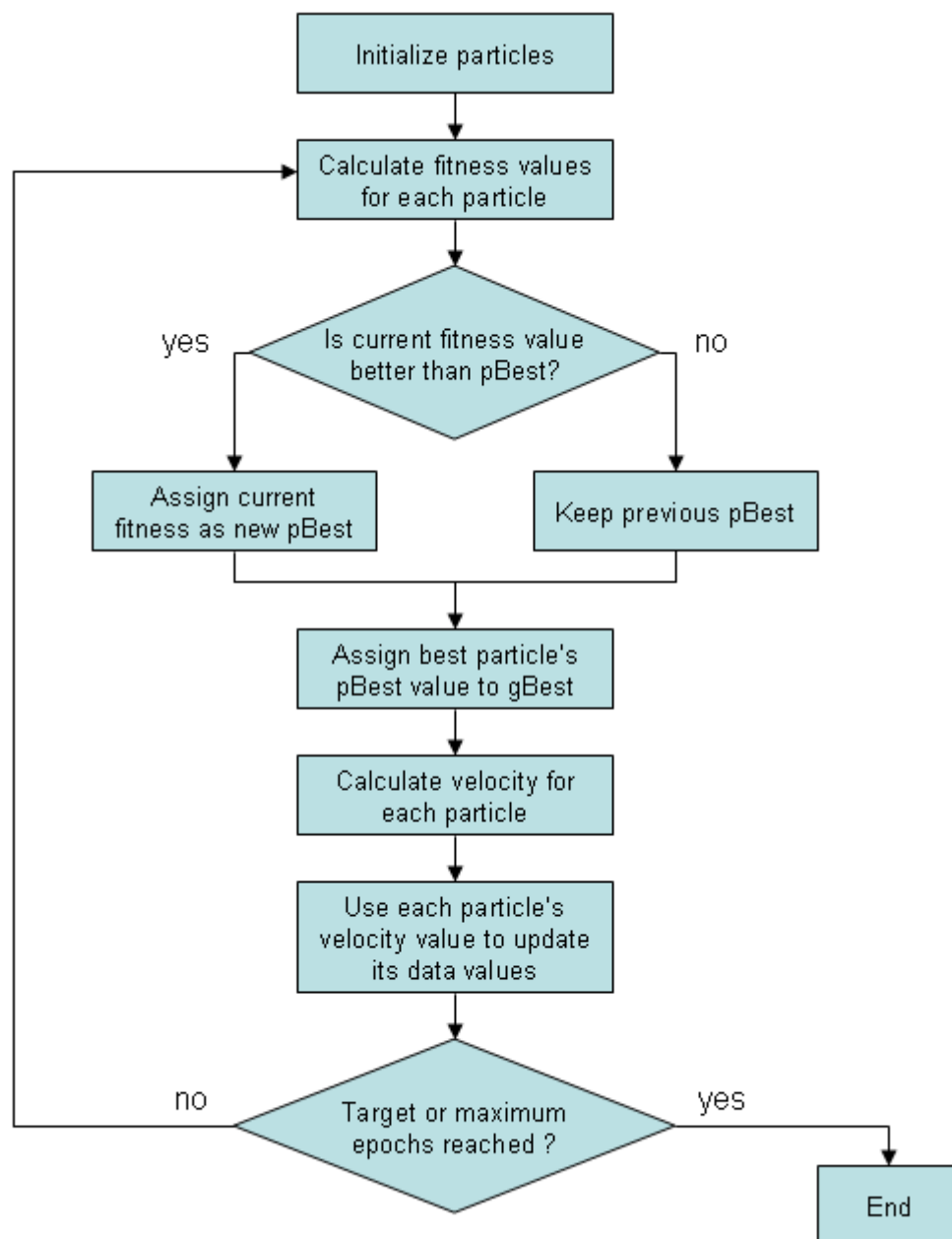


Figure 2.11: Simple pso flowchart [29].

2.5 Literature Review

In this section, number of studies were studied dealing with deep learning techniques and with the datasets that are used in this thesis. Then these studies were analysed and explained in detailed form to compare its results and techniques with our frameworks.

Lore et al. proposes a deep Auto-encoder-based method to separate features of signal from images having low-light and also modify glare images without over saturating the lighter accessories in images with a high variety [30].

K. Sun et al. proposes that a divergence of the stacked-sparse denoising auto-encoder, synthetic data used for training it, the new proposed extreme learning machine Auto-encoder (ELM-AE) called generalized extreme learning machine Auto-encoder (GELM-AE) which adds the forked regularization to the aim of ELM-AE [31].

In [32] Yihui Xiong et al. trains an Auto-encoder network to encode and remodel a geochemical pattern population with strange complex multivariate probability division.

In [33] Lyle D. Burgoonet et al. trains the Auto-encoder to Predict Estrogenic Chemical Substances (APECS). APECS consist of two deep Auto-encoder models which is less convoluted than the USEPA's method and perform at least the same achievement. However, proposed idea implements accuracies of 91% vs 86% and 93% vs 93% on the in vitro and in vivo datasets used in validating the US EPA method.

Chaoqun Hong et al. proposes a new pose retrieval technique which focuses on multimodal integration feature extraction and back-propagation deep neural network by using multi-layered deep neural network with non-linear mapping [34].

In [35] Tzu-Hsi Song et al. focuses on bone marrow trepan biopsy images and proposed a hybrid deep Auto-encoder (HDA) network with Curvature Gaussian method for active and exact bone marrow hematopoietic stem cell detection via related high-level feature correspondence.

In [36] Yosuke Suzuki et al. proposes a collaborative filtering based recommendation algorithm that employs the variation of similarities among users derived from different layers in stacked denoising auto-encoders.

Yu-Dong Zhang et al. presents a novel system counting on susceptibility-weighted imaging as computer-aided detection application which increased in the last years. Unsupervised feature learning was done by using SAE. Then, a deep auto-encoder neural network was formed using the learned features and stacked auto-encoders for training all of them together as supervised learning. The proposed approach produced a sensitivity of “ $93.20 \pm 1.37\%$ ”, a specificity of “ $93.25 \pm 1.38\%$ ”, and an accuracy of “ $93.22 \pm 1.37\%$ ” the results obtained over “10x10-fold” cross validation [37]. As presented above, Deep auto-encoders have been gathered lots of attention from researchers recently.

On the other, Taguchi method is a statistical technique proposed by Taguchi and Konishi Taguchi, which was essentially proposed for optimizing the quality manufacturing process development [38]. Especially in recent years, this method is used in number of critical studies to design experiment with best performance by different disciplines such as Engineering, Biotechnology and Computer Science. For instance, Mei-Ling Huang et al. (2014) combines a feature selection technique with SVM recursive feature elimination approach to validate the classification accuracy for Dermatology and Zoo databases [39].

In this study, the Taguchi method was adapted and combined with a SVM classifier so as to increase the overall classification accuracy by optimizing ‘ C ’ and ‘ γ ’ parameters respectively. Authors claims that the proposed method can produce more than 95% accuracy for Dermatology and Zoo databases. A study includes multi-stage metal forming process by considering workability also employs Taguchi method for optimization [40]. For this study, the Taguchi method is combined with artificial neural network to minimize the objective functions with respect to the forming process that the combinations of parameters used in finite element simulation are determined by orthogonal array in statistical design of experiments. The train data for artificial neural networks are obtained from orthogonal array and the result of simulation process.

Huimin Wang et al. [41] are adopted the Taguchi method to analyze the effect of “inertia weight”, “acceleration coefficients”, “population size”, “fitness evaluations”, and population topology on particle swarm optimization algorithm (PSO), and to determine the best mixed of them for various optimization problems. The experimental results illustrate that all the benchmark functions having their optimum solutions after the tuning process. Furthermore, acceptable results are also presented by the article when dealing with the optimization design of a “Halbach permanent magnet” motor. The paper concludes that the PSO based Taguchi method is quite appropriate for such popular engineering problems.

A recent study published in proposes a new predictive modelling of material removal rate (MRR) by employing Taguchi-entropy weight based GRA to optimize an Artificial Neural Network [42]. Further recent studies using Taguchi method can be also seen in the corresponding articles [43,44].

Furthermore, number of studies in the field of epilepsy detection are presented. In a recent study Sutrisno Ibrahim et al propose a few electroencephalography (EEG) feature selection and recognition methods for epilepsy and autism spectrum disorder (ASD) recognition [45].

EEG signal is first pre-processed and then it is decomposed into different EEG sub-bands via a DWT (Discrete Wavelet Transform). Two nonlinear approaches were proposed, namely, largest Lyapunov exponent and Shannon entropy, which calculate complexity in the EEG signal. The author also studies on the use of a cross-correlation techniques to calculate coincidence between EEG channels. Then the classification methods used to classify selected features. Different datasets are used to test the proposed system, and it is claimed that the proposed method achieves recognition result, with accuracy of up to 94.6% for a multi-channel classification problem. In an alternative approach, mel frequency cepstral coefficients are calculated in the feature selection level and used to feed a neural network for the classification problem [46]. Mel frequency cepstral coefficients are measured based on a frequency analysis. Besides, the experimental results claim that the proposed technique presents the best

results when compared with the previous studies in terms of recognition accuracy based on the same dataset.

In [47] Rafik Djemili et al. propose a new feature extraction technique, depending on empirical mode decomposition according to which signals are decomposed into considerable style functions by the empirical mode decomposition algorithm that the extracted features become input to the MLP architecture for the classification phase. Experimental results claim to achieve 100% accuracy rate with the Bonn University dataset.

In [48] Jesus Martinez-del-Rincon et al. validate two major contributions namely, the use of non-linear classifiers and the proposal of an unsupervised fashion technique, Bag-of-Words model, which extracted nonlinear features from an input EEG signal. The achievement of the technique is tested with public datasets, also private datasets are recorded under factual and in no ideal environment. The public datasets are used to compare the presented technique results with various studies using the same dataset and presents satisfactory results.

Srinivasan et al proposed new system based on Time–frequency domain for feature extraction and RNN used to classify the features, the proposed method presented 99.60 accuracy [49]. Subasi and Ercelebi proposed artificial neural network (ANN) based wavelet transform (WT) and produce only 92% performance [50]. Subasi proposes Discrete WT based on Mixture of expert model which presented only 94.5 performance [51]. Kannathal et al proposed adaptive neuro-fuzzy inference system (ANFIS) based on Entropy measures and produce 95% performance [52]. Tzallas et al. proposed a new method based on time–frequency analysis and ANN which produced a high accuracy of 100% [53]. Polat et al. proposed Fast Fourier transform and decision tree (DT) which present 98.72 performance [54]. Acharya et al. used Wavelet Packet Decomposition (WPD) to decompose segments and Principal Component Analysis (PCA) to extract eigenvalues from the resultant wavelet coefficients. Then, a supervised technique Gaussian Mixture Model (GMM) classifier was used to classify the extracted features and obtained 99% accuracy [55]. Acharya et al. proposed combination of entropies, HOS, Higuchi FD, Hurst exponent and FC

and the proposed method presented 99.70% accuracy [56]. Peker et al. proposed a complex value artificial neural network (CVANN) based on Dual Tree Complex Wavelet Transform (DTCWT), the proposed method presented 100% performance [57]. Karim et al. proposed a deep auto-encoder based on Taguchi method and the proposed method presented 100% accuracy [58].

Moreover, the most relevant research in Arrhythmias automatic methods are presented. In a recent study, a model for recognition of cardiac arrhythmias is presented [59]. The model employs k-NN and SVM supervised machine learning techniques respectively. 20-cross validation are used to obtain more realistic experimental results. The data was obtained from UCI, which is a famous machine learning datasets repository. The experimental results gave 73.8% accuracy rate via k-NN and 68.8% with an SVM classifier.

In [60] Anam Mustaqeem et al. presents a new system for classification of arrhythmias. The wrapper algorithm was used to extract best features and UCI dataset was used at the experimental section to test the accuracy of the proposed model. Different classifiers were tested with the proposed feature extracted techniques. 10-cross validation were applied on Multi-Layer Perceptron (MLP), SVM, KNN, random forest and Naïve Bayes classification techniques respectively. The experimental results showed that the MLP achieves the best classification with an average performance of 78.26%, whereas the performance evaluated for SVM and KNN were 74.4% and 76.6% respectively. They outperformed previous models in terms of accuracy.

In [61] WM Zuo et al. proposes a method for diagnosis of cardiac arrhythmia using a kernel difference weighted k-nearest neighbour classifier (KDF-WKNN). The proposed method is different from classic KNN, that the proposed method describes the weighted KNN rule as a least-squares based optimization of sample reconstruction from its neighbourhood.

Besides, in an alternative study, an ANN based architecture, relying on standard 12 lead ECG recordings for cardiac arrhythmia classification is presented in [62]. The author utilizes an MLP with a standard backpropagation algorithm to classify

arrhythmia cardiac as normal and abnormal respectively. It is claimed that the experimental results achieve 86.67% classification accuracy rate.

Furthermore [63] Golnaz Sahebi et al. propose a feature selection technique supported by a genetic algorithm to optimize the reliability of recognition in e-Health applications. The presented method known as PIGAS, employs the genetic algorithm to select the most efficient subset of features that produce the best classification accuracy. The k-NN classifier is applied to classify the selected data and the whole system is validated by using UCI Arrhythmia dataset. The paper claims that the proposed method presents 99.70% accuracy by using all features and 66.76% by using only half of features.

In [64] Anugerah Galang Persada et al. present a comparative research of several attribute selection approaches used in machine learning applications as a pre-processing step. Several search methods are employed to combine nine different attribute selection algorithms, followed by applying different attribute evaluator algorithm. Those results are then classified by using eight classifiers and the highest accuracy achieved by applying RBF Classifier on the combination of Best First search and CsfSubsetEval techniques. The paper claims to achieve 81% accuracy with the UCI Arrhythmia dataset.

In [65] Shivajirao M. Jadhav et al. propose a new approach for cardiac arrhythmia disease classification by employing a modular neural network (MNN) model, which essentially classifies arrhythmia data as either normal or abnormal. The paper claims to achieve 82.22% accuracy with the given data set. Further corresponding studies can be found in [66- 68].

In addition, this section presents several earlier studies in the field of SPECTF classification. The automatic classification is important issue for saving time and reducing the overall cost. A multi-agent classifier system relying on Q-learning for handling the data classification problem is presented in [69]. According to which, the Q-learning and Bayesian algorithms are combined and mathematically formulated for trust measurement. The method is validated by using number of small and large

datasets, and the results are compared with the previous studies to confirm the performance of the proposed system.

Srinivas et al. [70] proposes a standard multi-level classification method based on sparsity based dictionary learning SVM technique and used for health datasets. The presented approach allows satisfactory performance over medical datasets for classification accuracy and also requires no synthesis and ensemble approaches in multi-level classification. Five standard UCI medical datasets are employed to validate the competence of the proposed classification approach.

In [71] Wei et al. assorted Bayesian network with dataset extension and feature selection methods that the “Bayesian network structure learning algorithm” is suitable for huge features datasets. The experimental results claims that the new method performs higher precision than K2 algorithm and SDBNS method respectively. In an alternative study, a new support vector data description (SVDD), offering the concept of density weight and employing k-NN algorithm for target data density distribution, is proposed. The proposed method allows Priority for data points with high rise density regions. The new method is evaluated by using number of datasets with respect to the UCI repository so as to reveal the performance of the proposed method [72]. Furthermore, a new SVDD-based method, allowing to analyse uncertain data to detect outliers, is proposed. This method consists of two stages. In the first stage, pseudo-training set, each input instance is scored with a confidence value which labels the likelihood of an example tending normal class. In the second stage, on the other hand, the generated confidence score is integrated into the SVDD training phase so as to build a global distinctive classifier for outlier detection. The experiments results claims to obtain the best detection results when compared with the previous techniques [73].

Kumar et al. [74] proposes a new method known as Multiple Kernel Completion (MKC). This method completes similarity kernels and determines their better integration into a SVM framework in order to increase the discrimination margin. The performance of the proposed technique is tested by using datasets from UCI Machine Learning repository.

In an earlier work, Cui Li-lin et al. [75] proposes a method called as Transductive Confidence Machine (TCM) Improved k Nearest Neighbours (TCM-IKNN). TCM-IKNN is an efficient machine learning algorithm in which the TCM is improved by combining with k-NN and validated by using UCI dataset. Despite the claim that the proposed method reduces the number of false predication and increases the true predication, authors also state that it may present unsatisfactory results through high confidence level.

In [76] Tian et al. proposes a kind of discretization called “core-generating approximate minimum entropy discretization (C-GAME)” which determines a group of lowest entropy cuts able of creating discrete data points with nonempty cores. The C-GAME is introduced and the modeled as a “constraint satisfaction optimization problem”. Accordingly, branch and bound optimization algorithms are employed to solve it. Experiment results, performed on UCI dataset, present that the C-GAME outperforms to the number of techniques proposed in the corresponding field.

Bhatkoti et al. [77] proposed a new framework for Alzheimer’s disease diagnosis based on deep learning and KSA algorithm. In this application the results of the modified approach compared to non-modified k-sparse method. The σ KSA optimize the competence of diagnosis compared to the previous researches.

In [78] Haonan Tong et al. present software defect prediction application by using the advantages of stacked denoising auto-encoders (SDAEs) and two-stage ensemble (TSE). In the first step, SDAEs used to learn the DPs from the imitative software metrics. Moreover, a new ensemble learning method, TSE, is proposed to predicate the labels-imbalance problem. The proposed method trained and tested by using 12 NASA benchmark test data to show the effectiveness of the SDAEs TSE system which is significantly effective for SDP.

In [79] Kuo et al. proposed a stacked denoising Auto-encoder for building a deep neural network for student dropout predication is proposed. The system trained with recent year’s data and is used to estimate the results of current year to counseling in order to warn students who might dropout.

Yihui Xionget et al. [80] trained an auto-encoder neural network to encode and decode a geochemical data inhabitant with unidentified composite multivariate possibility distributions. During the training, small possibility examples contribute tiny to the deep auto-encoder network. These examples can be classified by the trained network as abnormal examples due to their reasonably greater reconstructed mistakes. The Southwestern Fujian district China is selected as a case research field.

Tao Han et al. [81] presented some ideas of deep sparse auto-encoder mix with compressed sensing (CS) theory, which can enhance the compacted selection process of CS with compressing of sparse auto-encoder in deep learning. The innovative CS theory has no function of autonomic instruction, so they present the idea of stacked auto-encoder of deep neural network to optimize CS theory. Then, calculate the mistakes between the retrieval the output features and the input features. By adjudicating the achieved error and the suitable error, the stacked auto-encoders compressed sensing (SAECS) model can select separately the most suitable sparsity and the most suitable length of dimension vector.

Syed Moshfeq Salaken et al. [82] Proposed a deep auto-encoder classification technique which first learns high level features and then trains an ANN with these learned features. Experimental results display the proposed technique presented satisfactory results when compared with all other classifiers when trained with all features and same training data.

Zahra Ezzati Khatab et al. [83] presented a novel technique which takes the advantages of deep learning, extreme learning machine (ELM) and deep extracted features by auto-encoder, to enhance the localization achievement in the feature learning and the classification. Moreover, the fingerprint dataset also needs to be updated, the author increase training data number, in order to enhance the localization achievement, progressively. experimental results point that the presented technique supplies a significant enhancement in localization achievement, by using deep features extracted by auto-encoder, and increasing the training data number.

In addition to the deep auto-encoder neural network, convolutional neural network has effective applications. Usman Mahmood Khan et al. Proposed a novel convolutional

neural network and random forest estimator to classify the complex time series input and define if it corresponds to a breathing activity and predict breathing rate. Furthermore, the author collects a comprehensive dataset for training the proposed method and evolve reference benchmarks for the future researches in the area. According to the obtained results, they conclude that convolutional neural network mixed with passive radars show high potential for end-to-end human action classification [84].

Xue-song Tang et al. [85] presented a new method based on pre-processing and deep two auto-encoders, in the pre-processing stage the input data divided into segments and then geometrics information extracted as input to the stacked auto-encoder, the proposed method produce satisfactory results when compared with CNN-based feature learning.

Chuanlong Yin et al. [86] Proposed a new approach to explore an intrusion recognition system depended on deep neural network, and proposes a model for intrusion recognition using recurrent neural networks (RNN-IDS). Furthermore, study the achievement of the proposed system in binary and multiclass classification. The proposed model compared with random forest, J48, support vector machine, artificial neural network, and other machine learning techniques presented in earlier studies on the common used data set. The experimental results present that RNN-IDS is actual appropriate for demonstrating a classification model with high performance and that its performance is larger to that of traditional machine learning classification techniques in mutually binary and multiclass classification.

Zhen Yu et al. [87] proposed a technique to automatically classify FFSP by using deep convolutional neural network (DCNN) method. The presented DCNN consists of 16 convolutional layers with small 3×3 size kernels and three fully connected layers. To reduce DCNN parameters a global average pooling (GAP) is adopted in the last pooling layer, which relieve the overfitting status and mend the achievement under fixed training data. Both the transfer learning technique and a data increase technique appropriate for FFSP are executed to further excess the classification accuracy. comprehensive experiments validate the benefit of proposed approach through

classical methods and the performance of DCNN to classify FFSP for clinical detection.

CHAPTER 3

MATERIALS AND METHODS

3.1 Deep Learning

DL or more famously introduced as deep structured learning or hierarchical learning is a partition of ML which is depended on a group of algorithms that try to model high-level abstractions in features [88, 89, 90].

DL algorithms try to learn representations because more abstract representations are often built depending on less abstract ones. A main benefit of more abstract representations is that they can be invariant to the local alterations in the input features. Learning such invariant data is a continuing main objective in pattern classification.

DL algorithms are really Deep structural design of sequential layers. Each layer used a nonlinear transformation on its input features and offers a representation in its output data. The aim is to abstract a complex and learn representation of the features in a hierarchical method by transitory the features over several transformation layers. The sensory features are served to the first layer (input layer). Accordingly, the input of each layer is output of previous layer.

The deep learning algorithms easily mean stacking number of nonlinear transformation layers. The extra layers the features go over in the deep structural design, the extra complex the nonlinear transformations which are built. These transformations epitomize the features, so Deep Learning can be measured as specific situation of representation learning technique which abstract representations of the features in a Deep Structural with several levels of representations. The realized last representation is an extremely non-linear function of the input features.

It's important to refer that the layers in the deep learning is nonlinear and try to extract and learn sensitive features from input data. The linear transformation such as PCA not used as transformation algorithms in the deep learning because its linear transformation and produce linear transformation too.

Then, the final representation of features are become output from last layer and become input to the classifier that classify the features in to labels.

Moreover, it is actual challenging to measure the effective presentations of by hand engineered features through various applications and also need time-consuming features extraction and dimension reduction technique specified beyond to get acceptable outcomes.

On the other hand, deep learning approaches for automatic feature learning supply the capability to represent features from raw data with tiny preprocessing. Multiple nonlinear transformation layer is used for abstraction, deep learning approaches representation intricate features learning from raw features and extract the best features to enhance recognition performance within minimum time. In recent years, several deep learning techniques presented such as auto-encoder, convolutional neural network and recurrent neural network etc. The details of deep learning organization shown in Figure 3.1.

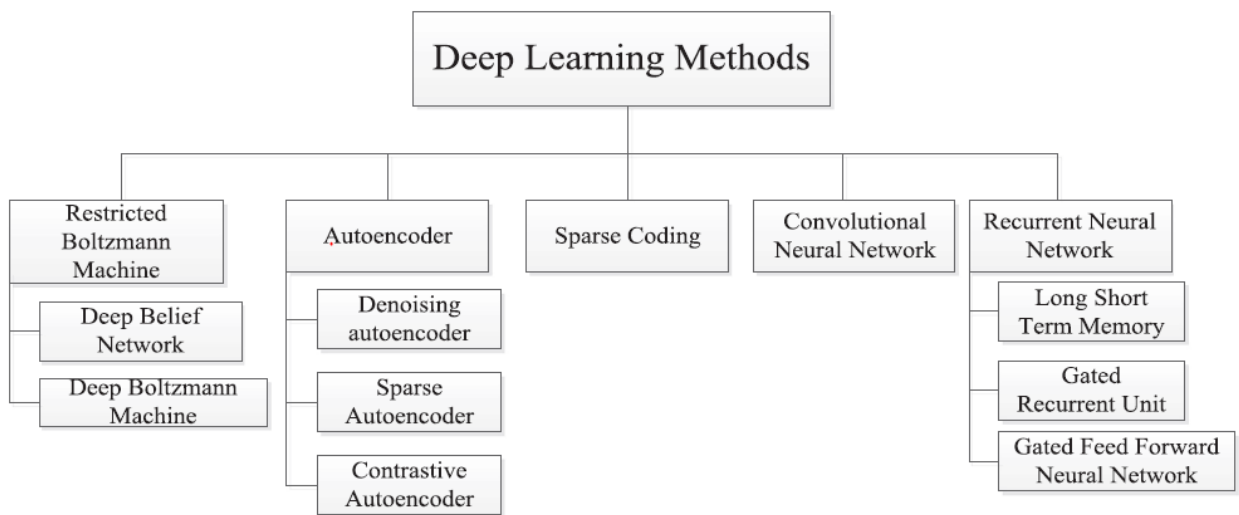


Figure 3.1: Deep learning structure [91].

3.1.1 Stacked Sparse Auto-encoder

Supervised learning is one of the most powerful tools of AI. The Stacked Sparse Auto-encoder (SSAE) is essentially a neural network consisting of multiple layers of sparse auto-encoders and mainly used as an unsupervised feature extraction method that automatically learns from unlabeled data. Output of each layer is wired to the inputs of the succeeding layer. Having a trained auto-encoder essentially refers to estimate optimal parameters by reducing the divergence between input ' x ' and output ' \hat{x} '. . The mapping between input ' x ' and output ' \hat{x} ' is given following equations:

$$\hat{x} = f(x) \quad (3.1)$$

$$n_1^{(1)} = M_f(w_{11}^{(1)} x_1 + \dots w_{15}^{(1)} x_{5+} + b_1^{(1)}) \quad (3.2)$$

$$n_i^{(1)} = M_f(w_{i1}^{(1)} x_1 + \dots w_{i5}^{(1)} x_{5+} + b_i^{(1)}) \quad (3.3)$$

Where $M()$ is an activation using sigmoid logistic function.

The final expression can be shown as follows (4):

$$n_{w,b}(x) = M_f(w_{11}^{(2)} n_1^{(2)} + \dots w_{15}^{(2)} n_5 + \dots + b_1^{(2)}) \quad (3.4)$$

The discrepancy between the input ' x ' and output ' \hat{x} '. is defined by using a cost function. This functions' first term refers the MSE whereas the second one is the regularization term. Different algorithms are preferred to solve the optimal parameters of the network; the details can be seen in [92].

3.1.2 Convolutional Neural Network

Convolutional neural network (CNN) was first presented in the end 1980 s, to deal with data that consists of matrices (multi-dimensional arrays) like images and videos [93]. Then CNN presented huge success in problems related to computer vision such as object detection, recognition and classification. Initial function of Deep CNN (DCNN) is feature extraction and (classification or regression). The CNN consists of two main functions as convolutional layers and pooling which are used for high

features extraction, where multiple convolutional layers and pooling used to extract features and the output of each layer wired to next layer. Then, extracted high level features are classified by using fully connected layers such as SVM and SoftMax. In the training phase of CNN, labelled data are used to perform training using training algorithms like backpropagation algorithm. During training, data samples are divided into mini-batches and controlled by “batch size” parameter. The stochastic gradient descent (SGD) is the basic algorithm used in training neural network. The SGD tries to update the neurons' biases and weights after every batch computation, and the goal of this algorithm is reducing the error rate [94].

The convolutional neural network consists of a number of important parameters and functions such as: padding, striding, convolutional layer, pooling, fully connected layer and normalization as can be seen in Figure 3.2.

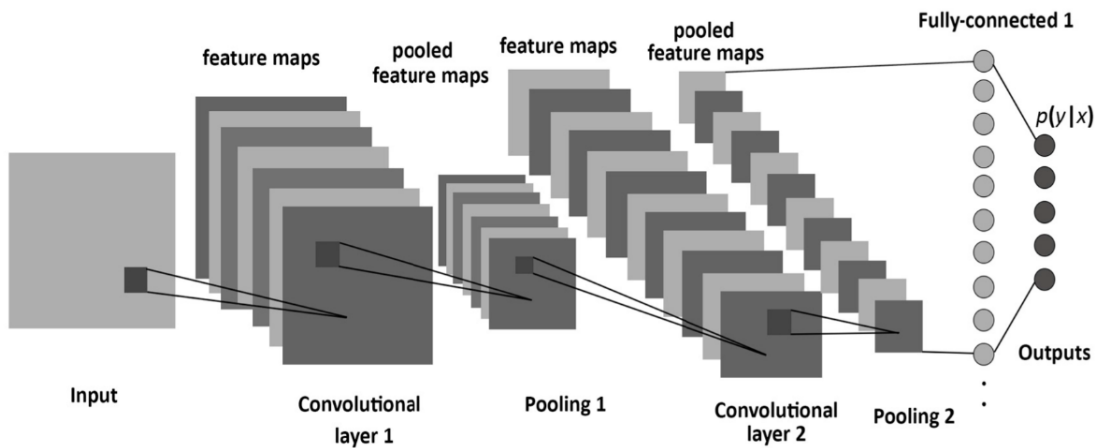


Figure 3.2: Convolutional neural network [95].

3.1.2.1 Convolutional Layer

The purpose of a Convolutional layer is to extract and learn sensitive and high level features of the input volume (image, video or sound). Only a portion of the input data is connected to the next Convolutional layer because the use of all data lead to too much computation time. Then, it applies dot products between selected range on data

and a filter on all the dimensions. The produce of this operation is the feature map [95].
The filter slides over the input matrix (image) using stride function see Figure 3.3.

1	2	1	3
3	2	2	3
4	5	9	6
4	6	8	9

1	2	1	3
3	2	2	3
4	5	9	6
4	6	8	9

1	2	1	3
3	2	2	3
4	5	9	6
4	6	8	9

1	2	1	3
3	2	2	3
4	5	9	6
4	6	8	9
1	2	1	3
3	2	2	3
4	5	9	6
4	6	8	9
1	2	1	3
3	2	2	3
4	5	9	6
4	6	8	9
1	2	1	3
3	2	2	3
4	5	9	6
4	6	8	9
1	2	1	3
3	2	2	3
4	5	9	6
4	6	8	9
1	2	1	3
3	2	2	3
4	5	9	6
4	6	8	9
1	2	1	3
3	2	2	3
4	5	9	6
4	6	8	9

Figure 3.3: Convolution layer

This example consists of 4*4 matrix (input feature), with one stride and 2*2 filter. The output of convolutional layer is a feature map as shown in Figure 3.4.



Figure 3.4: Features map of convolution layer

3.1.2.2 Pooling

The pooling layer is inserted between convolution layer in convolutional neural network structure. The pooling is a function to reduce the spatial size of the data to reduce the number of parameters and computations in neural network then control the overfitting [96, 97].

The most common form is a pooling layer with filters of size 2x2 or any size depend on the input properties applied with a stride integer number 2,3,4 down samples all depth portion in the features by any integer along both width and height, removal 75% of the activations.

Generally, two kinds of pooling are using the max pooling and average pooling methods. The max pooling means selecting the upper value feature in each stride. In this example 2*2 filter with 2 stride are studied see Figure 3.5 where 5,7,9 and 5 are the selected features and represented the greatest values.

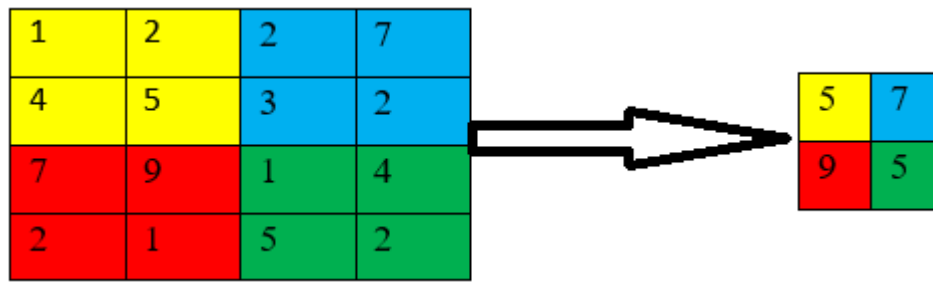


Figure 3.5: Max pooling

On the other hand, the same example used with average pooling and presented other results see Figure 3.6.

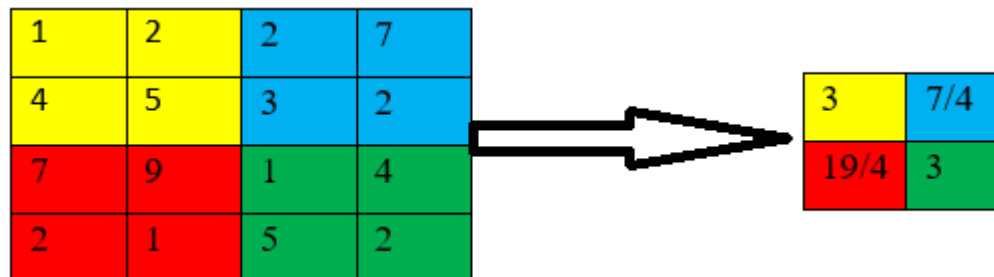


Figure 3.6: Average pooling

In each stride the features are collected and divided on the number of features, the use of max pooling or average pooling depended on the data type each of them presented satisfactory results according to the features behavior.

3.1.2.3 Fully connected layer

The fully connected layer is like a traditional neural network but in deep learning its used in the last layer after extracted the important features and number of feature (input data) are reduced. The fully connected layer mean connect every node in different layers with each other. Generally, SoftMax which is common used transfer function in this part. For example, see Figure 3.7.

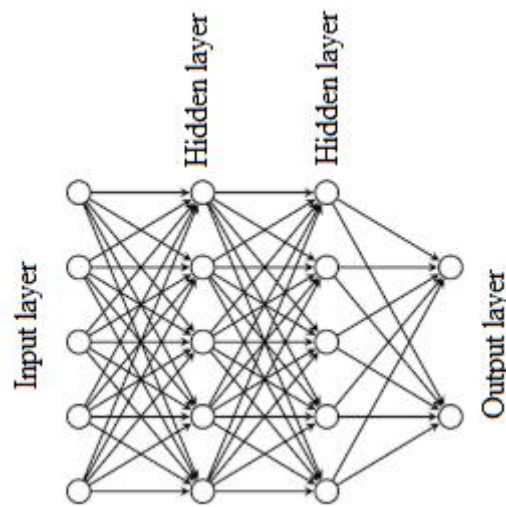


Figure 3.7: Fully connected layer

Generally, fully connected layer consists of three layers as input, hidden and output layers and there is an ability to increase the number of hidden layer according to the features behavior and size [98].

3.1.3 Recurrent neural network (RNN)

RNNs are famous models, which presented many tasks in different fields. The idea overdue RNNs is to brand procedure of the sequential data by representing features sequence to a sequence of hidden statuses, which can learn the difficult subtleties of sequence. The reappearance equalities from the hidden statuses to outputs is as presented:

$$h_t = \sigma(w_{xh} x_t + w_{hh} h_{t-1} + b_h) \quad (3.5)$$

$$O_t = \sigma(w_{ho} h_t + b_o) \quad (3.6)$$

Hence, the t th input features represented by x_t sequence $\{x_1, x_2, \dots, x_t\}$. Furthermore, the $\{x_1, x_2, \dots, x_t\}$ and $\{h_1, h_2, \dots, h_t\}$ represented the status of the hidden and output layer.

The RNN used to extracted high level and sensitive features from sequence data, which number of RNN can stacked and in the last layer fully connected layer used to classify the extracted features.

Long Short Term Memory, Gated Recurrent Unit, Gated Feed Forward Neural Network are recurrent neural network techniques and used in different fields. The Structures of RNN and LSTM shown in Figures 3.8 and 3.9.

Moreover, the RNN applied in several complex and difficult problems such as: speech recognition, EEG classification face recognition and malware detection etc [99- 104].

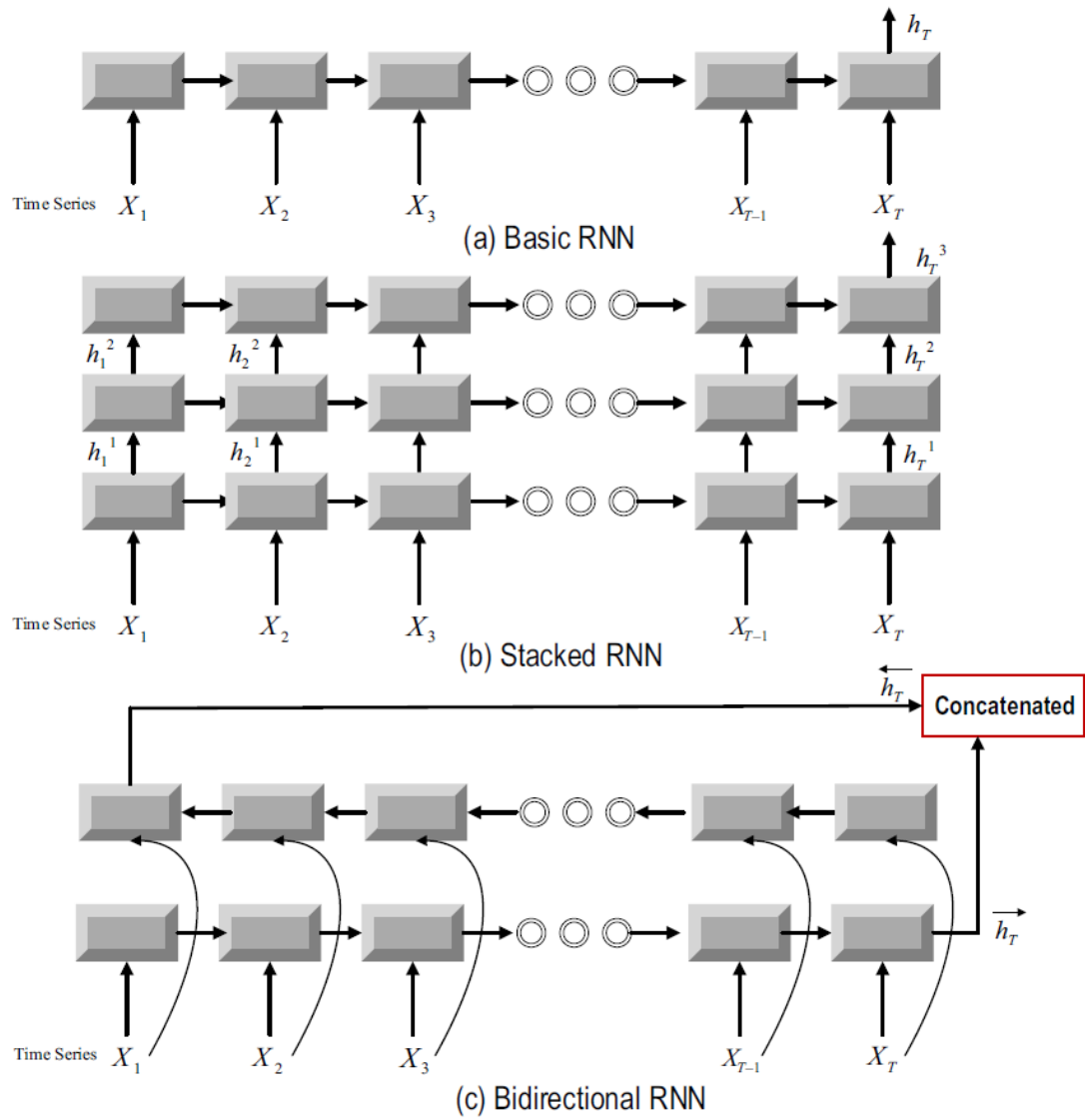


Figure 3.8: Recurrent neural network structures [105].

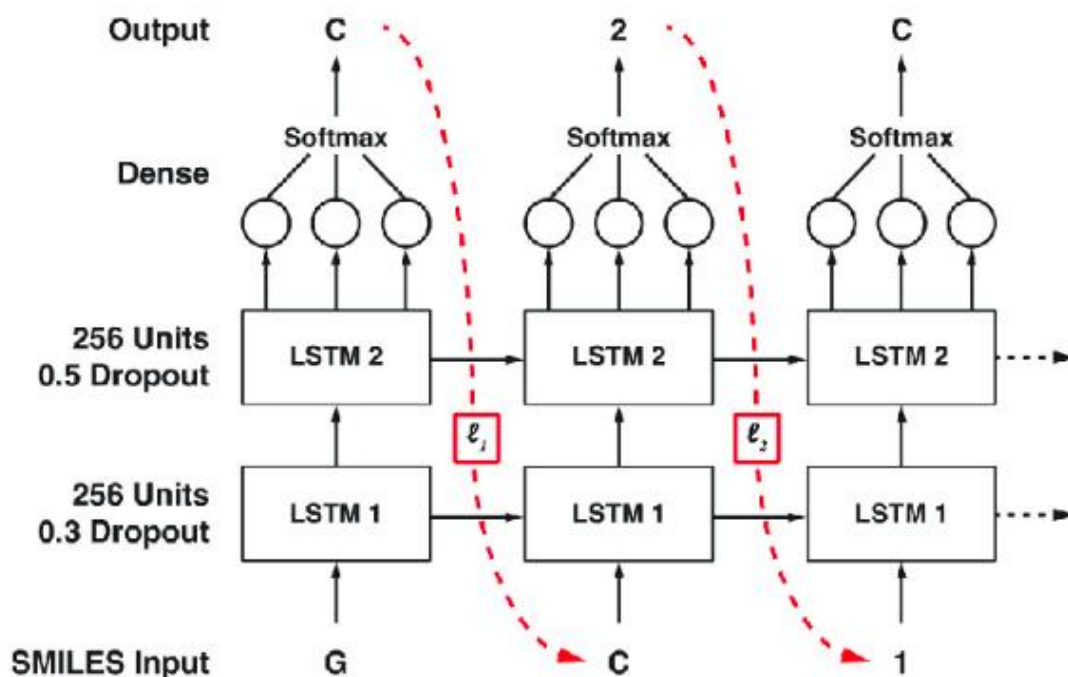


Figure 3.9: LSTM [106].

3.2 Proposed Frameworks

In this section, four frameworks are proposed for data classification and processing problems.

The first framework is a novel framework, which primarily integrates the Taguchi method to a deep auto-encoder based system without considering to modify the overall structure of the network, is presented. Several experiments are performed using various data sets from different fields, i.e., network security and medicine. The results show that the proposed method is more robust than some of the well-known methods in the literature as most of the time our method performed better. Therefore, the results are quite encouraging and verified the overall performance of the proposed framework.

Furthermore, a new framework for medical data processing which is essentially designed based on deep auto-encoder and energy spectral density concepts was proposed. According to which, the input data is first analyzed by using energy of signal

for feature extraction. This is achieved by dividing the signals in equal periods and measuring the energy of signal for each period. The produced output becomes an input to the first auto encoder for dimension reduction process that the number of auto encoders mainly depends on the characteristics and dimension of the data. The final layer employs a Soft-Max classifier so as to classify the extracted features obtained from auto-encoders and assign them into corresponding labels. In the final stage, the two Auto-encoder and the Soft-Max classifier are stacked and trained in a supervised fashion to increase the performance of the overall system. In order to validate the performance of the proposed framework, it has been tested with a number of comprehensive medical datasets, namely, Epilepsy Seizure Detection, SPECTF Classification and Diagnosis of Cardiac Arrhythmias. The obtained results are compared with literature and a very good agreement is achieved.

Moreover, Deep Auto-Encoder and Discrete Wavelet Transform (DWT) are combined for the detection of epilepsy from EEG signals. In the first stage, DWT was applied to analyze the EEG signal and 128 features were obtained by taking the A5 parameter. In the second stage, deep automatic encoders are used to obtain high level and sensitive features from the A5. In addition, these features are classified into two groups: normal and abnormal. Finally, the two auto-encoders and SoftMax stacked and trained by using backpropagation algorithm to improve the classification accuracy. The proposed method gives satisfactory results when compared with the common methods presented in this file.

Finally, the epileptic seizure detection performance has been improved by using a stacked sparse auto-encoder based on post-processing system consisting of a linear model based on the Particle Swarm Optimization Algorithm (PSO). In the first stage, all the sensitive and high-level features are extracted by using the first auto-encoder. Furthermore, the output of the first auto-encoder wired to the second auto-encoder, with its output linked to SoftMax which is followed by SoftMax training to classify the extracted features from last auto-encoder. The two auto-encoders and the SoftMax are stacked and trained in a supervised fashion using the well-known backpropagation algorithm to improve the performance of the neural network. In the second stage, the linear model transforms the predicted output of the deep stacked sparse auto-encoder

to a value close to the desired output by using the linear combination of the object features and the predicted output. This simple transformation increases the performance of stacked sparse auto-encoder. The PSO algorithm iteration is used to estimate the parameters of the linear model.

3.2.1 Deep Auto-encoder Based on Taguchi Method

Deep neural network designed from two auto-encoders and SoftMax layers, each one of them was trained alone as unsupervised training without using labelled data, the purpose of these first two layers is essentially to extract appropriate features Automatic feature extraction is one of the powerful characteristics of deep learning based architectures. The following section will briefly introduce the Taguchi method whereas the following sub-section will introduce the proposed method and the corresponding deep learning based architecture. The third layer is the SoftMax layer, which is one of the leading feature classifier, is responsible from classifying the features that extracted from the previous layers. The final layer is to stack all layers and train them together by using labelled data in supervised fashion. This basically allows to convert an unsupervised learning architecture into a supervised learning architecture. To obtain the best performance from the first auto-encoder, Taguchi method is integrated into the model aiming to estimate optimized combination of five parameters of first auto-encoder, namely, *L2 Weight Regularization*, *Sparsity Regularization*, *Sparsity Proportion*, *Hidden Size*, and *Max Epochs*. The effect of an L2 regularizer for the weights of the network is controlled by *L2 Weight Regularization* but not control the biases.

L2 Weight Regularization parameter should be very small and is represented in Eq. (3.7).

$$\Omega_{weight} = \frac{1}{2} \sum_l^L \sum_j^n \sum_i^k w_{ji}^{(l)^2} \quad (3.7)$$

Where number of hidden layers is represented by L , the number of observations is represented by n , and the training data variables number is represented by k .

The Sparsity regularizer effect is controlled by a Sparsity Regularization parameter, dealing to force a chain on the sparsity of the output from the hidden layers. This is different from applying a sparsity regularizer to the weights that sparsity regularization term can be the Kullback-Leibler divergence (KL) function [107] as illustrated in Eq. (3.8).

$$\Omega_{sparsity} = \sum_{i=1}^{D(1)} KL(\rho || \hat{\rho}_i) = \sum_{i=1}^{D(1)} \rho \log(\rho || \hat{\rho}_i) + (1 - \rho) \log\left(\frac{1 - \rho}{1 - \hat{\rho}_i}\right) \quad (3.8)$$

Where, ρ represents the desired value, $\hat{\rho}_i$ represents the average output activation of a neuron i , and KL is the function that measures the variation between two probabilities distribution through the same data. As it can be inferred that, the equation result value gets close to zero between ρ and $\hat{\rho}_i$ when input and output data resembles each other. On the other hand, when those values are not closed to each other, the sparsity will take a larger value [107].

Alternatively, Sparsity Regularizer Parameter is controlled by Sparsity Proportion (SP) parameter. The sparsity of the output from each hidden layer is controlled by the Proportion parameter. A low value for SP normally leads all neurons in the hidden layer specialized by only producing a high output value for a small amount of training examples. For instance, if SP value is selected as “0.2”, an average output for each neuron becomes “0.2” in the hidden layer over the training examples. The optimum value of SP varies depending on the nature of the problem between 0 and 1. Therefore, the technique for selecting the optimal value is very significant to improve the overall performance of the sparse auto-encoder [108]. In addition, Hidden Size (HS) is a parameter which controls the size of the feature on each layer so, it affects the performance of the auto-encoder. The last parameter is Maximum Epochs; one epoch represents one entire training cycle on the training data. Every sample in the training data is seen once, start still brand starting with the 2nd epoch. However, the Maximum Epochs mean for example: if maximum epoch equals to 10, this means the weights will be updated at least 10 time.

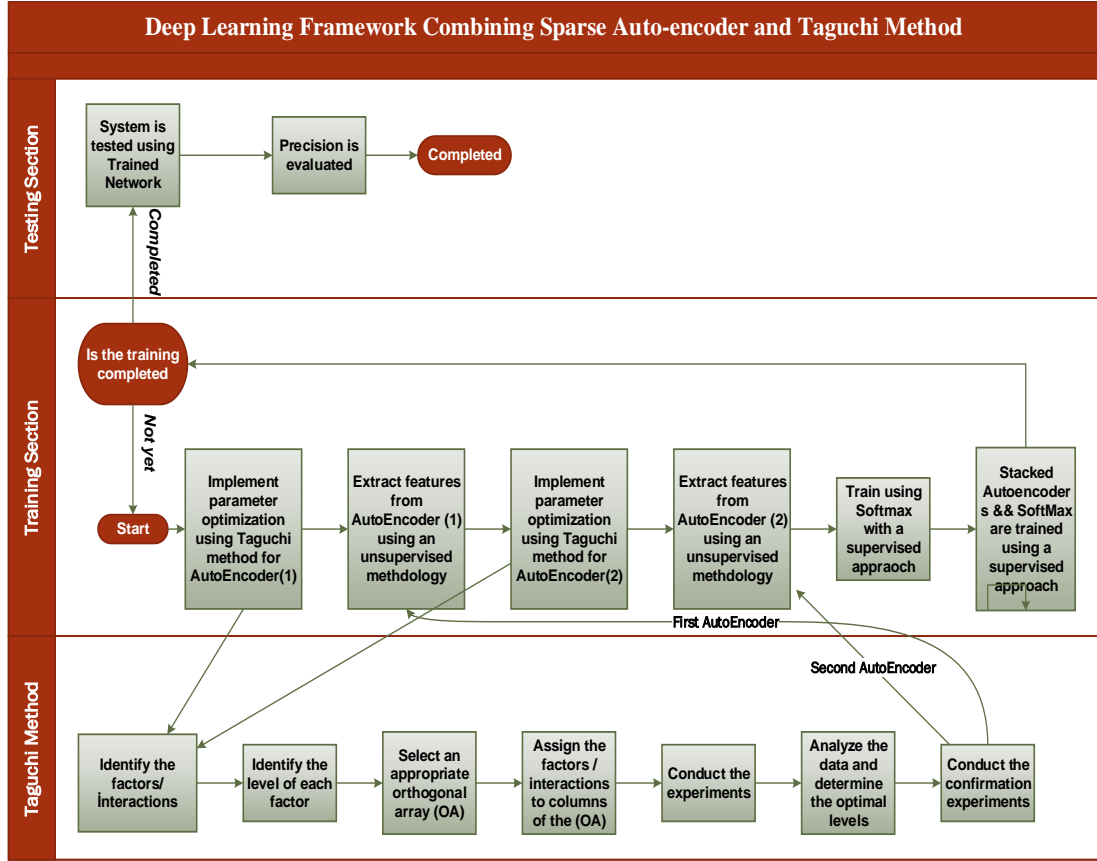


Figure 3.10: Deep learning framework combining sparse auto-encoder and taguchi method

All previously defined parameters are employed in the training phase and directly influences the success of the training process. The cost function of training sparse auto-encoder is also illustrated Eq. (10). The training algorithm tries to reduce the cost function by finding the optimal parameters that essentially aims to reduce the value of E .

$$E = \frac{1}{N} \sum_{n=1}^N \sum_{k=1}^k (x_{kn} - \hat{x}_{kn})^2 + \lambda * \Omega_{\text{weights}} + \beta * \Omega_{\text{sparsity}} \quad (3.9)$$

Here, E represented the loss rate (error rate), x is represented the input features, \hat{x} is the reconstructed features, λ is the coefficient for the L2 *Weight Regularization* and β coefficient for the *Sparsity Regularization*.

The given problem includes two auto-encoders. Each of those auto-encoder has 5 parameters and each parameter can be defined with 5 different levels. Consequently,

the traditional method for finding best combination of parameters for two auto-encoders requires $5^5 + 5^5 = 3125 + 3125 = 6250$ trails so as to test all parameter combinations by using full factorial design. This means that each auto-encoder entails 5^5 trails to obtain the best combination of parameters. Hence, a more optimized approach has been proposed in this study. According to which Taguchi method was utilized for finding the optimal parameters for the system by performing only 25 experiments, selecting L25 orthogonal index (5 parameters and 5 levels in each parameter), see Section 2.2. As the first Auto-encoder is performed by doing 25 experiments, the most optimum parameters were also determined by Taguchi method and best performance for the second Auto-encoder as well. This means the total experiments for the first two layers in our system are $25 + 25 = 50$. As motioned above, at the last step, all three components are stacked and trained in a supervised fashion by using backpropagation on multilayer network for improving the network performance. In order to validate the performance of the proposed system, a series of experiments were conducted.

3.2.2 Deep Auto-encoder based Energy Spectral Density

A new framework based on energy spectral density and deep sparse auto-encoder. Figure 3.11 illustrates the flowchart of the proposed framework. Primarily, the first step of the framework involves a pre-processing layer. This layer first converts the data having different dimensions into one dimensional array or signal data to create a matrix of array that essentially becomes an input to the framework. Energy spectral density defines how the energy of a signal is circulated with frequency. Afterwards, the Energy Spectral Density (ESD) of the signal is calculated for different periods based on Equation 3.10. This both allows to extract features from each period and to reduce the dimension of the input data. For Equation 3.10, ESD refers the total energy of a signal $x(f)$.

$$ESD = \int_{-\infty}^{\infty} x(f)^2 df \quad (3.10)$$

The energy of signal in time domain can also be calculated with the following equation 3.11.

$$E = \int_{-\infty}^{\infty} x(t)^2 dt \quad (3.11)$$

Where t refers features vector between $-\infty$ and ∞ and limits. Aforementioned, the ESD reduces the number of features which respect to the selected period value. For instance, if the number of features is 4096, it will be obtained a data structure with 1024 features after applying the ESD transformation with 0.25Hz frequency. This essentially refers that one of each 4 features are obtained respectively in order to reduce the features dimension. The period values may become an integer value such as 1,2,3,4,5...etc. The frequency or the period values are estimated by employing a trial and error approach which facilitates to obtain the best features in order to produce the highest accuracy.

As mentiond above, the usage of ESD transformation on big data is quite suitable to obtain high classification accuracy with reduced computation time. Despite the simplicity of ESD transformation, the results of this study verifies that it generates strong features for training phase. According to the proposed framework, illustrated in Figure 3.11, the output of the energy based feature extraction level becomes an input to the first auto-encoder. This Auto-encoder is trained in an unsupervised manner. The output of this auto-encoder becomes an input to the second auto-encoder and it is also trained in an unsupervised manner, producing a new data array with lower dimensions. Each Auto-encoder is trained by employing using the cost function illustrated in Equation 3.9. E value is regulated by employing mean square error (MSE) approach.

Here, E is the loss rate (error rate), x is the input data, \hat{x} is the reconstructed data, λ is the coefficient for the *L2 Weight Regularization* and β coefficient for the *Sparsity Regularization*, L is the number of hidden layers, n is the number of observations, k is the training data variables number.

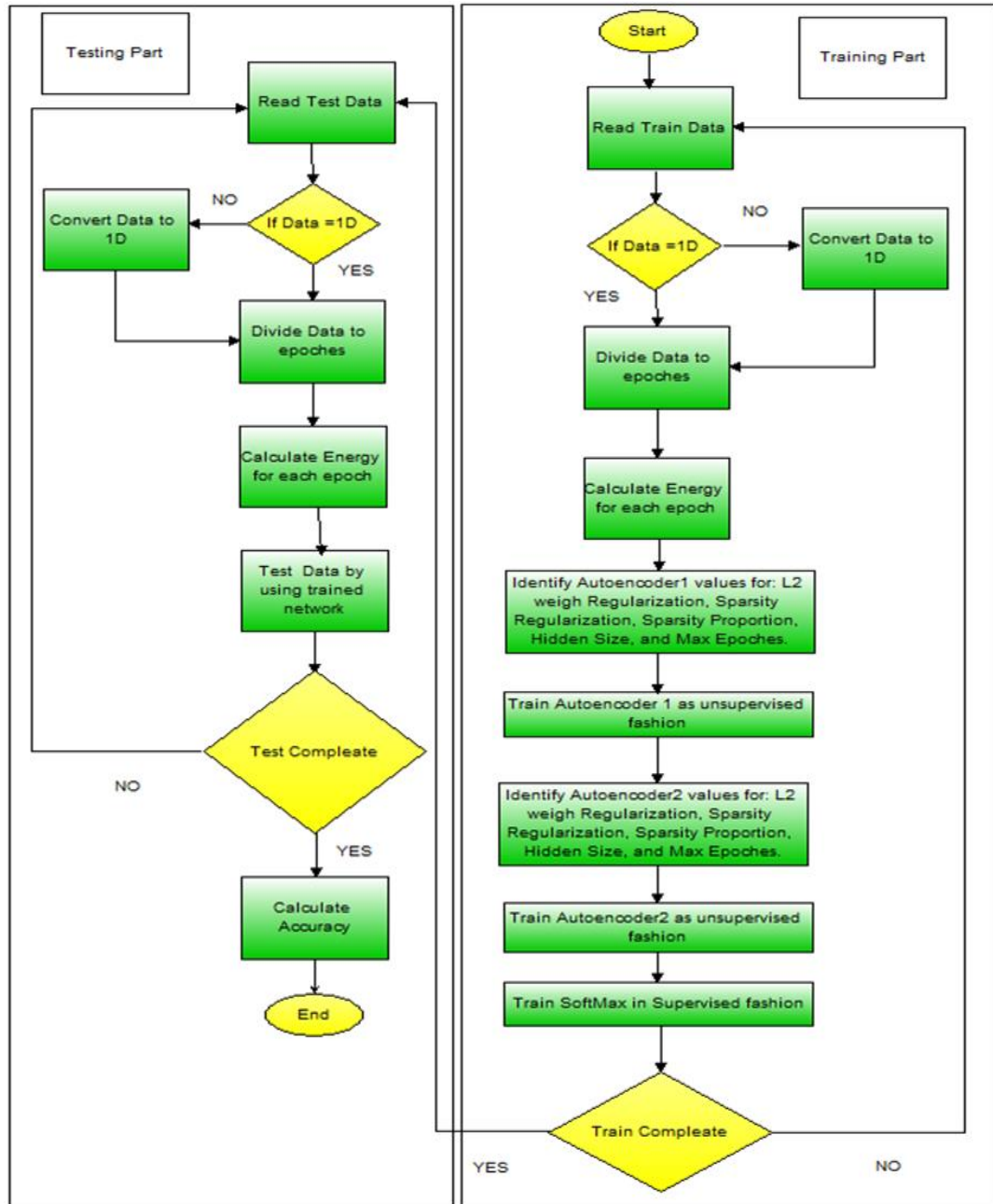


Figure 3.11: The proposed framework using SSAE architecture for training and ESD for feature estimation

The $\Omega_{weights}$ parameter represent the Weight Regularization and is calculated as follows in Equation (3.7):

Here L signifies number of hidden layers, n denotes the number of observations, and k is the number of hidden layers.

Moreover, $\Omega_{sparsity}$ illustrates the Sparsity Regularization and can be defined as follows in Equation (3.8) :

Here, desired value represented by ρ , $\hat{\rho}_i$ signifies the average output activation of a neuron i , and KL is the function that measures the difference between two probabilities distribution over the same data. Besides, SoftMax classifier also is trained to classify the output of the last auto-encoder into labelled output. That is because it gives normalized class probabilities as outputs and is a preferred way to maximize entropy between estimated and ground truth probabilities. As z_i refers to the class score assigned for class i in the final Auto-encoder layer, the Softmax function is illustrated as follows:

$$f_j(z) = \frac{e^{z_k}}{\sum_k e^{z_k}} \quad (3.12)$$

In the last stage, the two auto-encoders and SoftMax classifier are stacked and trained in a supervised manner. As shown in the architecture illustrated in Figure 3.11.

3.2.3 Deep Auto-encoder Based on Discrete Wavelet Transform

DWT was used to extract important features from input data that consist from 4096 values for each case. The features calculated by using the equation 1 and the A5 parameter calculated for input data. Then, each row data consists from only 128 features which mean gain in the processing time and reduce the computation of the problem see Figure 3.12.

As shown in the Figure 3.12 the number of features that obtained by A5 is only 128 which become input to the deep auto-encoder model for classification the data in to two case normal and abnormal.

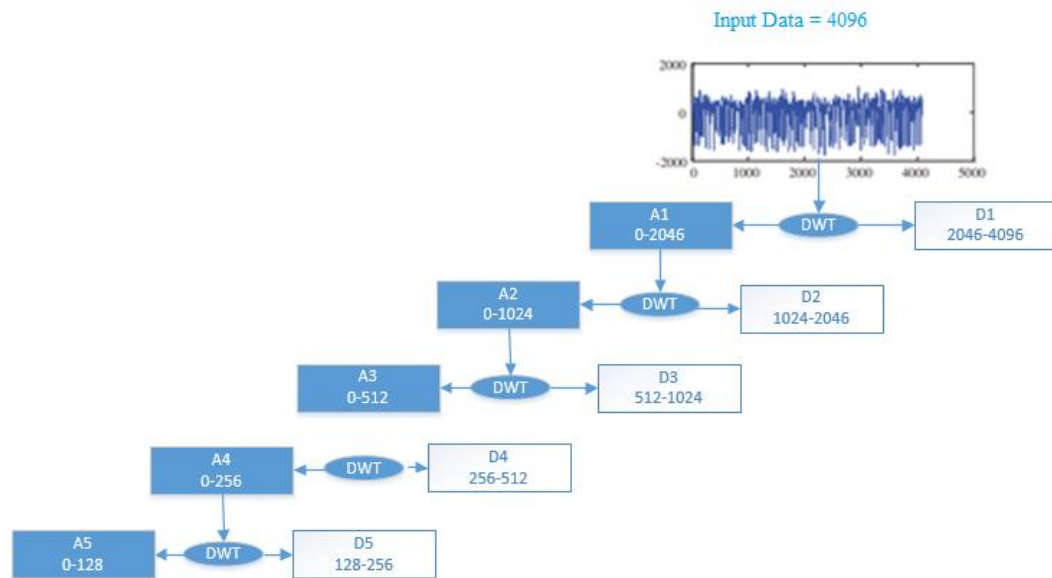


Figure 3.12: DWT process.

which also extracted sensitive features and reduce the dimension of the input signal. Furthermore, the output of last auto-encoder become input to the SoftMax which trained in supervised fashion to classify the features into normal and abnormal classes

Finally, the two auto-encoder and SoftMax stacked in supervised learning by using the backpropagation learning algorithm to increase the performance of the deep neural network see Figure 3.13.

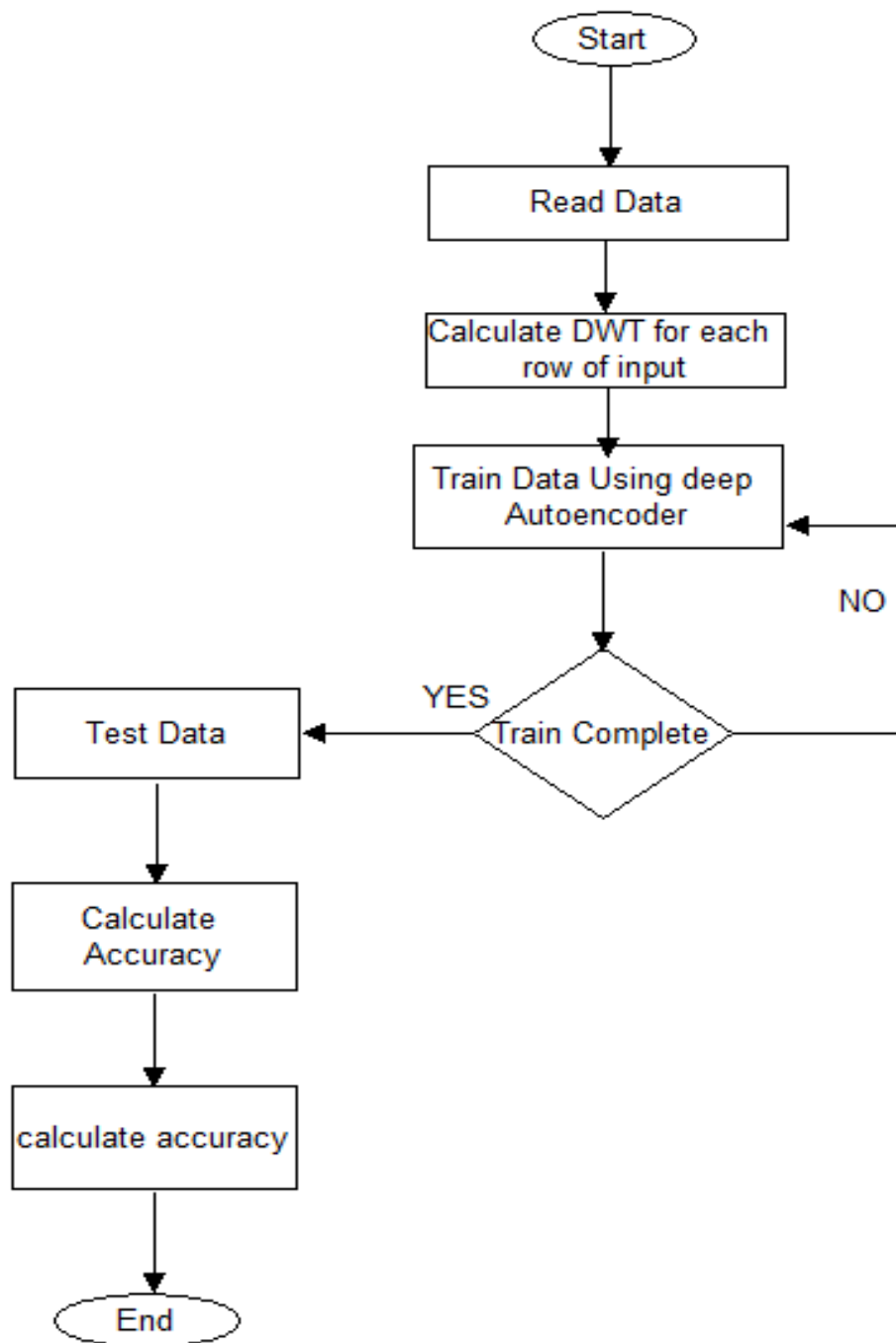


Figure 3.13: Deep auto-encoder based on DWT.

3.2.4 Deep Auto-encoder Optimized with Linear Model Based on PSO Discrete Wavelet Transform

Suppose a trained deep stacked auto-encoder is used to classify an object into one of M classes as in Figure 3.14. The input layer of the deep stacked auto-encoder consists of N neurons that correspond to object features X_1, X_2, \dots, X_N , and the output layer consists of M neurons that correspond to the desired output (class label) $Z_1, Z_2, Z_3, \dots, Z_M$ (see Figure 3.14).

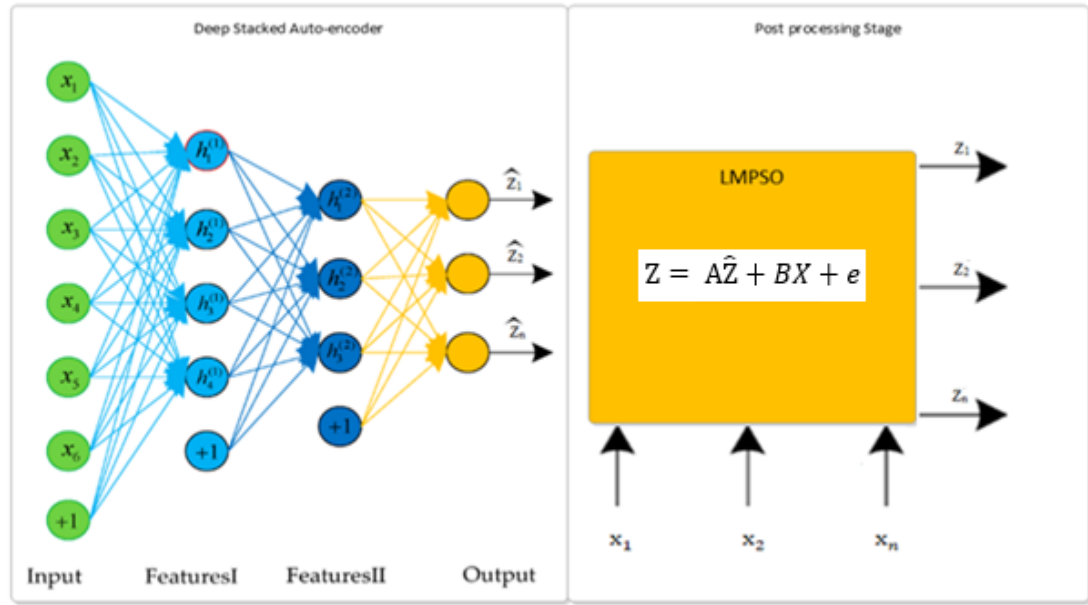


Figure 3.14: The deep learning framework based on a linear model and pso.

The deep auto-encoder consists of two auto-encoders and SoftMax, where the auto-encoders try to learn the sensitive and high-level features from the input data X . The aim of using a number of auto-encoders is to reduce the number of features gradually because reducing the number of features suddenly in one auto-encoder can lead to missing important features and affect the accuracy. The cost function of the stacked auto-encoders is represented as Equation 3.9.

where the error rate is represented by E , the input features are represented by x , the reconstructed features are represented by \hat{x} , λ is the coefficient for the L2 Weight Regularization, β is the coefficient for Sparsity Regularization, and $\Omega_{weights}$ represents the L2 Weight Regularization, which can be calculated as shown in Equation. 3.7.

where L represents the number of hidden layers, n represents the number of observations, and k represents the training data variables number.

Finally, $\Omega_{sparsity}$ is the Sparsity Regularization parameter which controls the dealing to power a chain on the sparsity of the output from the hidden layers, as illustrated in Equation 3.8.

where the desired value is represented by ρ , $\hat{\rho}_i$ represents the average output activation of a neuron i , and KL is the function that measures the variation between two probability distributions through the same data. Furthermore, the features that produce minimum cost in Equation 3.9 are selected and become input to SoftMax, see Equation 3.12. SoftMax is used as a classifier of the extracted features from X to the labels Z Figure 3.14.

where x is the input vector, i indexes the output units, so $i = 1, 2, \dots, q$. SoftMax is used without other classifiers because SoftMax is a transfer function and multiclass classifier which acts like an output layer as an output layer to the previous auto-encoders.

Then, the two auto-encoders and SoftMax are combined and trained by using a backpropagation algorithm in a supervised fashion to enhance the performance of the network. See Figure 3.15.

Moreover, antithetically to previous deep learning applications, the output of the deep auto-encoder is not the final prediction but optimized by using a linear model. The output of the deep auto-encoder \hat{Z} is processed in a linear model by using X , coefficients A , B and the error rate e to produce the optimized result Z see Equation 3.13.

$$Z = A\hat{Z} + BX + e \quad (3.13)$$

The details of linear model mathematics explained in [109] and the whole framework flowchart illustrated in Figure 2. The A and B are coefficient which values can be estimated by using PSO and the parameters of PSO selected depended on the problem

type and input features. In each iteration of PSO the predicated Z are controlled by using MSE with optimal predication Q as illustrated in Equation 3.14.

$$\text{MSE} = \frac{1}{m} \sum_{i=1}^m (Q_i - Z_i)^2 \quad (3.14)$$

Where m is number of examples, Q_i is the optimum labels for input features and MSE is the discrepancy rate between the z_i and Q_i .

The MSE is represented as cost function which PSO used to minimize its value by estimating best values for parameters A , B and e . The proposed framework is shown in Figure 3.15.

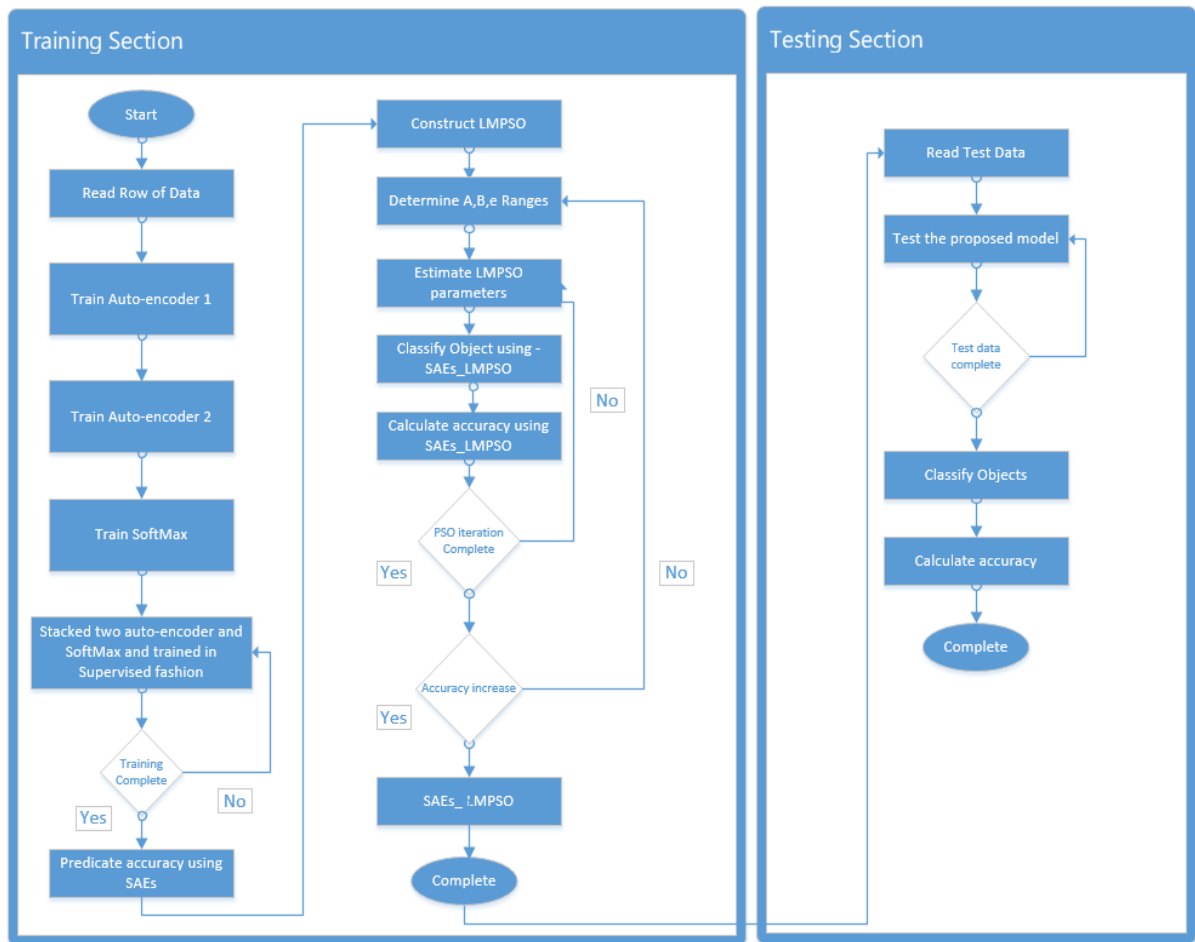


Figure 3.15: The deep learning framework based on a linear model and pso training and testing flowchart.

CHAPTER 4

EXPERIMENTAL RESULTS AND DISCUSSIONS

A computer with Intel Core i7–6700 CPU @ 2.60-GHz and 8-GB RAM is used for running the proposed framework which is used in several applications to detect computer network attacks including DDOS and IDS attacks as well as Epileptic Seizure Recognition and Handwritten Digit classification. The results obtained with the proposed method are compared to a number of studies in the respective field. In addition, some of the techniques implemented in this thesis to compare the results with our proposed method are: SVM, neural network, SoftMax and stacked sparse auto-encoder based support vector machine (SSAE-SVM). Each datasets and corresponding results will be detailed in the following sub-sections respectively.

In order to assess the performance of the proposed method, number of different performance evaluation criteria were employed, namely, the accuracy (i.e., hit rate), “sensitivity (TPR)”, “specificity (SPC)”, “Precision (PPV)”, “Negative Predictive Value (NPV)”, “False Positive Rate (FPR)”, “False Discovery Rate (FDR)”, “False Negative Rate (FNR)”, “Accuracy (ACC)”, “F1 Score (F1)” and “Matthews Correlation Coefficient (MCC)”. The equations of these parameters are presented below (4.1- 4.10). For all, TP refers True Positive, FP is False Positive, whereas TN is for True Negative and TP is for True Positive.

$$TPR = \frac{TP}{(TP + FN)} \quad (4.1)$$

$$SPC = \frac{TN}{(FP + TN)} \quad (4.2)$$

$$PPV = \frac{TP}{(TP + FP)} \quad (4.3)$$

$$NPV = \frac{TN}{(TN + FN)} \quad (4.4)$$

$$FPR = \frac{FP}{(FP + TN)} \quad (4.5)$$

$$FDR = \frac{FP}{(FP + TP)} \quad (4.6)$$

$$FNR = \frac{FN}{(FN + TP)} \quad (4.7)$$

$$ACC = \frac{(TP + TN)}{(P + N)} \quad (4.8)$$

$$F1 = \frac{2TP}{(2TP + FP + FN)} \quad (4.9)$$

$$MCC = \frac{TP*TN - FP*FN}{\sqrt{((TP+FP)*(TP+FN)*(TN+FP)*(TN+FN))}} \quad (4.10)$$

4.1 Experimental Results by Using Sparse Auto-encoder and Taguchi Method Framework

4.1.1 DDOS Detection using the proposed framework

Distributed Denial of Service attack is an offensive and threatening intrusive threats to online servers, websites, networks and clouds. The purpose of DDoS attack is to exhaust exchequer and to expend bandwidth of a network system. Due to the harmonious nature of DDoS attack, an attacker can generate massive amount of attack traffic using a huge number of compromised machines to smash a system or website [110][111]. Many organizations such as Amazon, eBay, CNN and Yahoo were the victims of DDoS attacks in the recent past. In this paper, our new framework was used to detect DDOS attack proposed in [112], which presented four attacks types (Smurf, UDP Flood, SIDDOS, HTTP Flood and normal). This dataset consists of 27 features: (*SRC ADD, DES ADD, PKT ID, FROM NODE, TO NODE, PKT TYPE, PKT SIZE,*

FLAGS, FID, SEQ NUMBER, NUMBER OF PKT, NUMBER OF BYTE, NODE NAME FROM, NODE NAME TO, PKT IN, PKTOUT, PKTR, PKT DELAY NODE, PKTRATE, BYTE RATE, PKT AVG SIZE, UTILIZATION, PKT DELAY, PKT SEND TIME, PKT RESEVED TIME, FIRST PKT SENT, LAST PKT RESEVED). In Table 4.1, parameters are classified into 5 classes and recognize the upper and lower boundaries of the parameters. The upper and lower boundaries of these parameters are determined by using trial and error approach. This approach considers the results of the predefined experiments and studies.

Table 4.1: Auto-encoder 1 upper and lower level values of factors for ddos detection.

Factors	Lower Limit	Upper Limit
Hidden Size (HS)	19	22
Max Epochs (ME)	300	500
L2Weight Regularization (L2)	0.0035	0.0045
Sparsity Regularization (SR)	4	6
Sparsity Proportion (SP)	0.13	0.16

As mentioned above, dataset consists of five classes that each class consists of 800 samples. 50% of them were used for training, and also the other % 50 was used for testing. Consequently, the proposed framework was trained by employing 2000 samples and then it was tested by employing another 2000 samples. Moreover, in Table 4.2, the operators' level values are presented. Minitab program experiments results present in Table 4.3. The error accounts obtained from stratifying the parameters to the Auto encoder 1 are represented in Table 4.5. Root mean square error (RMSE) is used to measure the performance of the Auto encoder 1, the smallest value which is closed to zero mean that the performance is well.

$$\chi = \frac{\sqrt{\sum_{i=1}^n (X_{(Obs,i)} - X_{(model,i)})^2}}{n} \quad (4.11)$$

Where spotted rate represented by X_{Obs} and modelled rate represented by X_{Model} at time/place 'i'. The experiment results acquired by using the Taguchi experimental design were estimated by transforming them into S/N ratios. The results acquired by

using the Taguchi experimental design were predestined by transforming the results into signal/noise (S/N) ratios (See Table 4.4).

Table 4.2: Auto-encoder (1) upper and lower level values of factors for ddos attack detection.

Factors	Level 1	Level 2	Level 3	Level 4	Level 5
HS	18	19	20	21	22
ME	200	300	400	500	600
L2	0.003	0.0035	0.004	0.0045	0.005
SR	3	4	5	6	7
SP	0.13	0.13	0.15	0.16	0.17

Table 4.3: Auto-encoder 1 parameters values obtained by using taguchi method for ddos attack detection.

Levels	HS	ME	L2	SR	SP	RMSE
1	18	200	0.0030	3	0.13	25.4501
2	18	300	0.0035	4	0.14	24.7128
3	18	400	0.0040	5	0.15	24.0935
.
.
24	22	500	0.0040	4	0.13	25.1571
25	22	600	0.0045	5	0.14	25.2895

Table 4.4: Auto-encoder 1 s/n ratios obtained in the taguchi experimental design for ddos attack detection.

Levels	HS	ME	L2	SR	SP
1	-27.72	-27.72	-27.33	-27.59	-27.83
2	-27.54	-27.69	-27.28	-27.09	-27.33
3	-27.31	-27.31	-27.49	-27.54	-27.25
4	-26.91	-27.44	-27.49	-27.54	-27.25
5	-27.68	-27.51	-27.32	-27.34	-27.18
Delta	0.80	0.46	0.48	0.51	0.65
Rank	1	5	4	3	2

Now, Table 4.5 presents the best parameters for Auto encoder 1 which represented the first layer in the deep auto encoder neural network. The parameters of Auto encoder 2 which represented the second layer can be obtained by using the same steps in different ranges for each parameter Table 4.6.

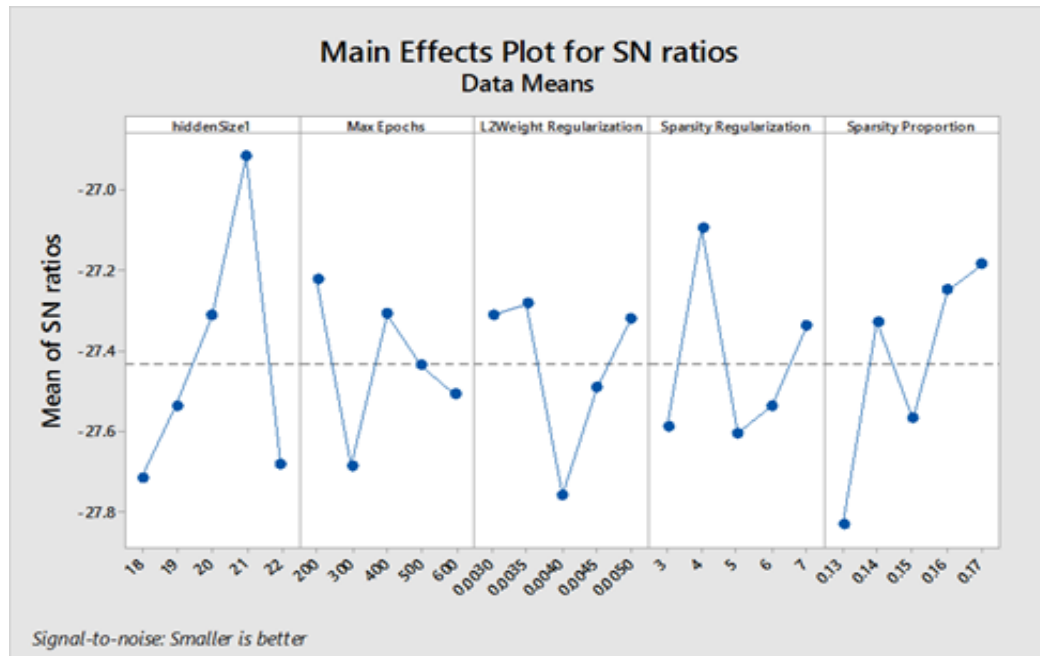


Figure 4.1: Auto-encoder 1 main effect of experimental parameters on the S/N ratio for DDOS

Table 4.5: Auto-encoder 1 parameters set obtained through optimization for ddos detection.

Factors	Value
HR	21
ME	200
L2	0.0035
SR	4
SP	0.17

Table 4.6: Auto-encoder 2 upper and lower level values of factors for ddos detection.

Factors	Lower limit	Upper limit
HS	18	22
ME	200	400
L2	0.003	0.005
SR	3	7
SP	0.13	0.17

By following the Tables 4.7, 4.8, 4.9 the best parameters obtained in the Figure 4.2 and Table 4.10, the same procedures in the Table 4.2, Table 4.3, Table 4.4 respectively that used to find the best parameters that represented in the Figure 4.2 and Table 4.4. This means that the best parameters of each Auto-encoder was determined in minimum number of tests.

Table 4.7: Auto-encoder 2 upper and lower level values of factors for ddos detection.

Factors	Level 1	Level 2	Level 3	Level 4	Level 5
HS	10	11	12	13	14
ME	50	60	70	80	90
L2	0.0010	0.0015	0.0020	0.0025	0.0030
SR	3	4	5	6	7
SP	0.10	0.11	0.12	0.13	0.14

Table 4.8: Auto-encoder 2 parameters values obtained by using taguchi method for ddos attack detection.

Levels	HS	ME	L2	SR	SP	RMSE
1	10	50	0.0010	3	0.10	0.011309
2	10	60	0.0015	4	0.11	0.012012
3	10	70	0.0020	5	0.12	0.012019
.
24	14	80	0.0020	4	0.10	0.011987
25	14	90	0.0025	5	0.11	0.011979

After finding the best parameters, for each Auto-encoder, this leads to obtain the best performance for training each Auto-encoder by using the best parameters. In other hand, the results that were obtained from the system presented by using confusion matrix for detail analysis for each type of DDOS attack is seen in Figure 4.3. The experimental results show that proposed method have satisfactory results when compared to other methods.

Table 4.9: Auto-encoder 2S/N ratios obtained in the taguchi experimental design for ddos attack detection

Levels	HS	ME	L2	SR	SP
1	38.53	38.85	38.70	38.68	38.45
2	38.72	38.56	38.36	38.61	38.60
3	38.30	38.50	38.62	38.43	38.71
4	38.76	38.59	38.66	38.35	38.56
5	38.60	38.41	38.56	38.84	38.58
Delta	0.46	0.44	0.33	0.49	0.25
Rank	2	3	4	1	5

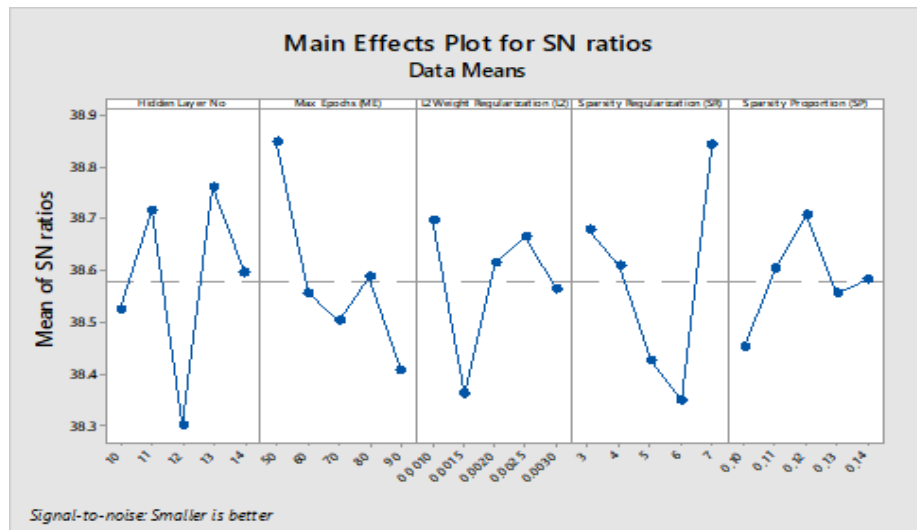


Figure 4.2: The main effect of experimental parameters on the S/N ratio for auto-encoder 2 for ddos.

Table 4.10: Auto-encoder 2 parameters set obtained through optimization for ddos detection

Factors	Value
HR	14
ME	50
L2	0.0010
SR	3
SP	0.13

Detection accuracy of 99.6% makes the proposed method slightly better than the other methods as shown in Table 4.11. The other feature of proposed method is that this system can learn effectively by using only 2000 samples which is very little when compared to previous methods. Data collection is very difficult and expensive procedure so that the system that learns faster by using less number of data sample, is more practical from others. The confusion matrix notation is used to present results in a more detailed fashion and to be more understandable. The proposed framework results compared with number of methods proposed in [112], also with number of methods proposed by us to detect DDOS attacks such as SVM, and SoftMax classifiers. Table 4.11 illustrates that the proposed framework produces the best results compared with the state-of-the-art methods for this problem.

Table 4.11: Ddos detection methods results comparison.

Methods	Accuracy %
MLP [112]	98.63
Random Forest [112]	98.02
Naïve Bayes [112]	96.91
SVM	97.29
SoftMax	93.14
Proposed Framework	99.60

Output Class	Smurf	400 20.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	100% 0.0%
	UDP Flood	0 0.0%	400 20.0%	0 0.0%	0 0.0%	0 0.0%	100% 0.0%
	SIDDOS	0 0.0%	0 0.0%	400 20.0%	0 0.0%	0 0.0%	100% 0.0%
	HTTP Flood	0 0.0%	0 0.0%	8 0.4%	392 19.6%	0 0.0%	98.0% 2.0%
	Normal	0 0.0%	0 0.0%	0 0.0%	0 0.0%	400 20.0%	100% 0.0%
		100% 0.0%	100% 0.0%	98.0% 2.0%	100% 0.0%	100% 0.0%	99.6% 0.4%
		Smurf	UDP Flood	SIDDOS	HTTP Flood	Normal	
		Target Class					

Figure 4.3: Confusion matrix for ddos detection results.

4.1.2 IDS Attack

In computer security systems, Intrusion Detection Systems (IDS) have become a necessity because of the growing demand in unlawful accesses and attacks. In computer security systems, IDS is a prime part that can be classified as Host-based Intrusion Detection System (HIDS) which superheats a confirmed host or system and Network-based Intrusion detection system (NIDS), which superheats a network of hosts and systems. In this paper, our framework is used to detect IDS attack by using new dataset [113], which consists of 47 features and 10 attack types. We will examine the UNSW-NB15 intrusion dataset in our research, as well as real time captured dataset. This dataset is a hybrid of intrusion data collected from real modernistic normal and abnormal activities of the network traffic. This dataset is newer and more efficient than KDD98, KDDCUP99 and NSLKDD which are the common and older

features datasets because they were generated two decades ago. By following the same procedures in the Figure 3.10, and the tables such as in the DDOS detection procedures, the best parameters were determined as showed in the Tables 4.12 and 4.13 for each Auto-encoders 1 and 2 to find the best parameters that produces the best performance to detect IDS attacks. 10000 data points were used to train and test the system (5000 data used for training and 5000 for testing). Divide half of the data for testing is also a challenging issue that previous studies employ more than %50 data for training. However, in order to reduce overall training time, training data percentage is pulled down. Experimental results for this dataset and configuration is illustrated in Figure 4.4. According to those results, the framework detection rate reaches 99.70% success rate which satisfactory when compared with previous studies, as illustrated in Table 4.14. This proves that even such a small percentage training set is employed for this problem. Satisfactory results can be obtained. Figure 4.4 also demonstrates results based on the corresponding confusion matrix of the output results.

Table 4.12: Auto-encoder 1 parameters set obtained through optimization for ids detection.

Factors	Value
HR	31
ME	250
L2	0.0040
SR	4
SP	0.16

Table 4.13: Auto-encoder 2 parameters set obtained through optimization for ids detection.

Factors	Value
HR	14
ME	80
L2	0.0012
SR	4
SP	0.14

Table 4.14: A comparison amongst ids detection methods results.

Methods	Accuracy %
DT [133]	85.56
LR [133]	83.15
NB [133]	82.07
ANN [133]	81.34
Ramp-KSVCR [133]	93.52
GA-LR [134]	81.42
SSAE-SVM [132]	84.71
SVM	83.16
SoftMax	80.13
Proposed Framework	99.70

Output Class	Exploits	497 9.9%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	100%
	Reconnaissance	0 0.0%	499 10.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	1 0.0%	0 0.0%	0 0.0%	99.8%
	DoS	0 0.0%	1 0.0%	498 10.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	99.8%
	Generic	0 0.0%	0 0.0%	0 0.0%	500 10.0%	0 0.0%	1 0.0%	0 0.0%	3 0.1%	0 0.0%	99.2%
	Fuzzers	0 0.0%	0 0.0%	0 0.0%	0 0.0%	500 10.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	100%
	Analysis	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	499 10.0%	1 0.0%	0 0.0%	0 0.0%	99.8%
	Backdoors	3 0.1%	0 0.0%	2 0.0%	0 0.0%	0 0.0%	0 0.0%	499 10.0%	0 0.0%	0 0.0%	99.0%
	Worms	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	498 10.0%	0 0.0%	100%
	Shellcode	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	497 9.9%	100%
	Normal	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	1 0.0%	0 0.0%	500 10.0%
		99.4%	99.8%	99.6%	100%	100%	99.8%	99.8%	99.6%	99.4%	100%
		0.6%	0.2%	0.4%	0.0%	0.0%	0.2%	0.2%	0.4%	0.6%	0.3%
		Exploits	Reconnaissance	DoS	Generic	Fuzzers	Analysis	Backdoors	Worms	Shellcode	Normal
		Target Class									

Figure 4.4: Confusion matrix for ids detection results.

4.1.3 Epileptic Seizure Recognition

According to the latest results, 1-2% inhabitants of the world suffer from epilepsy which is a neurological trouble [114]. It's distinguished by surprised frequent and evanescent troubles of perception or demean our produce from immoderate coincidence of cortical neural networks. Epileptic Seizure is a neurologic status in which caused from detonation of electrical discharges in the brain. The epileptic seizures mean lineament of epilepsy is recurrent seizures. Observation brain performance over the EEG has become a seizure agent in the detection of epilepsy [115]. There are two kinds of abnormal actions: inter-ictal, abnormal EEG recorded between epileptic crisis and ictal that occurs in the patient's EEG records. The EEG subscription of an inter-ictal action is accidental passing waveforms, as either separated trainer, sharp waves or spike wave complexes [116]. Commonly, veteran

physicians by visual surveying of EEG records for inter-ictal and ictal actions can detect the epilepsy crises. However, visual survey of the huge size of EEG data has business-like disadvantages and weaknesses. Visual search is very time consuming and inactive, essentially in the situation of long size of data [117]. In addition, contention among the physician on the many EEG results in some time lead to individual decision of the analysis and due to the set of inter-ictal spikes morphology. Therefore, computer aided systems are developed to detect blood diseases [118] heart disease recognition [119] and epilepsy detection systems which listed in Table 4.17. Epileptic dataset [120] is used to train and test in the proposed method. Two parts vector matrices are generated with the size of (100×4096) datasets A representing (healthy) and E representing the (epileptic activity condition). A, E divided into two parts, each of them is 50% of the vector matrices then two (50×4096) vector matrices are generated for training and other one for testing. Epileptic Seizure dataset consists of 4096 features by using 2 Auto-encoders, the first one reduces the number of features to 2004 and 103 in second Auto-encoder which means reducing the time consuming. The best parameters for Auto-encoder 1 and Auto-encoder 2 that were obtain from our system are listed in Table 4.15 and Table 4.16. This leads to obtain the best results for Epileptic Seizure Recognition which is represented in Figure 4.6 The proposed method results compared with previous results in Epileptic Seizure Recognition are presented in Table 4.17. Svm, Nlp and SoftMax were implemented by us to obtain results that's compared with our proposed method.

Table 4.15: Auto encoder 1 parameters set obtained through optimization for epileptic seizure recognition.

Factors	Value
HR	2004
ME	350
L2	0.0035
SR	5
SP	0.16

Table 4.16: Auto-encoder 2 parameters set obtained through optimization for epileptic seizure recognition.

Factors	Value
HR	103
ME	160
L2	0.002
SR	4
SP	0.01

The comparison in Table 4.17 shows that there are a number of methods have same accuracies with proposed method such as Tzallas et al. [31] and Srinivasan [33], but our proposed method has a good feature which uses deep learning techniques that give advantage when there are huge numbers of instances of epilepsy data for classification and use only 50% of data in training when other methods used 60%.

Table 4.17: A Comparison of epileptic seizure recognition results.

Methods	Accuracy %
Srinivasan et al. [49]	99.60
Subasi and Ercelebi [50]	92.00
Subasi [51]	94.5
Kannathal et al [52]	92.22
Tzallas et al. [53]	100
Polat et al. [54]	98.72
Acharya et al. [55]	99.00
Acharya et al [56]	99.70
Musa Peker et al.[57]	100
SoftMax	87.13
SVM	92.09
MLP	94.11
Proposed Framework	100

Output Class	Normal	50 50.0%	0 0.0%	100% 0.0%
	Abnormal	0 0.0%	50 50.0%	100% 0.0%
		100% 0.0%	100% 0.0%	100% 0.0%
		Normal	Abnormal	
		Target Class		

Figure 4.5: Confusion matrix for epileptic seizure recognition results.

4.1.4 Handwritten Digit Classification

The proposed framework is finally tested by employing MNSIT dataset which was proposed for handwritten digit classification problem [121]. The framework is trained by using “5000” images that “500” for each example. Each image consisting of “28x28” pixels, meaning there are “784” values for each image when converted to vectors to build the matrixes of vectors. In the second stage, the matrix of arrays become input to the first auto-encoder in which parameters are also optimized by using Taguchi method, as illustrated in Table 4.18. Besides Table 4.19 illustrates the optimized parameters for the second auto-encoder. According to the characteristics of the proposed framework, extracted features from the second auto-encoder are conveyed to the SoftMax layer that classify them into ten separate classes. Overall, the two auto-encoders and SoftMax layer are stacked and trained in a supervised manner. The confusion matrix of the system obtained according to the experimental results is illustrated in Figure 4.7. These results are compared with the state-of-the-art studies regarding this problem and satisfactory results are obtained, as illustrated in Table 4.20.

Table 4.18: Auto-encoder 1 parameters set obtained through optimization for handwritten digit classification.

Factors	Value
HR	100
ME	400
L2	0.004
SR	4
SP	0.15

Table 4.19: Auto-encoder 2 parameters set obtained through optimization for handwritten digit classification.

Factors	Value
HR	50
ME	100
L2	0.002
SR	4
SP	0.1

Output Class	0	499	1	0	0	0	0	2	1	0	0	99.2%
		10.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.8%
	1	0	498	0	0	0	0	0	0	0	0	100%
		0.0%	10.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%
	2	0	0	500	0	0	0	0	0	0	0	100%
		0.0%	0.0%	10.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%
	3	0	0	0	498	0	0	0	0	0	0	100%
		0.0%	0.0%	0.0%	10.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%
	4	0	0	0	0	499	1	0	0	0	0	99.8%
		0.0%	0.0%	0.0%	0.0%	10.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.2%
	5	0	0	0	0	1	499	0	0	0	0	99.8%
		0.0%	0.0%	0.0%	0.0%	0.0%	10.0%	0.0%	0.0%	0.0%	0.0%	0.2%
	6	1	1	0	0	0	0	498	0	0	0	99.6%
		0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	10.0%	0.0%	0.0%	0.0%	0.4%
	7	0	0	0	2	0	0	0	499	0	0	99.6%
		0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	10.0%	0.0%	0.0%	0.4%
	8	0	0	0	0	0	0	0	0	500	0	100%
		0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	10.0%	0.0%	0.0%
	9	0	0	0	0	0	0	0	0	0	500	100%
		0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	10.0%	0.0%
		99.8%	99.6%	100%	99.6%	99.8%	99.8%	99.6%	99.8%	100%	100%	99.8%
		0.2%	0.4%	0.0%	0.4%	0.2%	0.2%	0.4%	0.2%	0.0%	0.0%	0.2%
		0	1	2	3	4	5	6	7	8	9	
		Target Class										

Figure 4.6: Confusion matrix for handwritten digit classification.

Table 4.20: A comparison of handwritten digit classification results.

Reference	Methods	Accuracy %
Anupama Kaushik et al. [122]	J48	70.0
Anupama Kaushik et al. [122]	NaiveBayes	72.65
Anupama Kaushik et al. [122]	SMO	89.95
Olarik Surinta et al. [123]	Hotspot + SVM	92.70
U Ravi Babu et al. [124]	Hotspot + k-NN	96.94
Hinton GE et al. [125]	Deep Belief Network	98.75
LeCun Y et al. [126]	Deep Conv. Net LeNet-5	99.05
Wan L [127]	Deep Conv. Net (dropconnect)	99.43
Zelier MD [128]	Deep Conv. Net (stochastic pooling)	99.53
Goodfellow IJ [129]	Deep Conv. Net (maxout units and dropout)	99.55
Lee CY [130]	Deep Conv. Net (deeply-supervised)	99.61
Proposed Framework	Deep Autoencoder based on Taguchi Method	99.80

4.2 Experimental Results by Using Using Deep Auto-encoder and Energy Spectral Density

4.2.1 Epileptic Seizure Detection

The proposed framework is first adapted to detect epileptic seizure that performs satisfactory results when compared to the previous results obtained by several researchers. The proposed framework is trained and tested by employing the configuration defined in [120]. Dataset consists of two groups, namely A, and E. While ‘A’ represents the healthy case, ‘E’ represents the epileptic activity. Each case (patient) has 4096 data instances, and the proposed framework calculates the energy spectral density, by assigning 4 as the operation period that basically reduces the data into 1024 features. This essentially allows to calculate the ESD equation within an interval of 4 that discretizing the data in a more proper manner. The size of the features and the applied period may be different from one data set to another. An example data instance (patient) is illustrated in Figure 4.7. The extracted features from energy spectral density stage from this case is also shown in Figure 4.8. Afterwards, this features become input to the first auto-encoder, which, by the way, converts input size from 1024 to 500. This operation reduces the dimension of the data to the half and the second Autoencoder converts this input from 500 to 100. The parameters of the auto-encoders presented in the Table 4.21.

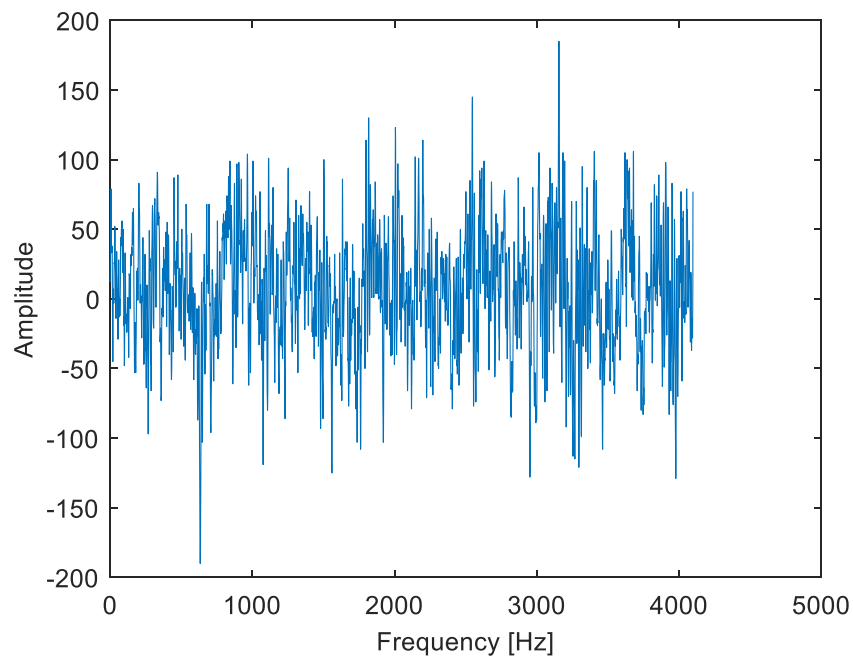


Figure 4.7: An instance (patient data) from epilepsy dataset.

SoftMax has been used to classify the extracted features obtained from the last Autoencoder which essentially labels them as the patient is epileptic seizure or not. This Autoencoders and the Softmax layer are stacked as previously defined that allows greater expressive power and facilitates hierarchical grouping.

The total patient data for this framework is 200 and the proposed framework employs only 50% of the overall data for the training phase. The testing and training data are selected randomly and the experiments are repeated 5 times and the average performance of these experiments are presented in Table 2.22.

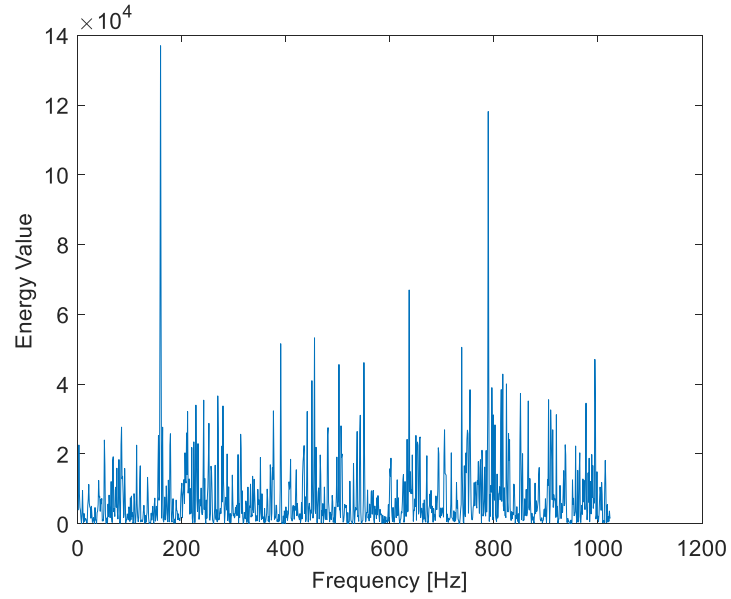


Figure 4.8: Features extracted by using energy spectral density from an ‘instance of epilepsy dataset.

Table 4.21: Auto-encoders factors values for epileptic seizure detection.

Factors	Auto-encoder 1	Auto-encoder 2
Hidden Size (HS)	500	100
Max Epochs (ME)	380	80
L2Weight Regularization (L2)	0.004	0.002
Sparsity Regularization (SR)	3	2
Sparsity Proportion (SP)	0.13	0.1

Table 4.22: Epilepsy seizure detection results.

Results	DSAEs Based ESD
Sensitivity	100
Specificity	100
Precision	100
Negative Predictive Value	100
False Positive Rate	0
False Discovery Rate	0
False Negative Rate	0
Accuracy	100
F1 Score	1
Matthews Correlation Coefficient	1

This framework, employing ESD based feature extraction technique and proposing a stacked Auto-encoder structure, presents satisfactory results when compared with the previous studies as presented in Table 4.23. These studies essentially utilize popular feature extraction and classification methods. However, these methods entail long processing time for handling the epilepsy problem. Besides, previous studies used %60 or more of data for training phase where the proposed framework achieves better or equal results with previous studies by only employing %50 of the data for training.

Table 4.23: Comparisons of epilepsy detection results.

Reference	Methods	Data Selection	Accuracy %
Srinivasan et al. [49]	Time–frequency domain feature-Recurrent neural network	50% Train-50% Test	99.60
Subasi and Ercelebi [50]	WT+ANN	60% Train-40% Test	92.00
Subasi [51]	Discrete WT-Mixture of expert model	60% Train-40% Test	94.5
Kannathal et al [52]	Entropy measures-ANFIS	58% Train-42% Test	92.22
Tzallas et al. [53]	Time–frequency analysis—ANN	60% Train-40% Test	100
Polat et al. [54]	Fast Fourier transform-DT	Tenfold cross validation	98.72
Acharya et al. [55]	WPD-PCA-GMM	60% Train-40% Test	99.00

Acharya et al [56]	Entropies + HOS +Higuchi FD+ Hurst exponent+ FC	60% Train-40% Test	99.70
Musa Peker et al. [57]	DTCWT + CVANN-3	60% Train-40% Test	100
Ahmad Karim et al. [58]	Deep Auto-encoder based Taguchi Method	50% Train-50% Test	100
Proposed Framework	Energy Spectral Density+Deep auto-encoders+ SoftMax	50% Train-50% Test	100

As shown in Table 4.23, DSAEs based ESD presents better results than all studies detailed in [49-52, 54- 56]. On the other hand, despite the studies in [53, 57, 58] also achieves 100% accuracy with the same dataset, they consume more training time. That is because the ESD has a simple mathematical definition compared with conventional feature extraction approaches [58] and is able to extract features in a low computational time.

4.2.2 SPECTF Classification

In this section, the proposed framework is evaluated by employing the SPECTF (Single Proton Emission Computed Tomography) Heart data sets that, presented in [131]. SPECTF Heart dataset represents normal and abnormal classes that involves 267 samples, each of these instances consists of 44 features. There exist 40 instances of each class in the training datasets and the validation datasets includes 172 normal and 15 abnormal samples respectively. An example is illustrated in Figures 4.9 and 4.10. Table 4.25 illustrates the results of the experiment based on previously defined performance evaluation parameters. Then, the parameters of auto-encoders presented in Table 4.24. Besides, the experimental results prove that the proposed framework

achieves quite encouraging results as compared with the previous studies using the SPECTF dataset. These comparison results are presented in Table 4.26.

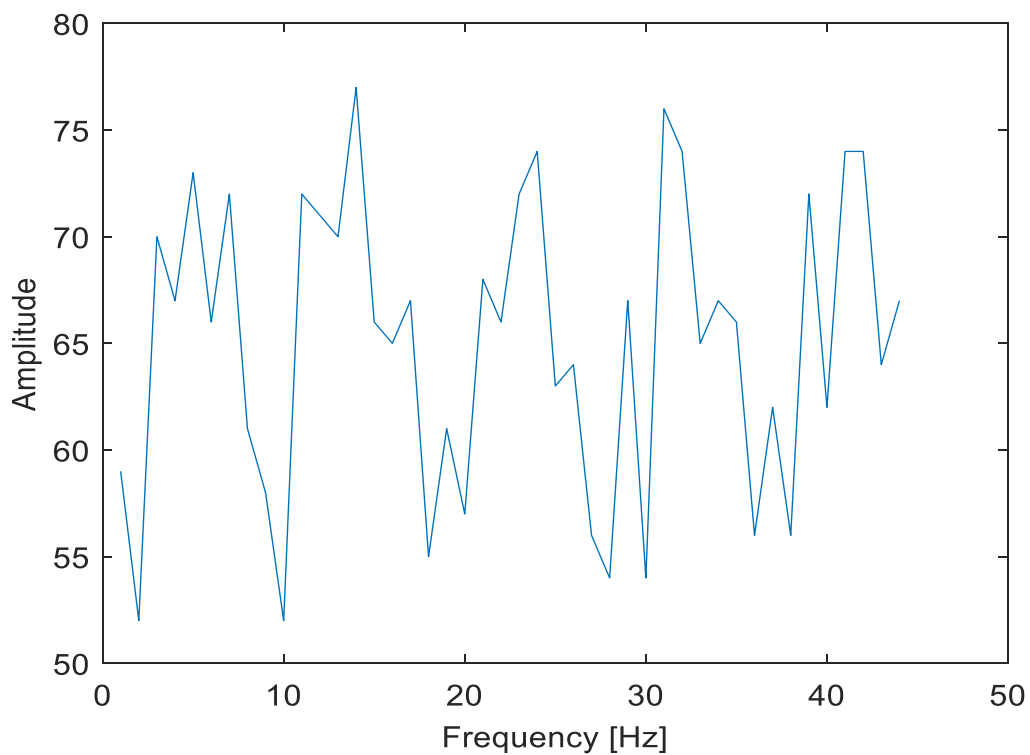


Figure 4.9: An instance (patient data) from spectf dataset.

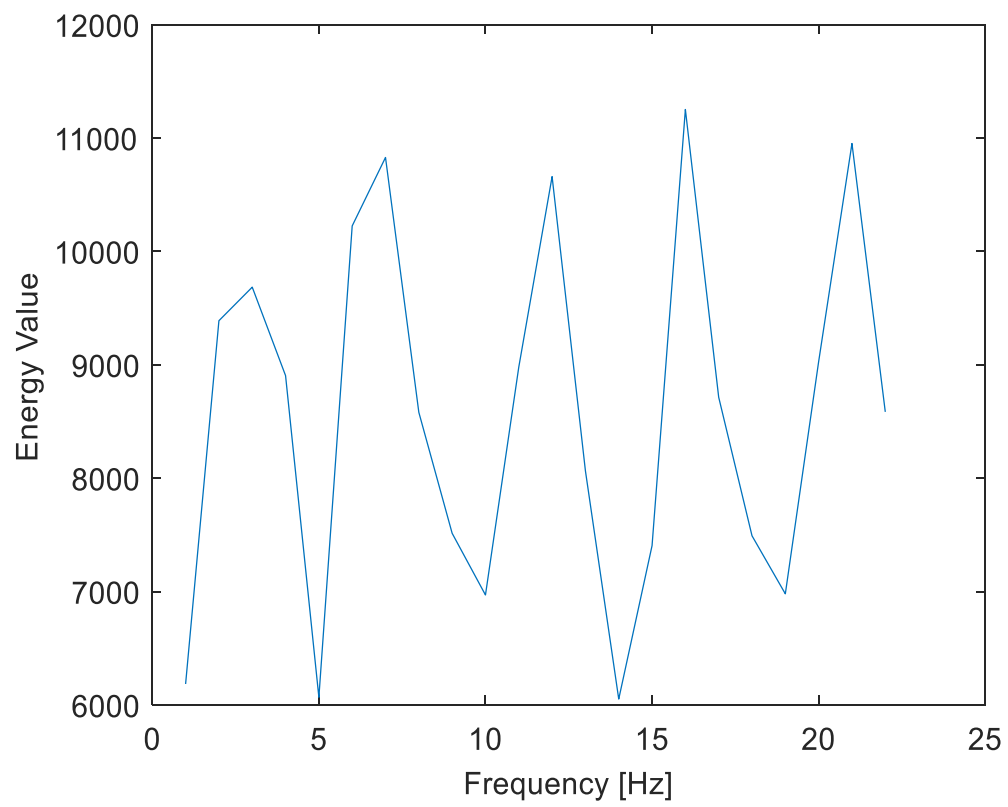


Figure 4.10: Features extracted via esd from spectf dataset.

Table 4.24: Auto-encoders factors values for spectf classification.

Factors	Auto-encoder 1	Auto-encoder 2
Hidden Size (HS)	20	17
Max Epochs (ME)	100	40
L2Weight Regularization (L2)	0.003	0.001
Sparsity Regularization (SR)	2	1
Sparsity Proportion (SP)	0.1	0.1

Table 4.25: Spectf classification results.

Results	DSAEs using ESD
Sensitivity	97.67
Specificity	86.67
Precision	98.82
Negative Predictive Value	76.47
False Positive Rate	0.1333
False Discovery Rate	0.0118
False Negative Rate	0.0233
Accuracy	96.79
F1 Score	0.9825
Matthews Correlation Coefficient	0.7969

Table 4.26: Comparisons of spectf classification results.

References	Methods	ACC %
Myungrae Cha et al. [72]	SVDD	82.7
		95.4
Liu et al. [73]	SVDD-based outlier detection	90
Jing Wei et al. [71]	K2	94.03
	SDBNS	95.59
	ECFBN	95.76
Kumar, R. et al. [74]	mc-MKC	79.9
	mc-SVM	79.1
Cui Li-lin et al. [75]	TCM-IKN N	90
Tian, D. et al. [76]	C-GAME+Johnson+c4.5	84.4
	RMEP+Johnson+c4.5	41
		81.7
M Srinivas [70]	sparsity based dictionary learning+SVM	97.8
Proposed Framework	Energy Spectral Density+ Deep auto-encoders+ SoftMax	96.79

The proposed frameworks achieve better results than all studies discussed in [71- 76]. However, the study proposed in [70] presents %97.8 accuracy results which is better than the proposed approach. This results prove that DSAEs using ESD feature extractor have weakness with low dimensional data. Table 4.26 presents the performance comparison between the proposed technique and other popular studies on SPECTF data.

4.2.3 Diagnosis of Cardiac Arrhythmias

The final experimental section is designed to diagnose the Cardiac Arrhythmias by using a popular dataset, shown in [131]. The dataset used in this part consist of 452 instances from 16 classes that each of them has 279 different attributes. First, a pre-processing operation is performed. For simplification, 278 features are employed and the period is set to 2. Accordingly, 139 features obtained from each data with respect to the ESD technique.

In the first stage, the diagnosis procedure performed by using ESD based on SAEs, which only the 278 of features used because of the ESD need even number of features. In this study, the ESD calculated for each 2 periods which lead to extracted only 139 features as shown in Figures 4.11 and 4.12.

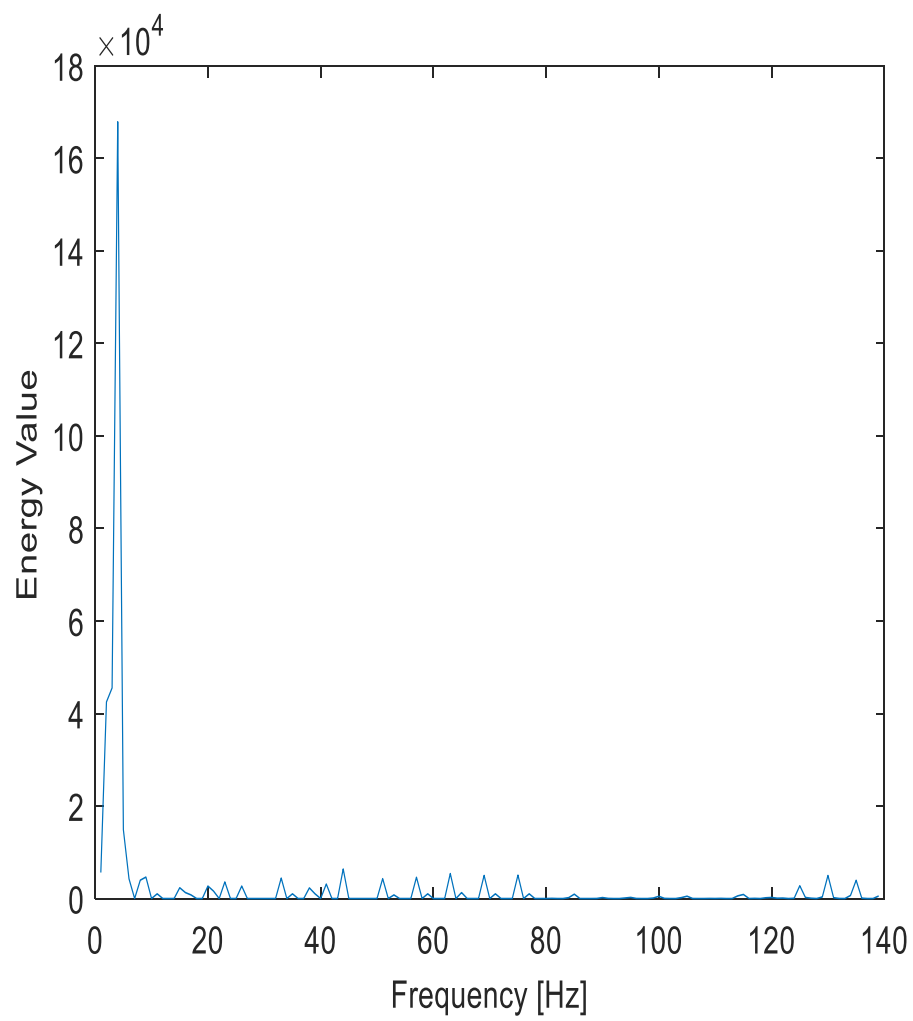


Figure 4.11: An instance (patient data) from cardiac arrhythmias dataset.

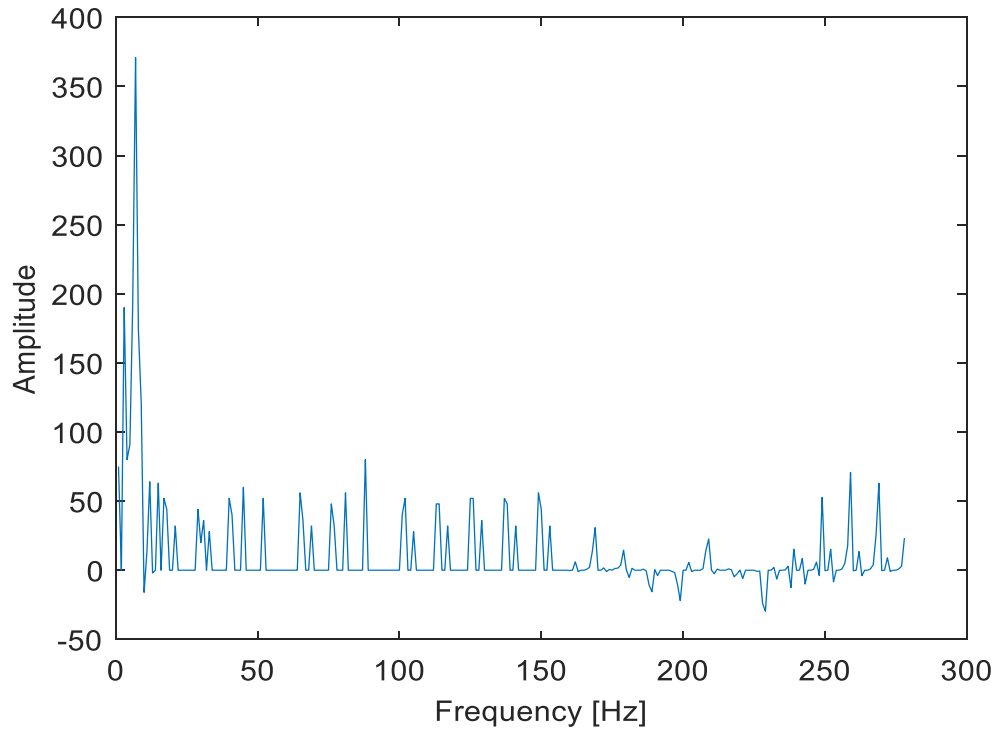


Figure 4.12: Features extracted via esd from cardiac arrhythmias dataset

The extracted features obtained from ESD layer are employed by the first Auto-encoder layer. The output of this Auto-encoder layer generates and input to the second Auto-encoder as it is expected. The features are reduced gradually, from 124 to 120 and the parameters of auto-encoders presented in Table 4.27. This data is then employed by the SoftMax Classifier to be classified.

Table 4.27: Auto-encoders factors values for diagnosis of cardiac arrhythmias.

Factors	Auto-encoder 1	Auto-encoder 2
Hidden Size (HS)	124	120
Max Epochs (ME)	121	103
L2Weight Regularization (L2)	0.003	0.001
Sparsity Regularization (SR)	3	1
Sparsity Proportion (SP)	0.12	0.1

Table 4.28 illustrates the proposed framework performance with respect to the performance evaluation parameters. Table 4.29, on the other hand compares the proposed study with the leading studies using the same dataset.

Table 4.28: Diagnosis of cardiac arrhythmias results.

Results	DAEs Based ESD
Sensitivity	1.000
Specificity	0.9706
Precision	0.9877
Negative Predictive Value	1.000
False Positive Rate	0.0294
False Discovery Rate	0.0123
False Negative Rate	0.0000
Accuracy	0.9912
F1 Score	0.9938
Matthews Correlation Coefficient	0.9791

Table 4.29: Comparison of diagnosis of cardiac arrhythmias results with previous studies.

References	Methods		Classes	ACC %
	Feature Extraction Techniques	Classifiers		
Niazi et al. [59]	Improved F-score and sequential forward search	k-NN	16	73.8
		SVM		68.8
Anam et al. [60]	Wrapper algorithm	MLP	16	78.26
		k-NN		76.6
		SVM		74.4
WM Zuo et al. [61]	Principal Component Analysis	Kernel Difference Weighted k-NN	16	70.66
S. M. Jadhav et al. [62]	-	MLP+Static backpropagation algorithm	16	86.67
Golnaz Sahebi et al. [63]	Multi-population Weighted Intelligent Genetic Algorithm	k-NN (Binary Classification) in normal and abnormal	2	99.70

Anugerah et al. [64]	Best First and CsfSubsetEval	RBF	16	81
Shivajirao et al. [65]	-	Modular Neural Network Model	16	82.22
S. M. Jadhav et al. [66]	-	ANN Models + Static backpropagation algorithm + momentum learning rule	16	86.67
N. Kohli et al. [67]	one-against-all	SVM	16	73.40
A. Özçift et al. [68]	-	A resampling Strategy based Random Forests (RF) ensemble classifier	16	90
Proposed Framework	Energy Spectral Density+ Deep auto-encoders	SoftMax	16	99.12
Proposed Framework	Energy Spectral Density+ Deep auto-encoders	SoftMax	2	99.94

Those studies can be seen in [59-62,64-68]. The results prove the superiority of the proposed framework over other studies with Cardiac Arrhythmias dataset. The only exception being the study proposed in [63]. However, this study employs a binary classifier (normal/ abnormal), whereas the proposed framework and other previous studies employ 16 classes for the classification problem. As it can be seen from the Table, the proposed method is also able to outperform the previous study [63] once the classification layer is converted into a binary classifier.

4.3 Experimental Results by Using Deep Auto-encoder and DWT

The Bonn university dataset [120] used to validate the proposed method. The data consist from 200 row data which 100 of them are normal and other 100 are abnormal (each row consist from 4096 features). Randomly the data divided into two groups training and testing sets 50% for each class. Then, the proposed method trained by using %50 of data which mean 100 case, where 50 of them are normal and other 50 abnormal. The data become input to the DWT which A5 are calculated and the output become only 128 features. The aim of using DWT is to reduce the number of features by extracting important features and Neglecting other redundancy and ineffective features which lead to reduce the execution time and increase accuracy of system. Moreover, the extracted features from DWT become input to the deep auto-encoder. Furthermore, the first auto-encoder reduce the number of features from 128 to 120 by removing only 8 ineffective features. Then, the output of first auto-encoder become input to the second auto-encoder and reduce the number of features from 120 to 100. The purpose of auto-encoders is to reduce the number of features and extracted high level features from input data. The last element in this system is SoftMax which trained as supervised fashion and classify the extracted features from last auto-encoder in two labels normal and abnormal. Then, the proposed method presented 98% accuracy, 94% sensitivity and 98% specificity which are satisfactory results when compared with number of previous studies see Figure 4.13.

Output Class	Target Class			
	Normal	Abnormal		
Normal	47 47.0%	1 1.0%	97.9% 2.1%	
Abnormal	3 3.0%	49 49.0%	94.2% 5.8%	
	94.0% 6.0%	98.0% 2.0%	96.0% 4.0%	

Figure 4.13: Confussion matrix of deep auto-encoder based dwt.

4.4 Experimental Results Using Novel Framework Combining the Deep Auto-Encoder and a Linear Model Based on PSO

4.4.1 Epileptic Seizure Detection

The dataset consists of 200 samples; each sample consisting of 4096 features. The dataset consists of 200 samples, with each sample consisting of 4096 features. The EEG data is split into two groups, namely for training and testing, with each group consisting of 100 examples, 50 of which are normal and the remaining 50 are abnormal. The normal and abnormal cases are shown in Figure 4.14. In the first stage, the first and second auto-encoders automatically extract high-level features from the EEG signal and reduce the number of features to 2007 and 112, respectively. The parameters of the settings of the auto-encoders are presented in Table 4.30. Afterwards, SoftMax classifies the extracted features as normal and abnormal. The linear model is then used to enhance the results, and the parameters of the linear model are estimated by using the PSO algorithm. The linear model parameters are estimated in 30 epochs and produce 0.09 performance, as shown in Figure 4.15. The parameters of PSO are presented in Table 4.31.

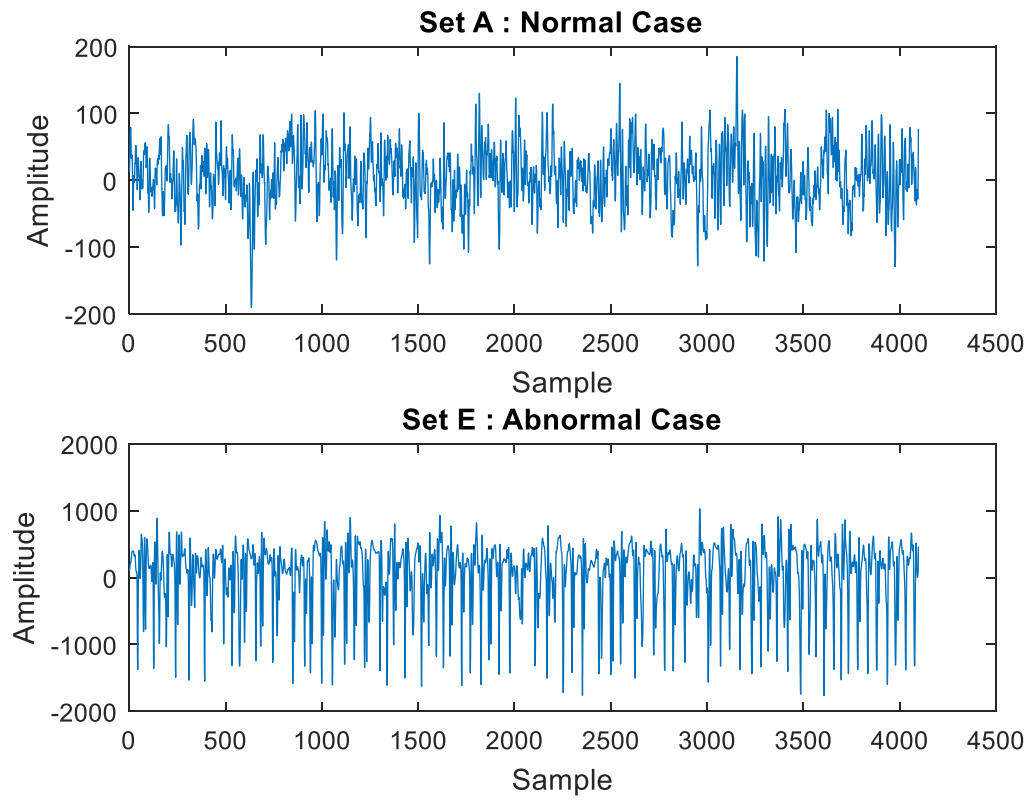


Figure 4.14: Datasets for normal and abnormal cases.

Table 4.30: Auto-encoders parameters of deep auto-encoder and a linear model based on pso.

Factors	Auto-encoder 1	Auto-encoder 2
Hidden Size (HS)	2007	112
Max Epochs (ME)	420	110
L2Weight Regularization (L2)	0.004	0.002
Sparsity Regularization (SR)	4	2
Sparsity Proportion (SP)	0.14	0.12

Table 4.31: Pso parameters for epilepsy seizure.

PSO parameters	values
Particle Number	50
Maximum iteration	60
Cognitive parameter	2
Social parameter	2
Minimum of inertia weight	0.9
Maximum of inertia weight	0.2

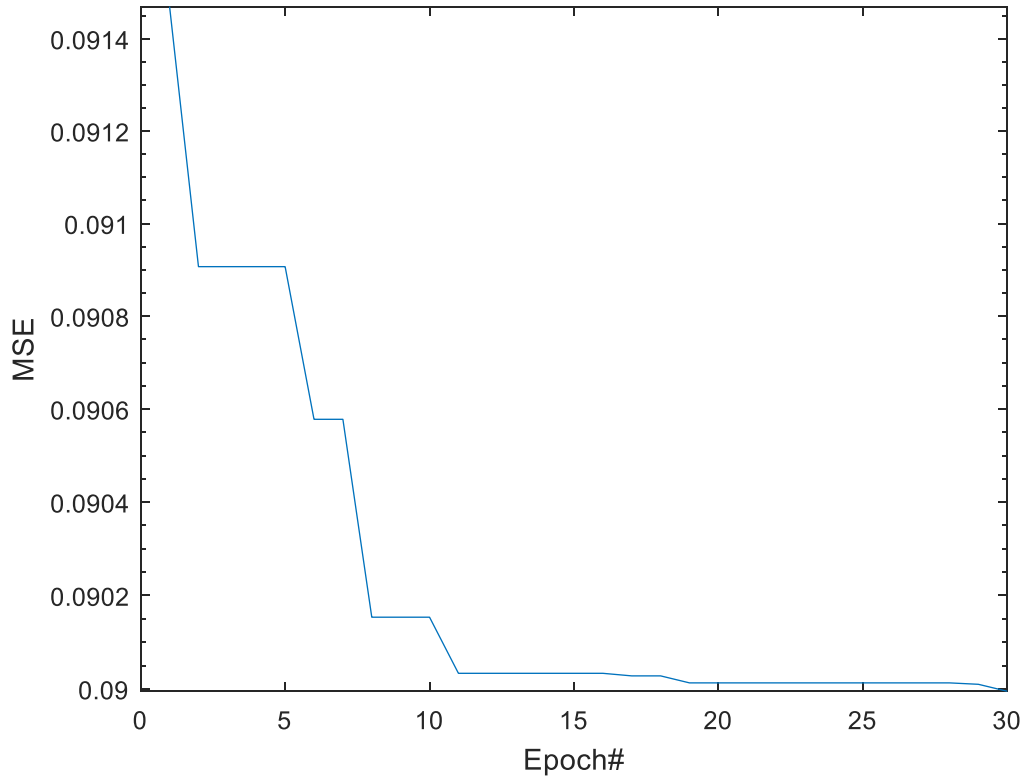


Figure 4.15: The mse for the linear system.

The test process is repeated five times with the same parameters and hidden layer values, but in each implementation the training and test data are randomly selected to avoid overfitting. Then, the average results and the values of parameters are calculated to improve the proposed method's results, as shown in Table 4.32.

The proposed method's results are compared to several studies presented in this field. Then, the previous studies are investigated to show that our proposed method has not been used in the previous studies and shows comparable results, as seen in Table 4.33.

Table 4.32: Epilepsy seizure detection results using post processing.

Results	DAEs Based post processing
Sensitivity	100
Specificity	100
Precision	100
Negative Predictive Value	100
False Positive Rate	0
False Discovery Rate	0
False Negative Rate	0
Accuracy	100
F1 Score	1
Matthews Correlation Coefficient	1

Table 4.33: Comparison proposed framework with previous studies.

Reference	Methods	Accuracy %
Srinivasan et al. [49]	Time–frequency domain feature-Recurrent neural network	99.60
Subasi and Ercelebi [50]	WT+ANN	92.00
Subasi [51]	Discrete WT-Mixture of expert model	94.5
Kannathal et al [52]	Entropy measures-ANFIS	92.22
Tzallas et al. [53]	Time–frequency analysis—ANN	100
Polat et al. [54]	Fast Fourier transform-DT	98.72
Acharya et al. [55]	WPD-PCA-GMM	99.00
Acharya et al [56]	Entropies + HOS +Higuchi FD+ Hurst exponent+ FC	99.70
Musa Peker et al. [57]	DTCWT + CVANN-3	100
Ahmad Karim et al. [58]	Deep Auto-encoder based Taguchi Method	100
Proposed Framework	Deep auto-encoder based post processing	100

The proposed framework presented better results than a number of studies [49- 52,54-56] and presented the same results as other studies with a difference in the complexity and execution time. Peker et al. [57] propose traditional machine techniques which require a long processing time when compared with our proposed framework exactly in high-dimensional features such as epileptic seizure detection. Moreover, in [58] the

deep auto-encoder based Taguchi method, which is a complex system where the parameters are fitted manually when compared with our proposed framework that automatically optimizes the obtained results without needing to repeat experiments manually to obtain the best accuracy.

4.4.2 SPECTF Classification

In this section, the proposed framework is evaluated by employing the SPECTF (Single Proton Emission Computed Tomography) Heart datasets that are presented in [131]. SPECTF Heart dataset represents normal and abnormal classes that involves 267 samples, each of these instances consists of 44 features. There exist 40 instances of each class in the training datasets and the validation datasets includes 172 normal and 15 abnormal samples respectively. The input features example is illustrated in Figures 4.16 and the data features in Auto-encoders 1 and 2 represented in Figures 4.17 and 4.18 respectively. The features in auto-encoders 1 and 2 reduced step by step to 40 and 35 where high level and sensitive features are extracted from input data

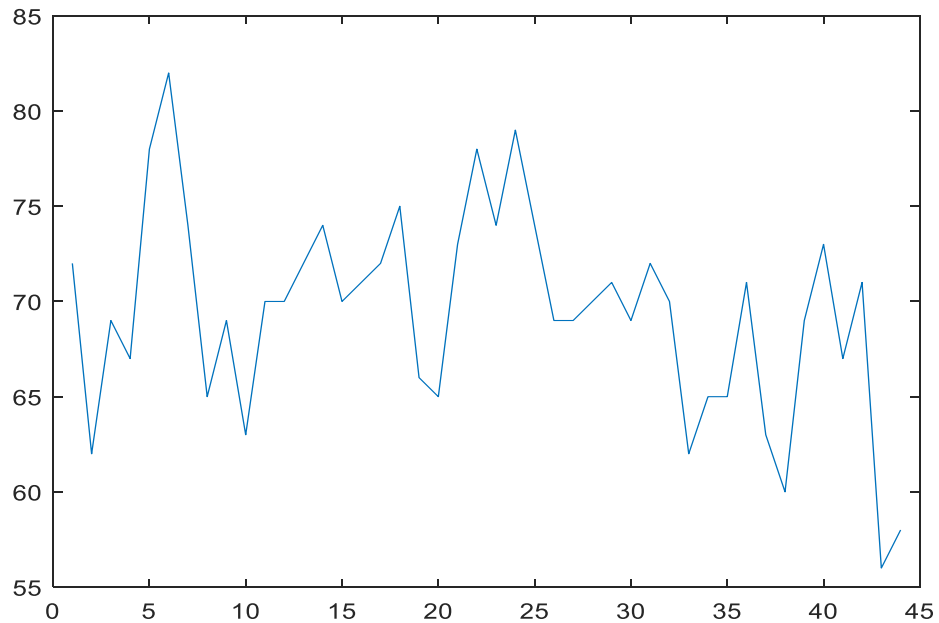


Figure 4.16: An instance (patient data) from spectf dataset.

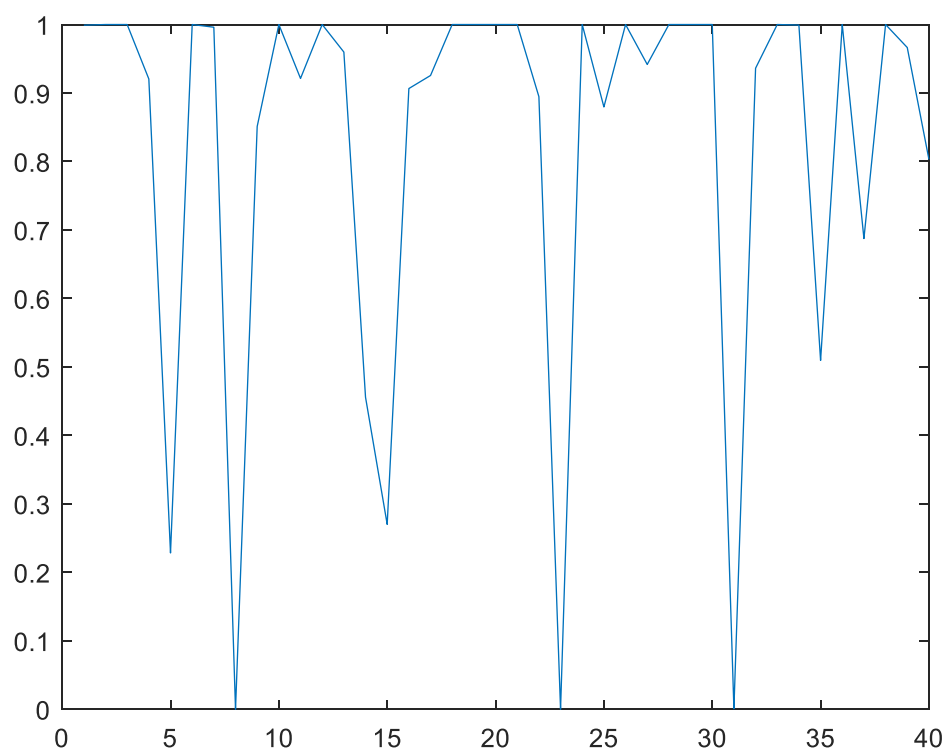


Figure 4.17: Spectf dataset features in auto-encoder 1.

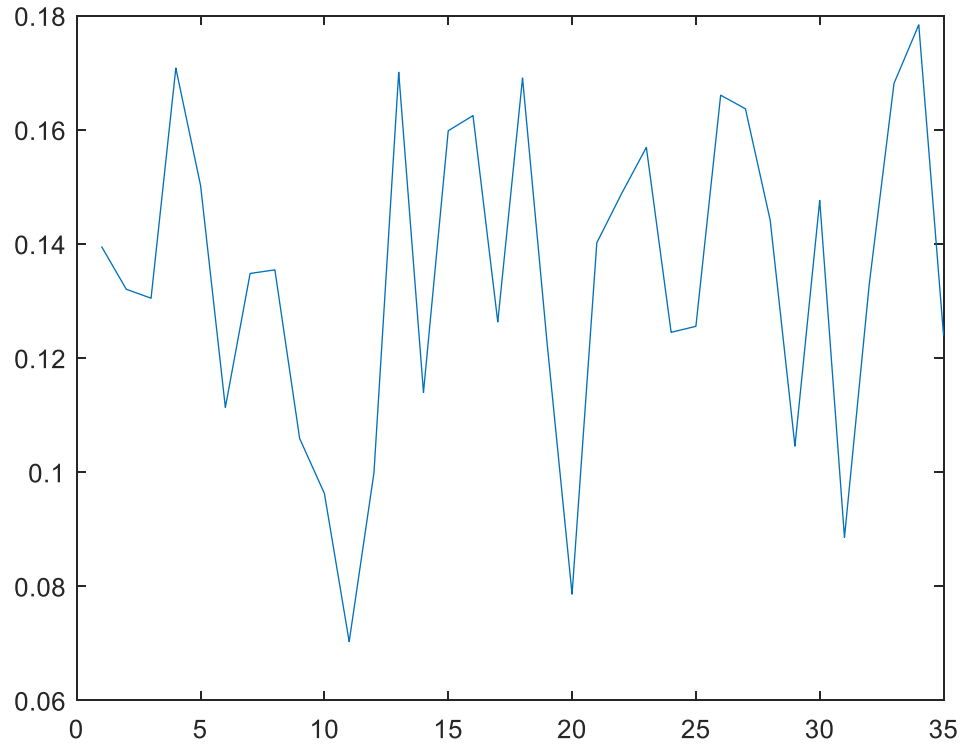


Figure 4.18: Spectf dataset features in auto-encoder 2.

The Figures 4.17 and 4.18 show that the auto-encoders can effectively extracted important features from input data. The important and effective features assist the classifier to produce high accuracy.

Additionally, Table 4.36 illustrates the results of the experiment based on previously defined performance evaluation parameters. Then, the parameters of auto-encoders presented in Table 4.34 and PSO parameters presented in Table 4.35. Besides, the experimental results prove that the proposed framework achieves quite encouraging results as compared with the previous studies using the SPECTF dataset. These comparison results are presented in Table 4.37.

Table 4.34: Auto-encoders factors values for spectf classification using post processing.

Factors	Auto-encoder 1	Auto-encoder 2
Hidden Size (HS)	40	35
Max Epochs (ME)	110	60
L2Weight Regularization (L2)	0.003	0.001
Sparsity Regularization (SR)	2	1
Sparsity Proportion (SP)	0.1	0.1

Table 4.35: Pso parameters for spectf.

PSO parameters	values
Particle Number	40
Maximum iteration	40
Cognitive parameter	2
Social parameter	2
Minimum of inertia weight	0.9
Maximum of inertia weight	0.2

Table 4.36: Spectf classification results using post processing.

Results	DSAEs using PSO
Sensitivity	1.0000
Specificity	0.8750
Precision	0.9884
Negative Predictive Value	1.0000
False Positive Rate	0.1250
False Discovery Rate	0.0116
False Negative Rate	0.0000
Accuracy	0.9893
F1 Score	0.9942
Matthews Correlation Coefficient	0.9300

Table 4.37: Comparisons of spectf classification results using post processing.

References	Methods	ACC %
Myungrae Cha et al. [72]	SVDD	82.7
		95.4
Liu et al. [73]	SVDD-based outlier detection	90
Jing Wei et al. [71]	K2	94.03
	SDBNS	95.59
	ECFBN	95.76
Kumar, R. et al. [74]	mc-MKC	79.9
	mc-SVM	79.1
Cui Li-lin et al. [75]	TCM-IKN N	90
Tian, D. et al. [76]	C-GAME+Johnson+c4.5	84.4
	RMEP+Johnson+c4.5	41
		81.7
M Srinivas [70]	sparsity based dictionary learning+SVM	97.8
Proposed Framework	Deep auto-encoder based post processing	98.93

The proposed framework achieves better results than all studies discussed in [70- 76].

4.4.3 Diagnosis of Cardiac Arrhythmias

This section is designed to diagnose cardiac arrhythmia by using a popular, shown in [131]. The dataset used in this part consist of 452 instances from 16 classes that each of them has 279 different attributes. First, a pre-processing operation is performed. For simplification, 278 features are employed and the period is set to 2. Accordingly, 139 features obtained from each data with respect to the ESD technique.

In the first stage, the diagnosis procedure performed by using ESD based on SAEs, which only the 278 of features used because of the ESD need even number of features. In this study, the ESD calculated for each 2 periods which lead to extracted only 139 features as shown in Figures 4.19, 4.20 and 4.21.

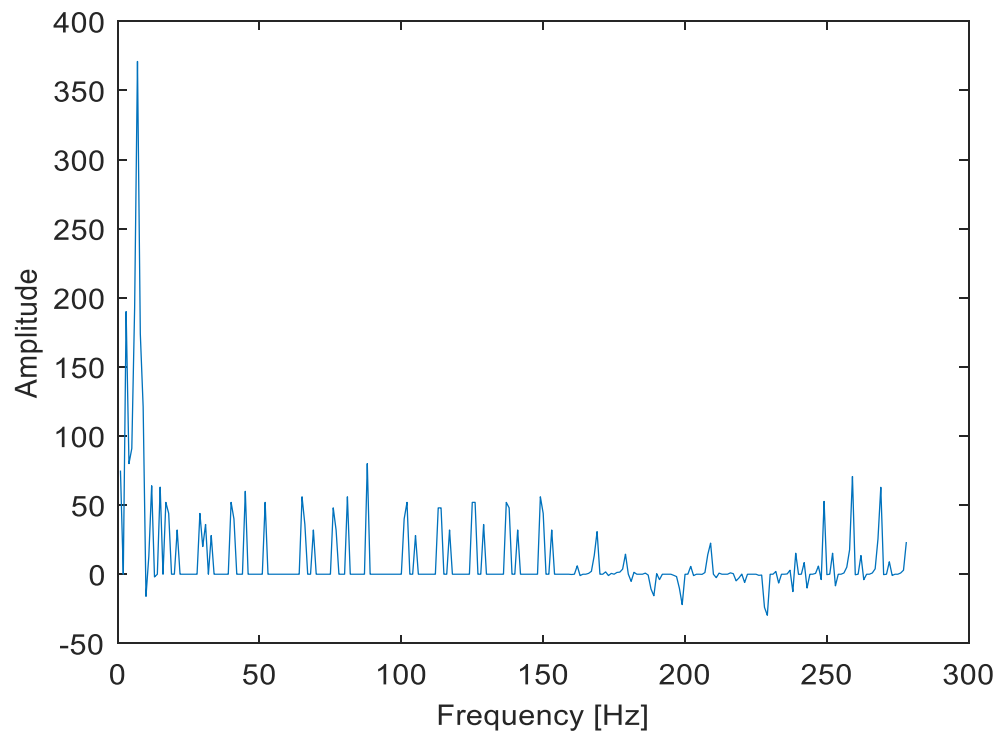


Figure 4.19: An instance (patient data) from cardiac arrhythmias dataset.

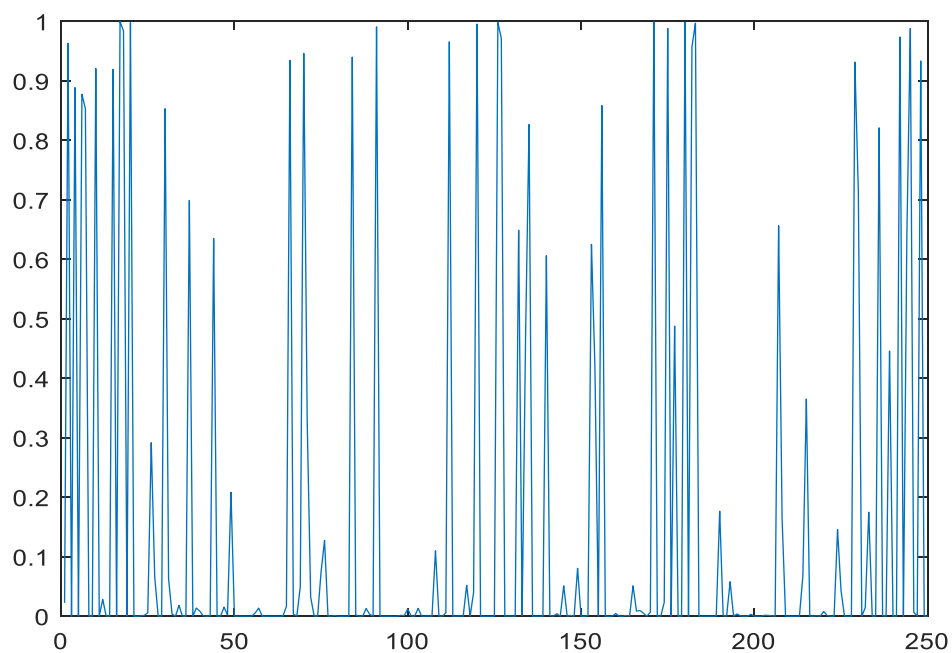


Figure 4.20: Cardiac Arrhythmias Data in Auto-encoder 1.

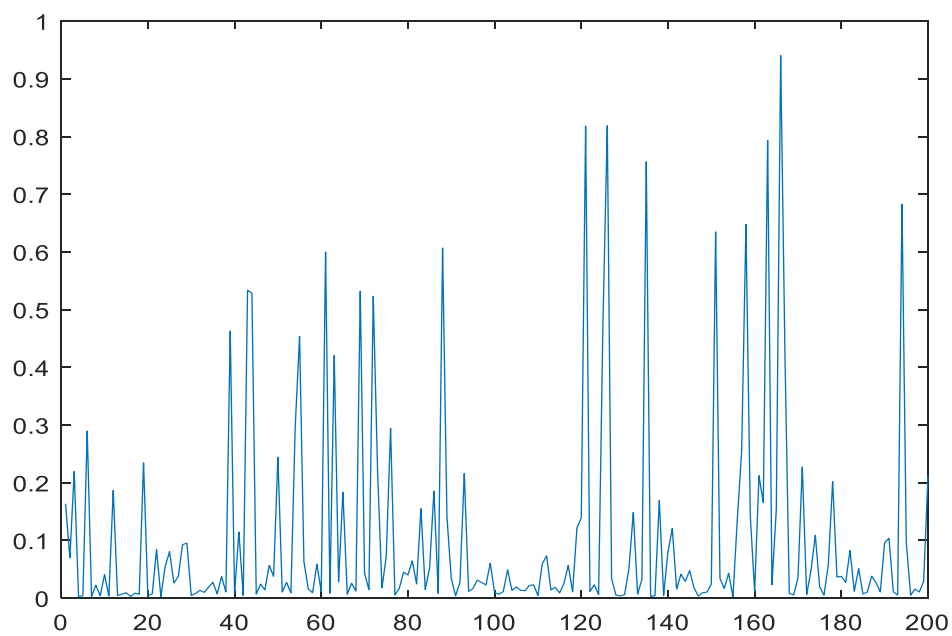


Figure 4.21: Cardiac arrhythmias data in auto-encoder 2.

The first Auto-encoder input layer determined with 278 which represented the number input data. The output of this Auto-encoder1 layer generates and input to the second Auto-encoder as it is expected. The features are reduced gradually, from 124 to 120 and the parameters of auto-encoders presented in Table 4.38. This data is then employed by the SoftMax Classifier to be classified.

Table 4.38: Auto-encoders factors values for diagnosis of cardiac arrhythmias using post processing.

Factors	Auto-encoder 1	Auto-encoder 2
Hidden Size (HS)	250	200
Max Epochs (ME)	130	109
L2Weight Regularization (L2)	0.003	0.001
Sparsity Regularization (SR)	3	1
Sparsity Proportion (SP)	0.12	0.1

Table 4.40 illustrates the proposed framework performance with respect to the performance evaluation parameters. Furthermore, Table 4.39 presented the PSO parameters for Cardiac Arrhythmias. Table 4.41, on the other hand compares the proposed study with the leading studies using the same dataset.

Table 4.39: Pso parameters for cardiac arrhythmias.

PSO parameters	values
Particle Number	60
Maximum iteration	45
Cognitive parameter	2
Social parameter	2
Minimum of inertia weight	0.9
Maximum of inertia weight	0.2

Table 4.40: Diagnosis of cardiac arrhythmias results.

Results	DAEs Based PSO
Sensitivity	0.9959
Specificity	0.9904
Precision	0.9918
Negative Predictive Value	0.9952
False Positive Rate	0.0096
False Discovery Rate	0.0082
False Negative Rate	0.0041
Accuracy	0.9934
F1 Score	0.9939
Matthews Correlation Coefficient	0.9866

Table 4.41: Comparison of diagnosis of cardiac arrhythmias results.

References	Methods		ACC %
	Feature Extraction Techniques	Classifiers	
Niazi et al. [59]	Improved F-score and sequential forward search	k-NN	73.8
		SVM	68.8
Anam et al. [60]	Wrapper algorithm	MLP	78.26
		k-NN	76.6
		SVM	74.4
WM Zuo et al. [61]	Principal Component Analysis	Kernel Difference Weighted k-NN	70.66
S. M. Jadhav et al. [62]	-	MLP+Static backpropagation algorithm	86.67
Golnaz Sahebi et al. [63]	Multi-population Intelligent Genetic Algorithm	k-NN (Binary Classification) in normal and abnormal	99.70

Anugerah et al. [64]	Best First and CsfSubsetEval	RBF	81
Shivajirao et al. [65]	-	Modular Neural Network Model	82.22
S. M. Jadhav et al. [66]	-	ANN Models + Static backpropagation algorithm + momentum learning rule	86.67
N. Kohli et al. [67]	one-against-all	SVM	73.40
A. Özçift et al. [68]	-	A resampling Strategy based Random Forests (RF) ensemble classifier	90
Proposed Framework	Deep auto-encoder based post processing	SoftMax	99.34

Those studies can be seen in [59-62,14,15,64- 68]. The results prove the superiority of the proposed framework over other studies with Cardiac Arrhythmias dataset. The only exception is the study presented in [63] which has better accuracy results than the results of the proposed framework. However, this study performs only a binary classification as either labelling them normal or abnormal. As previously mentioned,

there exists 16 different classes for labelling for the dataset. Accordingly, the proposed method achieves the best result as compared with the other state of the art studies.

4.4.4 Digit Classification

The proposed framework tested also by using MNSIT dataset, this dataset proposed by leeCun in [121] for handwritten digit classifications. The framework trained by using 5000 image 500 for each example. Each image consists from 28*28 pixels, this mean there is 784 values for each image when converted to vectors to built the matrixes of vectors. These features will become input to the first autoencoder, and produce 380 features. Then, these features will have wired to the second auto-encoder and reduced to the 120 features. The output of last autoencoder classified by using SoftMax classifier. In the laset stage, autoencoders and SoftMax stacked and trained in supervised fashion. The proposed framework preduce 99.4 accuracy. Then, linear model based PSO used to optimize the obtained results and preduce 99.5 accuracy see Table 4.36.

Table 4.42: Digit classification using post processing.

Labels	0	1	2	3	4	5	6	7	8	9	Spectivity %
0	499	0	0	0	1	1	2	0	0	0	99.2
1	0	50	0	1	0	0	0	0	0	5	99.8
2	0	0	496		0	0	0	0	0	0	100
3	0	0	1	498	1	0	0	3	0	0	99.0
4	0	0	0	0	495	1	0	1	1	0	99.4
5	0	0	0	0	1	498	0	0	0	0	99.8
6	1	0	0	1	0	0	498	0	0	0	99.6
7	0	0	3	0	1	0	0	496	0	0	99.2
8	0	0	0	0	0	0	0	0	499	1	99.6
9	0	0	0	0	0	0	0	0	0	494	100
Sensitivity %	99.8	100	99.2	99.6	99.0	99.6	99.6	99.2	99.8	98.8	Acc=99.5 %

The Proposed method have satisfactory results when compared with different previous researchs see Table 4.43. Furthermore, the proposed framework also used 50% of data for training data which training and testing data are selected randomly and the experimental repeated 5 time and average accuracy of 5 experiment...

Table 4.43: A comparison of handwritten digit classification results.

Referance	Methods	Acc %
Anupama Kaushik et al. [122]	J48	70.0
Anupama Kaushik et al. [122]	NaiveBayes	72.65
Anupama Kaushik et al. [122]	SMO	89.95
Olarik Surinta et al. [123]	Hotspot + SVM	92.70
U Ravi Babu et al. [124]	Hotspot + k-NN	96.94
Hinton GE et al. [125]	Deep Belief Network	98.75
LeCun Y et al. [126]	Deep Conv. Net LeNet-5	99.05
Wan L [127]	Deep Conv. Net (dropconnect)	99.43
Zelner MD [128]	Deep Conv. Net (stochastic pooling)	99.53
Goodfellow IJ [129]	Deep Conv. Net (maxout units and dropout)	99.55
Lee CY [130]	Deep Conv. Net (deeply-supervised)	99.61
Proposed Framework	Deep Autoencoder based LMPSO	99.50

CHAPTER 5

CONCLUSION AND FUTURE WORK

5.1 Conclusion

In this thesis, several frameworks related to the auto-encoder have been presented. We started by presenting the new Deep learning framework that essentially combines Sparse Auto encoder and Taguchi Method, which is an organized approach for parameter optimization in a reasonable amount of time. Experimental results reveal that applying this method allows the proposed framework to optimize numerous factors and extract more quantitative data from fewer experimental trials simultaneously. This novel framework is tested with different experimental data sets and compared to state-of-the-art methods and studies in terms of overall accuracy. For instance, proposed framework achieves satisfactory results: 99.6% in DDOS Detection, 99.7% for IDS Attack, 100% in Epileptic Seizure Recognition and finally 99.8% precision result for Handwritten Digit Classification Problem. The results verify the validity of the proposed framework.

Furthermore, this thesis proposes a novel framework for medical data processing based on a deep sparse auto-encoder architecture. One of the main contribution of this study is to integrate the energy spectral density approach as a feature extractor into a deep sparse auto-encoder architecture. This mainly speeds up the performance of a DSAE architecture by reducing the dimension of the data, which also prevents employing unnecessary features for the training phase. The procedure first applies a pre-processing step to convert the data into a single dimensional array and calculates the energy density for different periods depending on characteristics of the problem. Afterwards, a stacked auto-encoder architecture with a SoftMax classifier layer are trained in a supervised manner. Three different and popular medical datasets are employed for performance evaluation. The proposed framework presents encouraging results and also provides mostly superior results when it is compared with previous researches, employed the same datasets for the experimental procedures.

The experimental results show that the ESD based DSAE architecture is both effective with datasets having high dimensional data representation such as Epilepsy Seizure Detection and Diagnosis of Cardiac Arrhythmias and it also generates reliable results with dataset having low dimensional data representation such as SPECTF Heart classification.

Moreover, a new framework proposed based on deep auto-encoder and DWT to recognize epileptic seizure. The combining of deep auto-encoder with DWT lead to increase the recognition rate and decrease the execution time. The proposed framework produces 96% accuracy because the DWT remove ineffective features and reduce the execution time because the DWT extracted only 128 features as A5 parameter by using simple mathematic process.

Finally, The PSO based linear system is new idea for increasing the deep stacked auto-encoder accuracy for epilepsy seizure detection. The goal of this system is to classify the EEG signal into two classes (epileptic vs non-epileptic). The proposed system consists from two stages, deep stacked auto-encoders and linear model based PSO. The deep stacked auto-encoders consist from two auto-encoders which extracted features and the output of second auto-encoder wired to the SoftMax. The two auto-encoders and SoftMax stacked and trained in supervised fashion using backpropagation technique to improve the classification accuracy.

In the second stage, the linear model used to optimize the predication of the deep stacked auto-encoders, and the parameters of the linear model estimated by using PSO.

Then, the proposed framework presented satisfactory results when compared with number of famous studies proposed in this field by producing 100% accuracy.

5.2 Future Work

In the end of this study, number of ideas and questions proposed by the researcher which lead to improve the results or produce new results.

As well as, authors are encouraged to improve overall performance of this architecture for more complex problems such as 3D image processing and real-time robotic system. Accordingly, different heuristic optimization algorithm, including genetic algorithms, particle swarm optimization or colony optimization algorithms will be used to estimate auto encoder parameters and compared with the Taguchi Method as future works. It is also noticed that the proposed architecture can also be employed for comprehensive recognition and estimation problems, including gesture recognition, URL reputation, SMS spam collection etc.

We recommend to use preprocessing technique (filters, unsupervised learning etc..) to speed up and enhance the auto-encoders performance.

Additionally, nonlinear and dynamic linear model systems can be proposed as post-processing technique for enhancing the classification accuracy of classification models. Moreover, other optimization algorithms can be used instead of PSO such as genetic algorithm, ant colony optimization algorithm and bat algorithm, other classifications model can be combined with linear and non-linear models such as: Support vector machine, Bayes naïve and decision tree. The proposed framework can have used to classify any data by changing only input features, hidden neurons and output classes easily.

REFERENCES

- [1] Chandra, B., and Sharma, R. K., “Deep learning with adaptive learning rate using laplacian score,” *Expert Syst. Appl.*, vol. 63, pp. 1–7, 2016.
- [2] Ding, S., Zhang, N., Xu, X., Guo, L., and Zhang, J., “Deep extreme learning machine and its application in eeg classification,” *Mathematical Problems in Engineering*, vol. 2015, pp. 1–11, 2015, article ID 129021.
- [3] Urban, G. *et al.*, “Do Deep Convolutional Nets Really Need to be Deep and Convolutional?,” 2016.
- [4] D. T. Grozdic and S. T. Jovicic, "Whispered Speech Recognition Using Deep Denoising Autoencoder and Inverse Filtering," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 25, no. 12, pp. 2313-2322, 2017.
- [5] Castro, L., Wasserman, E. A., and Lauffer, M., “Unsupervised learning of complex associations in an animal model,” *Cognition*, vol. 173, no. October 2017, pp. 28–33, 2018.
- [6] Xu, P., Peng, H., and Huang, T., “Unsupervised learning of mixture regression models for longitudinal data,” *Comput. Stat. Data Anal.*, vol. 125, pp. 44–56, 2018.
- [7] Tryfos, P., “Factor analysis,” *Methods Bus. Anal. Forecast.*, p. 592, 1998.
- [8] Škrbić, B. and Durišić-Mladenović, N., “Principal component analysis for soil contamination with organochlorine compounds,” *Chemosphere*, vol. 68, no. 11, pp. 2144–2152, 2007.
- [9] Shabani, S. and Norouzi, Y., “Speech recognition using Principal Components Analysis and Neural Networks,” *2016 IEEE 8th Int. Conf. Intell. Syst.*, pp. 90–

95, 2016.

- [10] Tan, P.-N., Steinbach, M., and Kumar, V., “Chap 8 : Cluster Analysis: Basic Concepts and Algorithms,” *Introd. to Data Min.*, p. Chapter 8, 2005.
- [11] Sardar, T. H. and Ansari, Z., “An analysis of MapReduce efficiency in document clustering using parallel K-means algorithm,” *Futur. Comput. Informatics J.*, pp. 1–10, 2018.
- [12] Mohadab, M. El, Bouikhalene, B., and Safi, S., “Predicting Rank for Scientific Research Papers using Supervised Learning,” *Appl. Comput. Informatics*, 2018.
- [13] Shenfield, A., Day, D., and Ayesh, A., “Intelligent intrusion detection systems using artificial neural networks,” *ICT Express*, vol. 4, no. 2, pp. 95–99, 2018.
- [14] Khoshroo, A., Emrouznejad, A., Ghaffarizadeh, A., Kasraei, M., and Omid, M., “Sensitivity analysis of energy inputs in crop production using artificial neural networks,” *J. Clean. Prod.*, vol. 197, pp. 992–998, 2018.
- [15] Denis, M., Wan, L., Fatemi, M., and Alizad, A., “Ultrasound Characterization of Bone Demineralization Using a Support Vector Machine,” *Ultrasound Med. Biol.*, vol. 44, no. 3, pp. 714–725, 2018.
- [16] Subasi, O. *et al.*, “Exploring the capabilities of support vector machines in detecting silent data corruptions,” *Sustain. Comput. Informatics Syst.*, 2018.
- [17] Zhou, Y. and Qiu, G., “Random forest for label ranking,” *Expert Syst. Appl.*, vol. 112, pp. 99–109, 2018.
- [18] Brownlee, J., “Summary of Data Mining Algorithms,” 2016. [Online]. Available: <http://web.utk.edu/~wfeng1/html/algsummary.html>.

- [19] Athreya, S. and Venkatesh, Y. D., “Application Of Taguchi Method For Optimization Of Process Parameters In Improving The Surface Roughness Of Lathe Facing Operation,” vol. 1, no. 3, pp. 13–19, 2012.
- [20] Cui, F., Su, Y., Xu, S., Liu, F., and Yao, G., “Optimization of the Physical and Mechanical Properties of a Spline Surface Fabricated by High-Speed Cold Roll Beating Based on Taguchi Theory,” vol. 2018, 2018.
- [21] Peker, M., “A new approach for automatic sleep scoring: Combining Taguchi based complex-valued neural network and complex wavelet transform,” *Comput. Methods Programs Biomed.*, vol. 129, pp. 203–216, 2016.
- [22] Hsu, Q. C. and Do, A. T., “Minimum porosity formation in pressure die casting by taguchi method,” *Math. Probl. Eng.*, vol. 2013, 2013.
- [23] Peker, M., Şen, B., and Kumru, P. Y., “An efficient solving of the traveling salesman problem: The ant colony system having parameters optimized by the Taguchi method,” *Turkish J. Electr. Eng. Comput. Sci.*, vol. 21, no. SUPPL. 1, pp. 2015–2036, 2013.
- [24] Nandhini, M., Suchithra, B., Saravana-thamizhan, R., and Prakash, D. G., “Optimization of parameters for dye removal by electro- -oxidation using Taguchi Design,” *J. Electrochem. Sci. Eng.*, vol. 4, no. 4, pp. 227–234, 2014.
- [25] Ivanović, L., Stojanović, B., Blagojević, J., Bogdanović, G., and Marinković, A., “Analysis of the flow rate and the volumetric efficiency of the trochoidal pump by application of taguchi method,” *Teh. Vjesn.*, vol. 24, pp. 265–270, 2017.
- [26] Kaveh, A., *Advances in metaheuristic algorithms for optimal design of structures*, Second edition. Springer, 2016.

- [27] R. Harman, "A very brief introduction to particle swarm optimization," pp. 1–4, 1995.
- [28] Rini, D. P., Shamsuddin, S. M, and Yuhaniz, S. S., "Particle Swarm Optimization: Technique, System and Challenges," *Int. J. Comput. Appl.*, vol. 14, no. 1, pp. 19–27, 2011.
- [29] Oludare, O., Stephen, O., Ayodele, O., and Temitayo, F., "An optimized feature selection technique for email classification," *Int. J. Sci. Technol. Res.*, vol. 3, no. 10, pp. 286–293, 2014.
- [30] Lore, K. G., Akintayo, A., and Sarkar, S., "LLNet: A deep autoencoder approach to natural low-light image enhancement," *Pattern Recognit.*, vol. 61, pp. 650–662, 2017.
- [31] Sun, K., Zhang, J., Zhang, C., and Hu, J., "Generalized extreme learning machine autoencoder and a new deep neural network," *Neurocomputing*, vol. 230, no. November 2016, pp. 374–381, 2017.
- [32] Xiong, Y. and Zuo, R., "Recognition of geochemical anomalies using a deep autoencoder network," *Comput. Geosci.*, vol. 86, pp. 75–82, 2016.
- [33] Burgoon, L. D., "Autoencoder Predicting Estrogenic Chemical Substances (APECS): An improved approach for screening potentially estrogenic chemicals using in vitro assays and deep learning," *Comput. Toxicol.*, vol. 2, pp. 45–49, 2017.
- [34] Hong, C., Yu, J., Wan, J., Tao, D., and Wang, M., "Multimodal Deep Autoencoder for Human Pose Recovery," *IEEE Trans. Image Process.*, vol. 24, no. 12, pp. 5659–5670, 2015.
- [35] Song, T. H., Sanchez, V., Eidaly, H., and Rajpoot, N. M., "Hybrid deep autoencoder with Curvature Gaussian for detection of various types of cells in

- bone marrow trephine biopsy images,” *Proc. - Int. Symp. Biomed. Imaging*, pp. 1040–1043, 2017.
- [36] Suzuki, Y. and Ozaki, T., “Stacked denoising autoencoder-based deep collaborative filtering using the change of similarity,” *Proc. - 31st IEEE Int. Conf. Adv. Inf. Netw. Appl. Work. WAINA 2017*, pp. 498–502, 2017.
- [37] Zhang, Y. D., Hou, X. X., Lv, Y. D., Chen, H., Zhang, Y., and Wang, S. H., “Sparse autoencoder based deep neural network for voxelwise detection of cerebral microbleed,” *Proc. Int. Conf. Parallel Distrib. Syst. - ICPADS*, vol. 164, pp. 1229–1232, 2017.
- [38] Athreya, S. and Venkatesh, Y. D., “Application Of Taguchi Method For Facing Optimization Of Process Parameters In Improving The Surface Roughness Of Lathe Operation,” vol. 1, no. 3, pp. 13–19, 2012.
- [39] Huang, M. L., Hung, Y. H., Lee, W. M., Li, R. K., and Jiang, B. R., “SVM-RFE based feature selection and Taguchi parameters optimization for multiclass SVM Classifier,” *Sci. World J.*, vol. 2014, 2014.
- [40] Ko, D. C., Kim, D. H., and Kim, B. M., “Application of artificial neural network and Taguchi method to preform design in metal forming considering workability,” *Int. J. Mach. Tools Manuf.*, vol. 39, no. 5, pp. 771–785, 1999.
- [41] Wang, H., Geng, Q., and Qiao, Z., “Parameter Tuning of Particle Swarm Optimization by Using Taguchi Method and Its Application to Motor Design 20130415,” pp. 722–726, 2014.
- [42] Dhuria, G. K., Singh, R., and Batish, A., “Application of a hybrid Taguchi-entropy weight-based GRA method to optimize and neural network approach to predict the machining responses in ultrasonic machining of Ti–6Al–4V,” *J. Brazilian Soc. Mech. Sci. Eng.*, vol. 39, no. 7, pp. 2619–2634, 2017.

- [43] Yang, H.-F., Dillon, T. S., and Chen, Y.-P. P., "Optimized Structure of the Traffic Flow Forecasting Model With a Deep Learning Approach," *IEEE Trans. Neural Networks Learn. Syst.*, vol. 28, no. 10, pp. 1–11, 2016.
- [44] Wang, L., *et al.*, "Parameter optimization of a four-legged robot to improve motion trajectory accuracy using signal-to-noise ratio theory," *Robot. Comput. Integr. Manuf.*, vol. 51, no. September 2016, pp. 85– 96, 2018.
- [45] Ibrahim, S., Djemal, R., and Alsuwailem, A., "Electroencephalography (EEG) signal processing for epilepsy and autism spectrum disorder diagnosis," *Biocybern. Biomed. Eng.*, vol. 38, no. 1, pp. 16–26, 2018.
- [46] Yavuz, E., Kasapbaşı, M. C., Eyüpoğlu, C., and Yazıcı, R., "An epileptic seizure detection system based on cepstral analysis and generalized regression neural network," *Biocybern. Biomed. Eng.*, vol. 38, no. 2, pp. 201–216, 2018.
- [47] Djemili, R., Bourouba, H., and Korba, M. C. A., "Application of empirical mode decomposition and artificial neural network for the classification of normal and epileptic EEG signals," *Biocybern. Biomed. Eng.*, vol. 36, no. 1, pp. 285–291, 2016.
- [48] Martinez-del-Rincon, J. *et al.*, "Non-linear classifiers applied to EEG analysis for epilepsy seizure detection," *Expert Syst. Appl.*, vol. 86, pp. 99–112, 2017.
- [49] Srinivasan, V., Eswaran, C., and Sriraam, A. N., "Artificial neural network based epileptic detection using time-domain and frequency-domain features," *J. Med. Syst.*, vol. 29, no. 6, pp. 647–660, 2005.
- [50] Subasi, A. and Erçelebi, E., "Classification of EEG signals using neural network and logistic regression," *Comput. Methods Programs Biomed.*, vol. 78, no. 2, pp. 87–99, 2005.

- [51] Subasi, A., “EEG signal classification using wavelet feature extraction and a mixture of expert model,” *Expert Syst. Appl.*, vol. 32, no. 4, pp. 1084–1093, 2007.
- [52] Kannathal, N., Choo, M. L., Acharya, U. R., and Sadasivan, P. K., “Entropies for detection of epilepsy in EEG,” *Comput. Methods Programs Biomed.*, vol. 80, no. 3, pp. 187–194, 2005.
- [53] Tzallas, A. T., Tsipouras, M. G., and Fotiadis, D. I., “Automatic seizure detection based on time-frequency analysis and artificial neural networks,” *Comput. Intell. Neurosci.*, vol. 2007, 2007.
- [54] Polat, K. and Güneş, S., “Classification of epileptiform EEG using a hybrid system based on decision tree classifier and fast Fourier transform,” *Appl. Math. Comput.*, vol. 187, no. 2, pp. 1017–1026, 2007.
- [55] Acharya U., Sree, S. V., A. P. C. Alvin, and J. S. Suri, “Use of principal component analysis for automatic classification of epileptic EEG activities in wavelet framework,” *Expert Syst. Appl.*, vol. 39, no. 10, pp. 9072–9078, 2012.
- [56] Acharya, U. R., Sree, S. V., Ang, P. C. A., Yanti, R., and Suri, J. S., “Application of Non-Linear and Wavelet Based Features for the Automated Identification of Epileptic Eeg Signals,” *Int. J. Neural Syst.*, vol. 22, no. 2, p. 1250002, 2012.
- [57] Peker, M., Sen, B., and Delen, D., “A Novel Method for Automated Diagnosis of Epilepsy using Complex-Valued Classifiers,” *IEEE J. Biomed. Heal. informatics*, vol. 2194, no. c, pp. 1–11, 2015.
- [58] Karim, A. M., Güzel, M. S., Tolun, M. R., Kaya, H., and Çelebi, F. V., “A New Generalized Deep Learning Framework Combining Sparse Auto-encoder and

Taguchi Method for Novel Data Classification and Processing,” pp. 1–22

- [59] Ahmed, K., Nia, K., Khan, S. A., and Shaukat, A., “Identifying Best Feature Subset for Cardiac Arrhythmia Classification,” pp. 494–499, 2015.
- [60] Mustaqeem, A., Anwar, S. M., Majid, M., and Khan, A. R., “Wrapper method for feature selection to classify cardiac arrhythmia,” *Proc. Annu. Int. Conf. IEEE Eng. Med. Biol. Soc. EMBS*, pp. 3656–3659, 2017.
- [61] Zuo, W. M., Lu, W. G., Wang, K. Q., and Zhang, H., “Diagnosis of cardiac arrhythmia using kernel difference weighted KNN classifier,” *Comput. Cardiol.*, vol. 35, pp. 253–256, 2008.
- [62] Jadhav, S. M., Nalbalwar, S. L., and Ghatol, a., “Artificial Neural Network based cardiac arrhythmia classification using ECG signal data,” *Electron. Inf. Eng. (ICEIE), 2010 Int. Conf.*, vol. 1, no. Iceie, pp. 228–231, 2010.
- [63] Sahebi, G., Majd, A., Ebrahimi, M., Plosila, J., and Tenhunen, H., “A reliable weighted feature selection for auto medical diagnosis,” *Proc. - 2017 IEEE 15th Int. Conf. Ind. Informatics, INDIN 2017*, pp. 985–991, 2017.
- [64] Persada, A. G., Setiawan, N. A., and Nugroho, H. A., “Comparative study of attribute reduction on arrhythmia classification dataset,” *Proc. - 2013 Int. Conf. Inf. Technol. Electr. Eng. "Intelligent Green Technol. Sustain. Dev. ICITEE 2013*, pp. 68–72, 2013.
- [65] Jadhav, S. M., Nalbalwar, S. L., Ghatol, A. A., and Advisor, T., “ECG Arrhythmia Classification using Modular Neural Network Model,” *2010 IEEE EMBS Conference on Biomedical Engineering and Sciences (IECBES)*.
- [66] Jadhav, S. M., Nalbalwar, S. L., and Ghatol, A. A., “Artificial Neural Network Based Cardiac Arrhythmia Disease Diagnosis,” *2011 Int. Conf. Process Autom.*

Control Comput., pp. 1–6, 2011.

- [67] Kohli, N., Verma, N. K., and Roy, A., “SVM based methods for arrhythmia classification in ECG,” *2010 Int. Conf. Comput. Commun. Technol. ICCCT-2010*, pp. 486–490, 2010.
- [68] Özçift, A., “Random forests ensemble classifier trained with data resampling strategy to improve cardiac arrhythmia diagnosis,” *Comput. Biol. Med.*, vol. 41, no. 5, pp. 265–271, 2011.
- [69] Saraçoğlu, R., “Hidden Markov model-based classification of heart valve disease with PCA for dimension reduction,” *Eng. Appl. Artif. Intell.*, vol. 25, no. 7, pp. 1523–1528, 2012.
- [70] Srinivas, M., Bharath, R., Rajalakshmi, P., and Mohan, C. K., “Multi-level classification: A generic classification method for medical datasets,” *2015 17th Int. Conf. E-Health Networking, Appl. Serv. Heal. 2015*, pp. 262–267, 2016.
- [71] Wei, J., Yu, H., Lu, Y. Q., and Wang, J., “The Research of Bayesian Method from Small Sample of High-dimensional Dataset in Poison Identification,” pp. 705–709.
- [72] Cha, M., Kim, J. S., and Baek, J. G., “Density weighted support vector data description,” *Expert Syst. Appl.*, vol. 41, no. 7, pp. 3343–3350, 2014.
- [73] Liu, B., Xiao, Y., Cao, L., Hao, Z., and Deng, F., “SVDD-based outlier detection on uncertain data,” *Knowl. Inf. Syst.*, vol. 34, no. 3, pp. 597–618, 2013.
- [74] Kumar, R., Chen, T., Hardt, M., Beymer, D., Brannon, K., and Syeda-Mahmood, T., “Multiple Kernel Completion and its application to cardiac disease discrimination,” *Proc. - Int. Symp. Biomed. Imaging*, pp. 764–767, 2013.

- [75] Li-lin, C., Hai-chao, Z., Lin-ke, Z., and Rui-peng, L., “Improved k Nearest Neighbors Transductive Confidence Machine for Pattern Recognition,” vol. 3, no. Iccda, pp. 172–176, 2010.
- [76] Tian, D., Zeng, X. J., and Keane, J., “Core-generating approximate minimum entropy discretization for rough set feature selection in pattern classification,” *Int. J. Approx. Reason.*, vol. 52, no. 6, pp. 863–880, 2011.
- [77] Pushkar, B. and Paul, M., “Early Diagnosis of Alzheimer ’ s Disease : A Multi - class Deep Learning Framework with Modified k- sparse Autoencoder Classification,” pp. 1–6, 2016.
- [78] Tong, H., Liu, B., and Wang, S., “Software Defect Prediction Using Stacked Denoising Autoencoders and Two-stage Ensemble Learning,” *Inf. Softw. Technol.*, no. November, pp. 1–18, 2017.
- [79] Xr, L. K. *et al.*, “Using stacked denoising autoencoder for the student droupout predication,” pp. 483–488, 2017.
- [80] Xiong, Y. and Zuo, R., “Recognition of geochemical anomalies using a deep autoencoder network,” *Comput. Geosci.*, vol. 86, pp. 75–82, 2016.
- [81] Han, T., Hao, K., Ding, Y., and Tang, X., “Neurocomputing A sparse autoencoder compressed sensing method for acquiring the pressure array information of clothing,” *Neurocomputing*, vol. 275, pp. 1500–1510, 2018.
- [82] Salaken, S. M., Khosravi, A., Khatami, A., Nahavandi, S., and Hosen, M. A., “Lung Cancer Classification Using Deep Learned Features on Low Population Dataset,” 2017.
- [83] Khatab, Z. E., Hajihoseini, A., Ghorashi, S. A., and Member, S., “Sensor Applications A Fingerprint method for Indoor Localization using Autoencoder

based Deep Extreme Learning Machine,” vol. 1, no. 3, pp. 1–4, 2017.

- [84] Khan, U. M., Kabir, Z., Hassan, S. A., and Ahmed, S. H., “A Deep Learning Framework using Passive WiFi Sensing for Respiration Monitoring,” 2017.
- [85] Song Tang, X., Hao, K., Wei, H., and Ding, Y., “Using line segments to train multi-stream stacked autoencoders for image classification,” *Pattern Recognit. Lett.*, vol. 94, pp. 55–61, 2017.
- [86] Yin, C., Zhu, Y., Fei, J., and He, X., “A Deep Learning Approach for Intrusion Detection Using Recurrent Neural Networks,” vol. 5, 2017.
- [87] Yu, Z. *et al.*, “A Deep Convolutional Neural Network Based Framework for Automatic Fetal Facial Standard Plane Recognition,” vol. 2194, no. c, pp. 1–12, 2017.
- [88] Hordri, N. F., Yuhaniz, S. S., and Shamsuddin, S. M., “Deep Learning and Its Applications: A Review,” no. May 2017, 2016.
- [89] Najafabadi, M. M., Villanustre, F., Khoshgoftaar, T. M., Seliya, N., Wald, R., and Muharemagic, E., “Deep learning applications and challenges in big data analytics,” *J. Big Data*, vol. 2, no. 1, 2015.
- [90] Deng, L. and Yu, D., “Deep Learning: Methods and Applications,” *Found. Trends® Signal Process.*, vol. 7, no. 3–4, pp. 197–387, 2014.
- [91] Yan, Y., Tan, Z., Su, N., and Zhao, C., “Building extraction based on an optimized stacked sparse autoencoder of structure and training samples using LIDAR DSM and optical images,” *Sensors (Switzerland)*, vol. 17, no. 9, 2017.
- [92] Nweke, H. F., The, Y. W., Al-garadi, M. A., and Alo, U. R., “Deep learning algorithms for human activity recognition using mobile and wearable sensor

- networks: State of the art and research challenges,” *Expert Syst. Appl.*, vol. 105, pp. 233–261, 2018.
- [93] Gu, J. *et al.*, “Recent advances in convolutional neural networks,” *Pattern Recognit.*, 2017.
- [94] Ryczko, K., Mills, K., Luchak, I., Homenick, C., and Tamblyn, I., “Convolutional neural networks for atomistic systems,” *Comput. Mater. Sci.*, vol. 149, no. March, pp. 134–142, 2018.
- [95] Affonso, C., Rossi, A. L. D., Vieira, F. H. A., and A. C. P. de L. F. de Carvalho, “Deep learning for biological image classification,” *Expert Syst. Appl.*, vol. 85, pp. 114–122, 2017.
- [96] Tran, D. T., Iosifidis, A., and Gabbouj, M., “Improving efficiency in convolutional neural networks with multilinear filters,” *Neural Networks*, vol. 105, pp. 328–339, 2018.
- [97] Wu, H. and Zhao, J., “Deep convolutional neural network model based chemical process fault diagnosis,” *Comput. Chem. Eng.*, vol. 115, pp. 185–197, 2018.
- [98] Guo, Y., He, Y., Song, H., He, W., and Yuan, K., “Correlational examples for convolutional neural networks to detect small impurities,” *Neurocomputing*, vol. 295, pp. 127–141, 2018.
- [99] Deng, H., Zhang, L., and Shu, X., “Feature memory-based deep recurrent neural network for language modeling,” *Appl. Soft Comput. J.*, vol. 68, pp. 432–446, 2018.
- [100] Folli, V., Gosti, G., Leonetti, M., and Ruocco, G., “Effect of dilution in asymmetric recurrent neural networks,” *Neural Networks*, vol. 104, pp. 50–59, 2018.

- [101] Li, Y., Zheng, W., Cui, Z., and Zhang, T., “Face recognition based on recurrent regression neural network,” *Neurocomputing*, vol. 297, pp. 50–58, 2018.
- [102] Shi, C. and Pun, C. M., “Multi-scale hierarchical recurrent neural networks for hyperspectral image classification,” *Neurocomputing*, vol. 294, pp. 82–93, 2018.
- [103] Simoncini, M., Taccari, L., Sambo, F., Bravi, Salti, L., S., and Lori, A., “Vehicle classification from low-frequency GPS data with recurrent neural networks,” *Transp. Res. Part C Emerg. Technol.*, vol. 91, no. December 2017, pp. 176–191, 2018.
- [104] Singh, S., Pandey, S. K., Pawar, U., and Janghel, R. R., “Classification of ECG Arrhythmia using Recurrent Neural Networks,” *Procedia Comput. Sci.*, vol. 132, no. Iccids, pp. 1290–1297, 2018.
- [105] Gupta, A., Müller, A. T., Huisman, B. J. H., Fuchs, J. A., Schneider, P., and Schneider, G., “Generative Recurrent Networks for De Novo Drug Design,” *Mol. Inform.*, vol. 37, no. 1, 2018.
- [106] Zhao, R., Yan, R., Chen, Z., Mao, K., Wang, P., and Gao, R. X., “Deep learning and its applications to machine health monitoring,” *Mech. Syst. Signal Process.*, vol. 115, pp. 213–237, 2019.
- [107] Huang, R., Liu, C., Li, G., and Zhou, J., “Adaptive Deep Supervised Autoencoder Based Image Reconstruction for Face Recognition,” *Math. Probl. Eng.*, vol. 2016, 2016.
- [108] Jha, D., and Kwon, G.-R., “Alzheimer’s Disease Detection Using Sparse Autoencoder, Scale Conjugate Gradient and Softmax Output Layer with Fine Tuning,” *Int. J. Mach. Learn. Comput.*, vol. 7, no. 1, pp. 13–17, 2017.
- [109] Siswanto, J., Prabuwo, A. S., Abdullah, A., and Idrus, B., “A linear model

- based on Kalman filter for improving neural network classification performance,” *Expert Syst. Appl.*, vol. 49, pp. 112–122, 2016.
- [110] Hoque, N., Kashyap, H., and Bhattacharyya, D. K., “Real-time DDoS attack detection using FPGA,” *Comput. Commun.*, vol. 110, pp. 48–58, 2017.
- [111] Gupta, B. B., “Predicting number of zombies in DDoS attacks using pace regression model,” *J. Comput. Inf. Technol.*, vol. 20, no. 1, pp. 33–39, 2012.
- [112] Alkasassbeh, M., Hassanat, A. B. A., and Al-naymat, G., “Detecting Distributed Denial of Service Attacks Using Data Mining Techniques,” vol. 7, no. 1, pp. 436–445, 2016.
- [113] Moustafa, N. and Slay, J., “UNSW-NB15: a comprehensive data set for network intrusion detection systems (UNSW-NB15 network data set),” *2015 Mil. Commun. Inf. Syst. Conf.*, pp. 1–6, 2015.
- [114] Yücel, S., Terzioğlu, P., and Özçimen, D., “World’s largest Science , Technology & Medicine Open Access book publisher c,” *RFID Technol. Secur. Vulnerabilities, Countermeas.*, 2016.
- [115] Mormann, F., Andrzejak, R. G., Elger, C. E., and Lehnertz, K., “Seizure prediction: The long and winding road,” *Brain*, vol. 130, no. 2, pp. 314–333, 2007.
- [116] Evidence, P., Eb, B., and Neprovociranom, P., “EVIDENCE BASE (EB) APPROACH TO THE FIRST UNPROVOKED.”
- [117] James, C. J. and Eng, B. E., “Detection of epileptiform activity in the electroencephalogram using artificial neural networks,” no. February 1997, 1997.
- [118] Karim, A. M., Çelebi, F. V., and Mohammed, A. S., “Software Development

- for Blood Disease Expert System,” *Lect. Notes Softw. Eng.*, vol. 4, no. 3, pp. 179–183, 2016.
- [119] Khemphila, A. and Boonjing, V., “Heart Disease Classification Using Neural Network and Feature Selection,” in *2011 21st International Conference on Systems Engineering*, 2011, no. 2007, pp. 406–409.
- [120] Andrzejak, R. G., Lehnertz, K., Mormann, F., Rieke, C., David, P., and Elger, C. E., “Indications of nonlinear deterministic and finite-dimensional structures in time series of brain electrical activity: Dependence on recording region and brain state,” *Phys. Rev. E*, vol. 64, no. 6, p. 61907, 2001.
- [121] LeCun, Y., Bottou, L., Bengio, Y., and Haffner, P., “Gradient-based learning applied to document recognition,” *Proceedings of the IEEE*, 86(11):2278-2324, November 1998.
- [122] Kaushik, A., Gupta, H., and Latwal, D. S., “Impact of Feature Selection and Engineering in the Classification of Handwritten Text,” *2016 Int. Conf. Comput. Sustain. Glob. Dev.*, pp. 2598-2601, 2016.
- [123] Schomaker, L. and Wiering, M. A., “Handwritten Character Classification using the Hotspot Feature Extraction Technique,” *Proc. 1st Int. Conf. Pattern Recognit. Appl. Methods*, no. Feb., pp. 261-264, 2012.
- [124] Babu, U. R., Venkateswarlu, Y., and Chintha, A. K., “Handwritten digit recognition using K-nn classifier,” *Proc.-2014 World Congr. Comput. Commun. Technol. WCCCT 2014*, pp. 60-65, 2014.
- [125] Hinton, G. E., Osindero, S., Teh YW, “A fast learning algorithm for deep belief nets,” *Neural computation*. 2006; 18:1527–1554. doi: 10.1162/neco.2006.18.7.1527 PMID: 16764513.

- [126] LeCun Y, Bottou L, Bengio Y, Haffner P. “Gradient-based learning applied to document recognition,” *Proceedings of the IEEE*. 1998; 86:2278–2324. doi: 10.1109/5.726791.
- [127] Wan L, Zeiler M, Zhang S, LeCun Y, Fergus R. “Regularization of neural networks using DropConnect,” *30th International Conference on Machine Learning*, Atlanta, Georgia, USA; volume 28; 2013.
- [128] Zelier MF, Fergus R. “Stochastic pooling for regularization of deep CNN. In *proc. International Conference on Learning Representations*, Scottsdale, USA, 2013.
- [129] Goodfellow IJ, Warde-Farley D, Mirza M, Courville A, Bengio Y. “Maxout networks,” *30th International Conference on Machine Learning*, Atlanta, Georgia, USA; *JMLR: W&CP* volume 28; 2013.
- [130] Lee CY, Xie S, Gallagher P, Zhang Z, Tu Z. “Deeply-supervised nets,” In: *Deep Learning and Representation Learning Workshop*, NIPS; 2014.
- [131] Dua, D. and Karra Taniskidou, “Machine Learning Repository,” <http://archive.ics.uci.edu/ml>, University of California, Irvine, School of Information and Computer Sciences, 2017.
- [132] Y. Ju, J. Guo, and S. Liu, “A Deep Learning Method Combined Sparse Autoencoder with SVM,” in *Proceedings of the 2015 International Conference on Cyber-Enabled Distributed Computing and Knowledge Discovery (CyberC)*, pp. 257–260, Xi'an, China, September 2015.
- [133] Bamakan, S. M. H., Wang, H., and Shi, Y., “Ramp loss K-Support Vector Classification-Regression; a robust and sparse multi-class approach to the intrusion detection problem,” *Knowledge-Based Systems*, vol. 126, pp. 113–126, 2017.

- [134] C. Khammassi and S. Krichen, “A GA-LR wrapper approach for feature selection in network intrusion detection,” *Computers & Security*, vol. 70, pp. 255–277, 2017.

CURRICULUM VITAE



PERSONAL INFORMATION

Name Surname : Ahmad
 KARIM
Date of Birth : 05/04/1987
Phone : +90 531 494 91 39
E-mail : ahmad.mozaffer.karim@gmail.com

EDUCATION

High School : Ibn Haldon High School- Kirkuk-Iraq
Bachelor : Northern Technical University, Engineering
 Technical College of Kirkuk. Kirkuk-Iraq (2009)
Master Degree : Çankaya University (2012)
Ph.D : Ankara Yıldırım Beyazıt University (2013/Continue)

WORK EXPERIENCE

Lecturer : Electrical& Electronic Engineering Dep. – Aksaray Uni
 (2016-2018)
System Analyst : CBKSOFT Yazılım, Ankara- Turkey (2014-2016)
Project Manager : EES, Ankara- Turkey (2013-2014)

TOPICS OF INTEREST

-Expert Systems
 - Deep Learning
 - Data Structure
 - Artificial intelligence

PUBLISHED WORKS RELATED TO THESIS

SCI JOURNAL PUBLICATIONS

[1] **Ahmad M. Karim**, Mehmet S. Güzel, Mehmet R. Tolun, Hilal Kaya, and Fatih V. Çelebi, “A New Generalized Deep Learning Framework Combining Sparse Autoencoder and Taguchi Method for Novel Data Classification and Processing,” *Mathematical Problems in Engineering*, vol. 2018, Article ID 3145947, 13 pages, 2018. <https://doi.org/10.1155/2018/3145947>.

[2] **Ahmad M. Karim**, Mehmet S. Güzel, Mehmet R. Tolun, Hilal Kaya, and Fatih V. Çelebi “A New Framework using Deep Auto-encoder and Energy Spectral Density for Medical Waveform Data Classification and Processing,” *Biocybernetics and Biomedical Engineering*, <https://doi.org/10.1016/j.bbe.2018.11.004>.

[3] **Ahmad M. Karim**, Mehmet R. Tolun, Mehmet S. Güzel, Hilal Kaya, and Fatih V. Çelebi “A New Framework Using Deep Auto-Encoders and Linear Model Based Particle Swarm Optimization Algorithm for Classification,” *Expert Systems*. (Revision)

CONFERENCE PUBLICATION

[1] **Ahmad M. Karim**, Ömer Karal, Fatih V. Çelebi, “A New Automatic Epilepsy Serious Detection Method by Using Deep Learning Based on Discrete Wavelet Transform,” 3rd International Conference on Engineering Technology and Applied Sciences (ICETAS) July 17-21 2018 Skopje Macedonia.