Republic of Iraq
Ministry of Higher Education and Scientific Research
University of Babylon
Collage of Information Technology
Software Department

Using Convolutional Neural Networks (CNNs) for Classifying Alzheimer's Disease Stages

A Thesis
Submitted to the Council of the College of Information Technology for Postgraduate Studies of the University of Babylon in Partial Fulfillment of the Requirements for the Degree of Master in Information Technology/Software

Prepared by:

Hussain Mohammed Khalil Al-Zubaidee

Supervised by:

Senior Lecturer Dr. Mohannad Mohammed Al-Yasiry

2022 A.C. **1444 A.H.**

﴿وَعَلَّمَكَ مَا لَمْ تَكُنْ تَعْلَمُ﴾

[سورة النساء: الآية 113]

Dedication

This thesis is dedicated to my parents who supported me financially and morally and prayed to God a lot for me, to my two sisters who were proud of me all the time, to my friends who helped me a lot in my studies, especially **Mohammed Qassim** who helped me a lot to complete my studies, to my former manager in The work **Mr. Mohsen Arab** who supported me and was understanding of the circumstances of my studies, my colleagues at work who filled my place and performed my work during my absence, to all my family and relatives who love me and support me and pray to God for me and last but not least I dedicate it to the love of my life ,my wife **Shahad**, who was the best blessing and gift from God to me, supported me a lot, encouraged me a lot, and worked hard for me to obtain a master's degree.

Hussain

Acknowledgment

In the name of God, the Most Gracious, the Most Merciful. Praise be to God. And prayers and peace be upon the Seal of the Prophets and Messengers, Mohammed and his infallible family.

First and foremost, I must acknowledge my limitless thanks to Allah, the Ever-Magnificent; the Ever-Thankful, for His help and blessing. I am sure that this work would have never become true, without His guidance.

I would first like to thank my thesis supervisor, Senior Lecturer Dr. Mohannad Mohammed Al-Yasiry. The door to the 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 work but steered me in the right direction whenever they thought I needed it.

I would like to take this opportunity to say warm thanks to all my beloved friends, who have been so supportive along the way of doing my thesis. I also would like to express my wholehearted thanks to my family for the generous support they provided me and for their unconditional love and prayers, I have the chance to complete this thesis.

Hussain

Supervisor Certification

I certify that this thesis was prepared under my supervision at the Department of Software / Collage of Information Technology / Babylon University, by Hussain M. Alzubaidee as partial fulfillment of the requirements for the degree of Master in Information Technology.

Signature:

Supervisor Name: **Senior Lecturer Dr. Mohannad Mohammed Al-Yasiry**

Date: / / 2022

The Head of the Department Certification

In view of the available recommendations, I forward this thesis for debate by the examination committee.

Signature:

Name:

Head of Software Department

Date: / / 2022

Examination Committee Certification

We certify that read the thesis entitled **"Using Convolutional Neural Networks (CNNs) for Classifying Alzheimer's Disease Stages"** and as examined the student **Hussain Mohammed Khalil Al-Zubaidee** in its content and what it related to, and that in our opinion, it meets the standard of a thesis for the degree of Master of Information Technology.

Signature:

Name: Prof. Dr. Zaki Seed Tawfik

(Chairman)

Date: / / 2022

Signature:

Name: Asst. Prof. Dr. Sura Zaki Alrashid

(Member)

Date: / / 2022

Signature:

Name: Dr. Safa Saad Al-Murib

(Member)

Date: / / 2022

Signature:

Name: Senior Lecturer Dr. Mohannad Mohammed Al-Yasiry

(Member and Supervisor)

Date: / / 2022

Abstract

Alzheimer's disease (AD) is a neurological illness that primarily affects the elderly and readily degrades human memory and behavior. According to reports, there are 26.6 million AD cases globally, including 14.99 million cases in the early stages of 2006. It is expected to harm one in every 85 persons worldwide. There are 47.5 million AD patients globally in 2022, and 58% of them will reside in low- and middle-income nations. Over the last few decades, machine learning approaches have been used to classify AD, with findings based on manually created features and a classifier with a multi-step design. Convolution Neural Networks (CNNs) have recently been used for AD classification due to the advancement of deep learning.

The proposed system includes five stages: preprocessing, data splitting, feature reduction, classification, and evaluation stages. The preprocessing stage was accomplished with the normalization method. The Hold-out validation was used to divide the AD dataset into the training set (70%) which equals (398 instances) and the testing set (30%) which equals (171 instances). The feature was reduced by using two feature reduction techniques: PCA and SVD. While the feature extraction is done by utilizing the proposed CNN feature extraction model. Then, the data on which all previous operations have been performed will be entered into the classification stage, in this stage, three CNN-1D models were proposed in order to increase the accuracy of the proposed system and decrease the time of execution and the error rate. The final stage is an evaluation where some evaluation metrics were utilized to evaluate the suggested system's models.

Experimental results show the accuracy rates equal to 95.60% in the first proposed method, 98.9% in the second proposed method, and 99.1% in the third proposed method used to classify AD data. The error rate of the first, second and

third classification methods is 4.035%, (1.164%, 1.12%, 1.19%, 1.1%), and (0.4%, 0.4%) respectively. The execution time of all proposed methods doesn't exceed a few seconds. The least time was obtained in the third method equal to 0.212 seconds with PCA and 0.219 seconds with SVD, where the use of feature reduction techniques PCA, SVD, and the proposed CNN feature reduction model resulted a less time for classification. The suggested system was built using the Python 3.6 programming language, on a computer with the following properties (Intel Core i7, CPU 2.20 GHz, RAM 16 GB, Operating system: 64-bit).

Table of Contents

Subjects	**Page No.**
Abstract	I
Table of Contents	III
List of Abbreviations	VII
Table of Figures	IX
List of Tables	XI
Table of Algorithms	XII
Chapter One: General Introduction	**Page No.**
1.1. Background	1
1.2. Literature Survey	2
1.3. Problem Statement	8
1.4. Aim of Thesis	8
1.5. Contribution of Thesis	9
1.6. Layouts of Thesis	9
Chapter Two: Theoretical Background	**Page No.**
2.1. Introduction	11
2.2. Overview of Alzheimer's Disease (AD)	11
2.3. Data Preprocessing Stage	14
2.3.1. Data Normalization	15
2.4. Data Splitting Stage	15
2.5. Features (Selection, Extraction, and Reduction)	17
2.5.1. Features Reduction	17
2.5.2. Feature Selection	19
2.5.3. Feature Extraction	19
2.5.3.1. Principal Component Analysis (PCA)	19

2.5.3.2. Singular Value Decomposition (SVD)	21
2.6. Deep Learning Models	22
2.6.1. Unsupervised Deep Learning	22
2.6.2. Supervised Deep Learning	22
2.6.2.1. Convolutional Neural Networks (CNN)	23
2.7. Performance Criteria	29
2.7.1. Accuracy	29
2.7.2. Precision	29
2.7.3. Recall or (Sensitivity)	30
2.7.4. F-measure	30
2.7.5. Loss (Error rate)	30
Chapter Three: The Proposed System	**Page No.**
3.1. Introduction	31
3.2. The Proposed System Architecture	31
3.2.1. AD Data Preprocessing (1st Stage)	33
3.2.2. Splitting Data Stage (2nd Stage)	33
3.2.3. Feature Reduction Stage (3rd Stage)	34
3.2.3.1. Extract Features by Using Principal Component Analysis (PCA)	34
3.2.3.2. Extract Features by Using Singular Value Decomposition (SVD)	35
3.2.3.3. CNN Feature Reduction Stage	36
3.2.4. Classification Stage (4th Stage)	39
3.2.4.1. First Method (Pure Data)	39
3.2.4.2. Second Method (PCA or SVD with CNN)	42

3.2.4.3. Third Method (PCA or SVD with Double CNN models)	45
3.2.6 Evaluation Stage (5th Stage)	47
Chapter Four: Experimental Results and Discussion	**Page No.**
4.1. Introduction	50
4.2. The Proposed System Requirements	50
4.2.1. Software Requirements	50
4.2.2. Hardware Requirements	50
4.3. Dataset Description	50
4.4. Results of the Proposed System	51
4.4.1. The First Classification Method Results	51
4.4.2. The Second Classification Method Results	53
4.4.2.1. CNN classification model "2" with PCA-10	54
4.4.2.2. CNN classification model "2" with PCA-15	54
4.4.2.3. CNN classification model "2" with SVD-10	55
4.4.2.4. CNN classification model "2" with SVD-15	56
4.4.3. The Third Classification Method Results	57
4.4.3.1. CNN classification model "3" with PCA	57
4.4.3.2. CNN classification model "3" with SVD	58
4.5. Discussion of Results	59
4.5.1. Discussion of first method results	59
4.5.2. Discussion of second method results	60
4.5.3. Discussion of third method results	60
4.6. Comparison with Previous Studies	62
Chapter Five: Conclusions and Recommendations	**Page No.**
5.1. Conclusions	67

5.2. Recommendations for Future Works	68
References	69

List of Abbreviations

Abbreviation	Meaning
1D CNN	One Dimension Convolution Neural Networks
2D CNN	Two Dimension Convolution Neural Networks
3D CNN	Three Dimension Convolution Neural Networks
AD	Alzheimer's disease
ADNI	Alzheimer's Disease Neuroimaging Initiative
AE	Acoustic Emission
CAE	Convolutional Auto Encoders
CFS	Correlation-based Feature Selection
CNNs	Convolution Neural Networks
CTL	Healthy Controls
DL	Deep Learning
DNN	Deep Neural Networks
FCBF	Fast Correlation-Based Filter
GPU	Graphics Processing Unit
IG	Information Gain
MCI	Mild Cognitive Impairment
ML	Machine Learning
MMSE	Mini-Mental State Examination
MRI	Magnetic Resonance Imaging
NC	Normal Cognitive
PCA	Principal Component Analysis
RBM	Restricted Boltzmann Machine
RF	Random Forest

RNN	Recurrent Neural Networks
SVD	Singular Value Decomposition
SVM	Support Vector Machines

Table of Figures

Figure No.	Figure Title	Page No.
(2-1)	Symptoms of Alzheimer's Diseases	12
(2-2)	Alzheimer's Disease (AD) Continuum	13
(2-3)	Hold-out y K-fold cross-validation	16
(2-4)	The classification of dimensionality reduction techniques	17
(2-5)	The SVD decomposition of an n × d matrix	21
(2-6)	A convolutional neural network (CNN) architecture	23
(2-7)	Examples of pooling operations by using 2 × 2 filters applied with a stride of 2	26
(3-1)	An overall diagram of the proposed system	32
(3-2)	The Proposed CNN feature extraction model	36
(3-3)	The Proposed CNN Classification Model "2"	42
(4-1)	Sample of AD dataset	51
(4-2)	Chart of first CNN classification model results	52
(4-3)	Description of genes	53
(4-4)	Chart of second CNN classification model with PCA-10 results	56
(4-5)	Chart of second CNN classification model with PCA-15 results	57
(4-6)	Chart of second CNN classification model with SVD-10 results	58
(4-7)	Chart of second CNN classification model with SVD-15 results	59

(4-8)	Chart of third CNN classification model with PCA results	60
(4-9)	Chart of second CNN classification model with SVD results	61
(4.10)	Chart of proposed models comparison	64
(4.10)	Chart of accuracy comparison with related studies	64

List of Tables

Table No.	Table Title	Page No.
(1-1)	Summary of the Related Works	6
(2-1)	Stages of development of Alzheimer's disease with their symptoms	13
(2-2)	Comparison of dimensionality reduction techniques	18
(3-1)	Proposed CNN feature reduction model layers	37
(3-2)	Proposed CNN classification model "2" layers	43
(4-1)	AD Dataset Description	51
(4-2)	Results of 1st CNN classification model.	52
(4-3)	The 15th effective features	53
(4-4)	The 10th effective features	54
(4-5)	The 3rd effective features	54
(4-6)	Results of 2nd CNN classification model with PCA-10	54
(4-7)	Results of 2nd CNN classification model with PCA-15	56
(4-8)	Results of 2nd CNN classification model with SVD-10	57
(4-9)	Results of 2nd CNN classification model with SVD-15	58
(4-10)	Results of 3rd CNN classification model with PCA	60
(4-11)	Results of 3rd CNN classification model with SVD	61
(4-12)	Proposed models comparison	63
(4-13)	Comparison with related works	64

Table of Algorithms

Algorithm No.	Algorithm Title	Page No.
(3-1)	AD Dataset Splitting	33
(3-2)	Principal Component Analysis (PCA)	35
(3-3)	Singular Value Decomposition (SVD)	35
(3-4)	Feature Reduction CNN Model	38
(3-5)	The proposed CNN classification model “1”	40
(3-6)	The proposed CNN classification model “2”	44
(3-7)	The proposed CNN classification model “3”	46
(3-8)	The Proposed System Evaluation	47

Chapter One
General Introduction

Chapter One
General Introduction

1.1. Background

Alzheimer's disease (AD), a chronic neurodegenerative illness that causes nerve cell death and tissue loss throughout the brain, often begins slowly and progresses over time. By the year 2050, Alzheimer's disease is anticipated to impact one out of every 85 persons on the planet. The expense of care for Alzheimer's patients is also predicted to climb considerably, necessitating the development of specific computer-aided methods for early and accurate Alzheimer's diagnosis [1]. Memory loss, difficulty planning, difficulty performing simple activities, and a reduction in other cognitive skills are all indicators of AD, which often leads to a need for outside care, which has a negative impact on both the individual with AD's family and society as a whole [2]. The illness is distinguished by the degeneration of certain nerve cells, the development of neuritic plaques outside the neurons, and the formation of neurofibrillary tangles within the neurons, where the latter two proteins cause cell death by interfering with numerous cell activities. As a result, persons with severe Alzheimer's disease have brain inflammation, substantial shrinkage from cell loss, and extensive debris from dead or dying neurons [3].

Magnetic Resonance Imaging (MRI) allows researchers to analyze degenerative brain alterations linked with Alzheimer's Disease (AD) in real time. Over the last few decades, neuroimaging data have increasingly been utilized to describe Alzheimer's disease (AD) using Machine Learning (ML) approaches, providing promising tools for customized diagnosis and prognosis [4]. Many research has proposed using predetermined features (containing regional and voxel-based measurements) derived from image preprocessing pipelines in conjunction

with various sorts of classifiers, such as Support Vector Machines (SVM) or Random Forest (RF). This method is sometimes referred to as standard ML [5]. Deep Learning (DL), a relatively new ML paradigm, has lately achieved significant progress in the field of medical imaging. The Convolutional Neural Network (CNN) has gained a lot of interest as the most extensively used DL architecture because of its significant performance in classification [6]. Unlike traditional ML, DL enables the automated abstraction of low- to high-level latent feature representations (e.g., lines, dots, or edges for low-level features and objects or bigger forms for high-level features) [7]. As a result, one might assume that DL is less reliant on data preprocessing and requires less previous preparation for more difficult operations, such as feature selection, leading to a more objective and less biased process [8].

1.2. Related Works

In order to diagnose this disease, different state-of-the-art methods have been used. AD classification has been widely studied, and it involves several issues and challenges. In this section, several studies on AD classification depending on CNNs are presented as follows:

- **(R. Jain et al., 2018) [9]** introduced a transfer learning method for reliably identifying brain MRI partitioned into three categories: AD, Normal Cognitive (CN), and Mild Cognitive Impairment (MCI). In this scenario, a pre-trained VGG16 network was used for transfer learning and as a feature extractor. Despite the fact that VGG16 was trained on images from the ImageNet dataset (natural images), it was able to extract useful features for the proposed approach classification task. Experiments are carried out using data from the Alzheimer's Disease Neuroimaging Initiative (ADNI) dataset. For the validation set, the accuracy of the 3-way classification utilizing the stated approach is 95.73 %.

- **(J. Islam and Y. Zhang, 2018) [10]** presented a deep Convolutional Neural Network (CNN) for Alzheimer's disease diagnosis utilizing brain MRI data processing. While most existing systems use binary classification, the suggested model may detect distinct phases of Alzheimer's disease and achieves improved results for early-stage diagnosis. Extensive tests were carried out to show that the suggested method outperforms comparison baselines on the Open Access Series of Imaging Studies dataset, with an accuracy of 93%.

- **(O. Kanghan et al., 2019) [11]** employed unsupervised learning based on Convolutional Auto Encoders (CAE) to solve the AD vs. NC classification challenge, and supervised transfer learning to handle the pMCI vs. sMCI classification job. The most significant biomarkers associated with AD and pMCI were identified using a gradient-based visualization technique that approximates the geographic influence of the CNN model's choice. Research using the ADNI database revealed that the proposed method beat current network models with accuracy rates of 86.60% and 73.95% for the AD and pMCI classification tasks, respectively.

- **(S. Basheera and M. Ram, 2019) [12]** employed 1820 T2-weighted brain magnetic resonance volumes, cut into 18,017 voxels, including 635 AD MRIs, 548 MCI MRIs, and 637 CN MRIs. The voxels are improved using a Gaussian filter, and the unnecessary tissues are removed using a skull stripping technique. The voxels are then segmented using hybrid improved independent component analysis. The CNN receives input from segmented gray matter. The suggested technique was used to do a clinical valuation, which resulted in 90.47% accuracy, 86.66 % recall, and 92.59% precision.

- **(S. Bringas et al., 2020) [13]** employed data from 35 patients with AD collected by smartphones for a week in a daycare center. The data sequences of each patient

recorded the accelerometer changes while daily activities were performed and they were labeled with the stage of the disease (early, middle, or late). Their methodology processes these time series and uses a Convolutional Neural Network (CNN) model to recognize the patterns that identify each stage. Results: The CNN-based method achieved a 90.91 % accuracy.

- **(Y. Fu'adah et al., 2020) [14]** presented the Convolutional Neural Network (CNN) as a method for creating an automated Alzheimer's disease classification system using the AlexNet architecture. The project uses Magnetic Resonance Imaging (MRI) information from 664 MRI datasets to categorize Non-Demented, Very Mild Demented, Mild Demented, and Moderate Demented. This study obtained a 95% accuracy rate from the trial.

- **(M. Al-Adhaileh, 2021) [15]** classified and recognized Alzheimer's disease utilizing two deep neural network algorithms, AlexNet and Restnet50. Data from brain magnetic resonance imaging (MRI) scans obtained from the Kaggle website were utilized to analyze and test the suggested model. To efficiently classify AD, a Convolutional Neural Network (CNN) technique was used. AlexNet and Restnet50 transfer learning models were used to train CNNs. For the brain MRI datasets, the AlexNet model performed admirably. AlexNet outperformed Restnet50 with an accuracy of 94.53%.

- **(Y. AbdulAzeem et al., 2021) [16]** suggested an end-to-end system for AD categorization based on CNN. The framework is composed of five layers, the first of which is in charge of MRI acquisition. Adaptive thresholding and data augmentation are employed in the second layer. The cross-validation approach is used to train the CNN at the third layer. The CNN model is used in the fourth layer. The CNN design is made up of three convolutional layers, with max-pooling done after each one. Following the convolutional layers are two fully

linked layers. The categorization procedure is carried out in the fifth layer using a variety of algorithms. The suggested framework obtained 97.5 % classification accuracy in classification experiments on the ADNI dataset.

- **(H. M. Mohammed, 2021) [17]** aimed to find genes that may be associated with Alzheimer's disease and offer a prediction model that can aid doctors in identifying Alzheimer's patients, preventing progression, and preserving health to achieve these goals, the following goals would be met: finding the most optimum subset of genes (informative genes) from gene expression data that are utilized for the prediction job, and developing a prediction model with the least possible error based on the selected genes. As a result, this model is used to distinguish individuals with Alzheimer's disease (AD) from those with Moderate Cognitive Impairment (MCI) or Healthy Controls (CTL). The AD dataset was utilized to accomplish the goals. The examination was carried out using two prediction measures (accuracy and loss). The findings suggest that the proposed system's performance is effective. The suggested model's greatest accuracy values (97.4%) were achieved with 150 genes in particular.
- **(M. Odusami et al., 2022) [18]** employed randomly concatenated deep features learned from two previously training images that concurrently learn deep features from brain functional networks utilizing Magnetic Resonance Imaging (MRI) images to address achieve their goal. ResNet18 and DenseNet201 were utilized in the studies to perform the goal of AD multiclass classification. To highlight the area of the image that discriminated between the specified model prediction and other options, a gradient class activation map was utilized. The experimental results revealed that the proposed method had a multi-class classification accuracy of 98.86%, a precision of 98.94%, and a recall of 98.89%.

- **(L. Jiyun et al., 2022) [19]** proposed a model called Feature Weighted-LSTM (FW-LSTM). The feature weight is defined by spatial characteristics calculating the frequency of connectivity of each brain region and further integrated into the LSTM. Thus, it can comprehensively model both temporal and spatial changes in rs-fMRI brain regions. The FW-LSTM model on the Alzheimer's Disease Neuroimaging Initiative (ADNI) dataset is used to extract the time-varying characteristics of 90 brain regions for Alzheimer's disease (AD) classification. The model performances are 77.80%, 76.41%, and 78.81% in accuracy, sensitivity, and specificity. It outperformed the one-dimensional convolutional neural networks (1D-CNN) model and LSTM model, which only used temporal features of brain regions.

A summary of the related works with additional details is listed in Table (1.1).

Table (1.1). Summary of the Related Works

Ref. No.	Authors and Work date	Dataset Name	Used Techniques	Accuracy Rate %
[9]	R. Jain et al., 2018	Alzheimer's Disease Neuroimaging Initiative (ADNI)	Convolutional Neural Network (CNN)	95.73 %
[10]	J. Islam and Y. Zhang, 2018	Open Access Series of Imaging Studies (OASIS)	Convolutional Neural Network (CNN)	93%

[11]	O. Kanghan et al., 2019	Open Access Series of Imaging Studies (OASIS)	Convolutional Neural Network (CNN)	86.60%
[12]	S. Basheera and M. Ram, 2019	Magnetic Resonance Imaging (MRI)	Convolutional Neural Network (CNN)	90.47%
[13]	S. Bringasa et al., 2020	Collected from smartphone	Convolutional Neural Network (CNN)	90.91 %.
[14]	Y. Fu'adah et al., 2020	Magnetic Resonance Imaging (MRI)	Convolutional Neural Network (CNN)	95%
[15]	M. Al-Adhaileh, 2021	Magnetic Resonance Imaging (MRI)	Convolutional Neural Network (CNN)	94.53%
[16]	Y. AbdulAzeem et al., 2021	Alzheimer's Disease Neuroimaging Initiative (ADNI)	Convolutional Neural Network (CNN)	97.5 %
[17]	**Heba M. Mohammed**	**AD Dataset**	**Convolutional Neural Network (CNN) + Sequential Gene Selection (SGS)**	**97.4%**

[18]	M. Odusami et al., 2022	Magnetic Resonance Imaging (MRI)	Convolutional Neural Network (CNN)	98.86%
[19]	L. Jiyun et al., 2022	Alzheimer's disease neuroimaging initiative (ADNI) dataset	Convolutional Neural Network (CNN)	77.80%

1.3. Problem Statement

Previous research has shown that the prevalence of Alzheimer's disease climbs sharply with age, doubling every 4-5 years beyond the age of 60, such that more than one-third of people over the age of 80 are at risk of developing the disease. As a result, controlling the Alzheimer's disease-affected population becomes increasingly challenging. Furthermore, Alzheimer's disease is one of the most expensive diseases for people of all socioeconomic backgrounds. As a result, researchers should focus more on characteristics that may assist predict AD patients. This is because these traits that can help predict AD still require more research. There is a need for an intelligent prediction (classification) system for the early diagnosis of AD that can assist to enhance disease prevention before it occurs, which is recognized as one of the most essential pillars of therapy at various stages of this illness.

1.4. Aim of Thesis

The main aim of this thesis is to build a disease-stage classification system for Alzheimer's patients and to identify genes that may influence the development of

Alzheimer's disease. By relying on multiple models of the Convolutional Neural Network (CNN), the performance of the proposed system can be improved in recognition and classification, as deep learning contributes significantly to increasing the accuracy of the system.

1.5. Contribution of Thesis

The main contributions of this thesis are:

1) Identifying the best collection of genes (informative genes) from gene expression data to utilize for prediction.
2) Suggest three models of the Convolutional Neural Network (CNN) and in a different way for each of them, whether in the way of extracting or classifying data (genes).
3) The CNN-1D was used in order to reduce the complexity of CNN-2D, in addition to reducing the execution time.

1.6. Layout of Thesis

Aside from the current chapter, there are four more chapters in this study, which are organized as follows:

- **Chapter Two, "Theoretical Background":** deals with the fundamental of the deep learning system, the history of Alzheimer's disease, features selection, and features extraction methods.
- **Chapter Three, "The Proposed System":** is concerned with describing the design of the presented system.
- **Chapter Four, "Experimental Results and Discussion":** The experimental results of each of the measures in this study are shown by the reasons.

- **Chapter Five, "Conclusions and Recommendations":** This chapter presents conclusions from the research carried out and offers a collection of points for possible directions for future works.

Chapter Two
Theoretical Background

Chapter Two

Theoretical Background

2.1. Introduction

In this chapter, the basic concepts of Alzheimer's Disease (AD) will be discussed in some detail. The theoretical foundation of this work will be explained, including deep learning algorithms, especially the Convolution Neural Network (CNN) algorithm, data preprocessing methods, and feature extraction techniques. At the end of the chapter, the metrics that will be utilized to evaluate the performance of the techniques used in classification are explained. The primary techniques and approaches used in this thesis are the major emphasis of this chapter.

2.2. Overview of Alzheimer's Disease (AD)

The most common kind of dementia is Alzheimer's disease (AD). In developed nations, the prevalence of Alzheimer's disease is believed to be approximately 5% after the age of 65, with a shocking 30% for those over the age of 85. It is anticipated that by 2050, about 0.64 billion individuals would have been diagnosed with Alzheimer's disease. Alzheimer's disease kills brain cells, leading patients to lose memory, mental processes, and the capacity to carry on with daily tasks. Alzheimer's disease initially impacts the portion of the brain that regulates language and memory [20].

As a consequence, Alzheimer's disease sufferers have memory loss, disorientation, and trouble speaking, reading, or writing. They frequently lose track of time and may fail to recognize family members. They have difficulty performing daily tasks such as brushing their teeth or combing their hair. All of these things make Alzheimer's patients nervous, hostile, or prone to wandering away from home. AD eventually damages the portion of the brain that controls respiration and the

heart function, resulting in death. Fig. (2.1) depicts some common Alzheimer's symptoms [21].

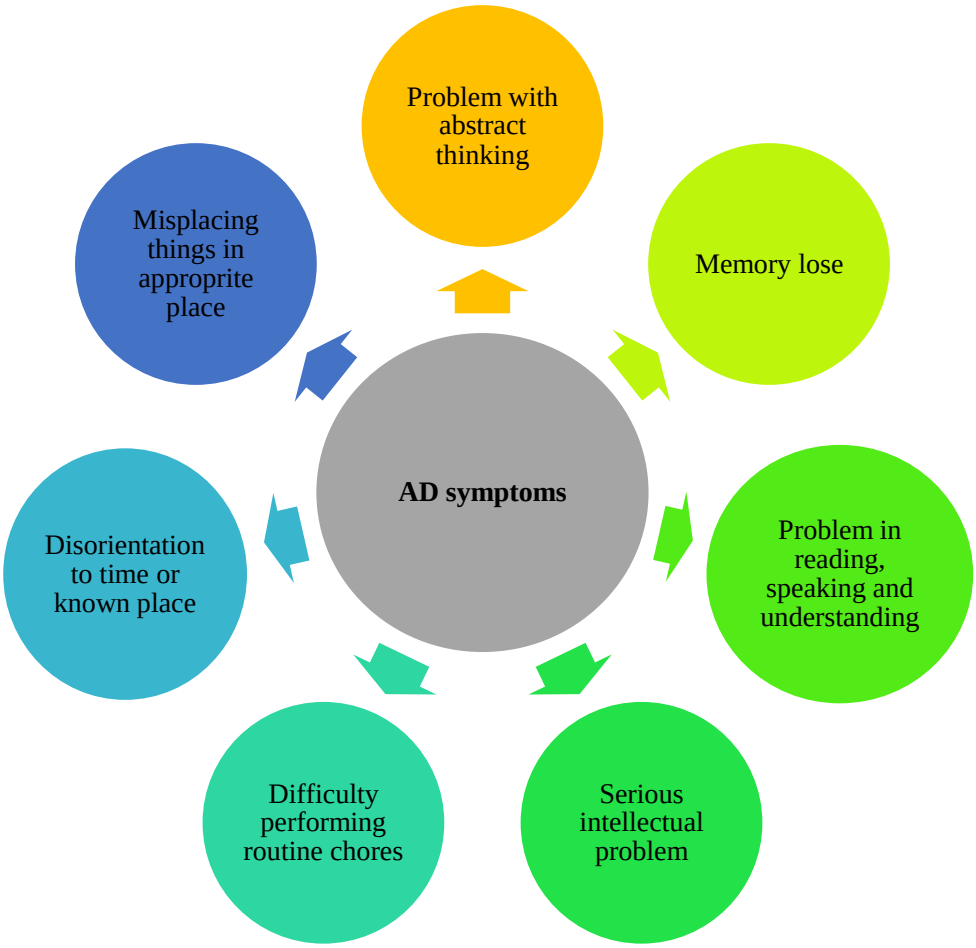

Figure (2.1). Symptoms of Alzheimer's Disease [21].

The progression of Alzheimer's disease from brain changes that are unnoticeable to the person affected to the brain changes that cause problems with memory and eventually physical disability is called the Alzheimer's disease continuum. On this continuum, there are three broad phases: preclinical Alzheimer's disease, mild cognitive impairment (MCI) due to Alzheimer's disease, and dementia due to Alzheimer's disease, also called Alzheimer's dementia, see Fig. (2.2). The Alzheimer's dementia phase is further broken down into mild, moderate, and severe dementia [22].

Figure (2.2). Alzheimer's Disease (AD) Continuum [22].

Table (2.1) shows the stages of development of Alzheimer's disease and the symptoms associated with it.

Table (2.1). Stages of development of Alzheimer's disease with their symptoms [22].

The phase of Alzheimer's disease	Symptoms
Preclinical AD	There are no symptoms, although there might be biochemical alterations in the brain
Mild Cognitive Impairment Due to AD	Very mild symptoms that could not affect daily activities
Mild AD	Symptoms make certain daily tasks difficult.
Moderate AD	Numerous daily tasks are hampered by symptoms.
Severe AD	Symptoms affect the majority of daily tasks.

Alzheimer's disease (AD) detection is still inaccurate until a patient reaches the intermediate AD stage. Physical and neurobiological exams, the Mini-Mental State Examination (MMSE), and a full history are all required for a proper medical assessment of Alzheimer's disease. Recently, clinicians have begun to use brain MRI to diagnose Alzheimer's disease. AD decreases the hippocampus and cerebral cortex while enlarging the ventricles [23].

The hippocampus is the portion of the brain that is in charge of episodic and spatial memory. It also serves as a link between our bodies and our brain. The hippocampal shrinkage causes cell death and injury, particularly to synapses and neuron endings. As a result, neurons can no longer connect via synapses. As a consequence, brain areas associated with remembering (short-term memory), reasoning, planning, and judgment are compromised [24].

2.3. Data Preprocessing Stage

Data may be found in a variety of formats, including organized and unstructured tables, images, audio files, and videos, among others. It is important to transform the provided data into 1s and 0s so that a machine can interpret the free text, video, or image. Therefore, the deep learning model cannot be fed raw data and is expected to be trained. The first phase in machine learning is called pre-processing, during which the input is processed or encoded so that the computer can swiftly scan through or understand it [25].

In other words, it might be understood as a quick analysis of the data's characteristics by the model algorithm. For a Deep Learning Algorithm to perform well in terms of generalization, data pre-processing is the most crucial and influential step. In relation to the input space dimension, the amount of training data increases exponentially. Pre-processing, it is estimated, can take up to 50% to 80% of the time required for the full classification process, demonstrating its significance in the development of models. For greater performance, it is also crucial to increase the quality of the data. Before beginning the analysis with the real data for the model, there are several procedures in data processing that must be taken [26].

2.3.1. Data Normalization

Data normalization is the process of changing raw data values into another form with qualities that are more suited for modeling and analysis. Normalization seeks to guarantee that all genes are measured in the same unit of measurement. As a consequence, it is utilized to prevent the discrepancy between the effect of tiny values and big values that dominate the outcomes. Many methods for data normalizing exist, including min-max and z-score normalization. Eq. (2.1) is used in min-max normalization to determine the value.

$$\hat{V} = \frac{V - min_a}{max_a - min_a} * (new\_max_a - new\_min_a) + new\_min_a \quad \dots (2.1)$$

Where:

V: represents the gene value.

$\boldsymbol{min_a}$ **:** is the minimum original value for any gene.

$\boldsymbol{max_a}$**:** is the maximum original value for any gene.

$\boldsymbol{new\_max_a}$ **and** $\boldsymbol{new\_min_a}$: are the maximum and minimum interval of values [27].

2.4. Data Splitting Stage

Data splitting, which divides a given dataset into training and testing sets, is a frequently used method for model validation. Following that, the training set is used to fit and evaluate the statistical and machine learning models. It can assess and compare the accuracy of various models' predictions without being concerned about the potential overfitting of the training set if it Hold-out a set of data for validation apart from the training set [28]. The most popular method for data splitting is random subsampling, which involves randomly selecting some rows from the dataset for training and some for testing without replacing them. There are other deterministic

approaches for splitting that aim to disperse the testing set in a way that better covers the original dataset's coverage area than a random testing set [29].

Once they can define a splitting ratio, the aforementioned data splitting techniques can be used. A typical ratio is 80:20, which denotes that 80% of the data is utilized for training and 20% is used for testing. In practice, other ratios like 70:30, 60:40, and even 50:50 are also employed. On what ratio is ideal or optimal for a particular dataset, there is no clear information. The well-known Pareto principle serves as the basis for the 80:20 split, but again, this is only a practice-based guideline. There is currently no agreement on the best data splitting ratio based on theoretical or numerical research [30].

Because it offers good performance, the K-Fold Cross Validation (k-FCV) approach is the most widely used for data splitting methodology. The data is divided into k equal sections (called k-fold). The remaining portion (the k-1 part) is used as the training set, while one part is reserved as the validation set. Each component is used as a validation set once after this process has been carried out k times. The predicted performance from these experiments is then averaged [31]. Fig. (2.3) show the two types of data splitting [32].

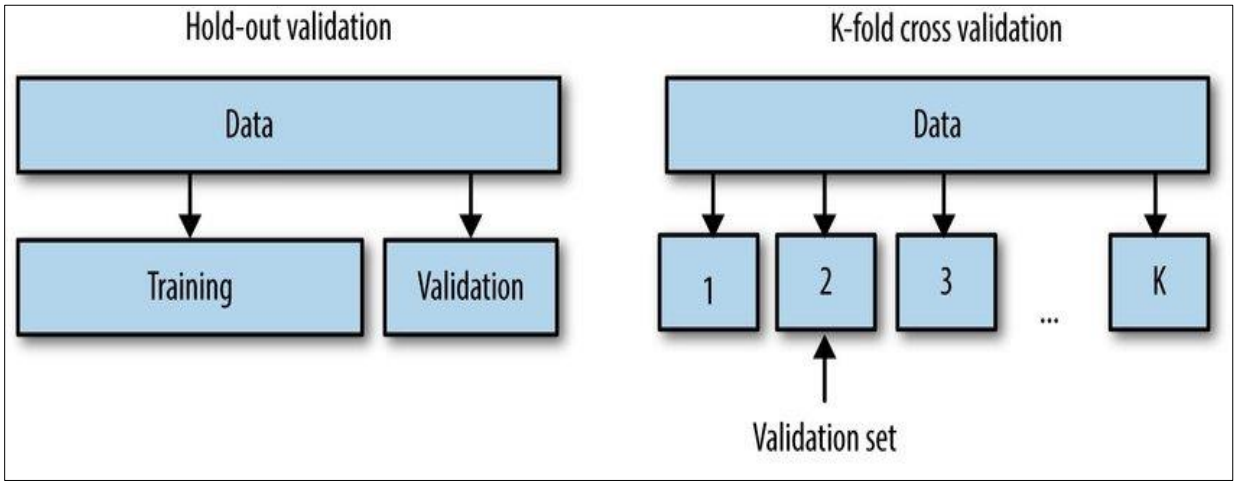

Figure (2.3). Hold-out y K-fold cross-validation [32].

2.5. Features (Selection, Extraction, and Reduction)

The features on which conventional Deep Learning (DL) algorithms are applied typically determine their effectiveness. In other words, the characteristics utilized to characterize each item in the DL models have an impact on their performance in addition to the parameters. This section gives an overview of the fundamental ideas and the reasons behind these strategies [33]

2.5.1. Features Reduction

Feature reduction strategies actually provide a wide range of advantages, including:

1. Enhancing the performance of learning algorithms in terms of learning speed or generalization ability.
2. Decreasing the storage required. The amount of information that is useful for the learning process is retained while the original material is reduced. As a result, in order to create reliable DL models in a reasonable amount of time for a dataset with a large number of features, data reduction is essential. As illustrated in Figure (2.4), it can be separated into feature selection (also known as feature elimination) and feature extraction.

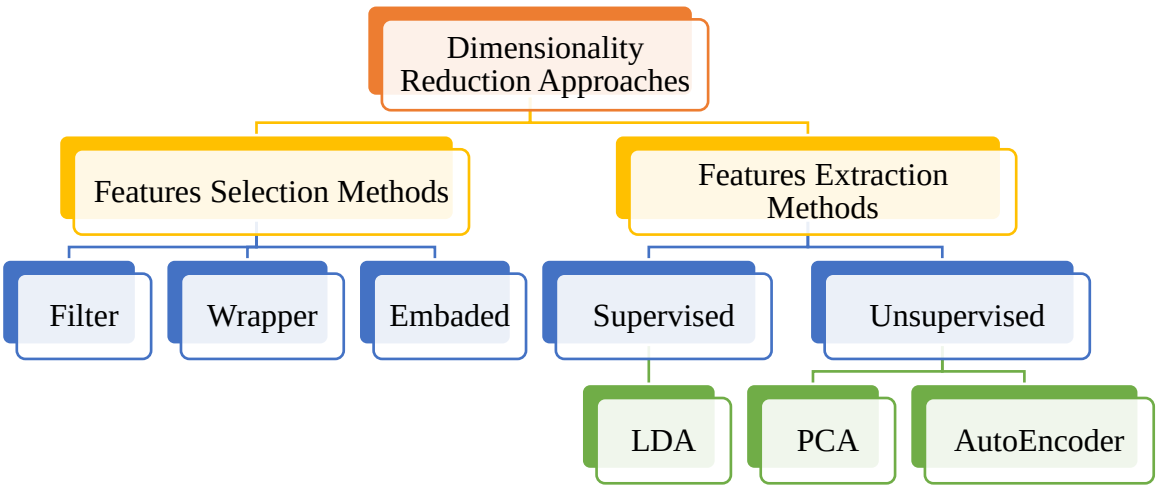

Figure (2.4). The classification of dimensionality reduction techniques [33].

The main difference between feature selection and feature extraction is that the former choices a subset of original features while the latter generates new features from the originals. The benefits and drawbacks of each strategy are shown in Table (2.2) [34].

Table (2.2). Comparison of dimensionality reduction techniques [34].

Technique	Advantages	Disadvantages	Methods
Filter	Low computational cost, fast, scalable	Ignores the interaction with the classifier	Correlation-based Feature Selection (CFS), Information Gain (IG), Fast Correlation Based Filter (FCBF)
Wrapper	Competitive classification accuracy, interaction with the classifier	Slow, expensive for large feature space, risk of over-fitting	Forward/backward direction
Feature extraction	Reduces dimension without losing information	There is no information about the original features.	Principal Components Analysis (PCA), Linear Discriminant Analysis (LDA), Acoustic Emission (AE)

2.5.2. Feature Selection

The feature selection technique is one of the most straightforward ways to minimize data size and has grown to be an essential part of deep learning algorithms. In order to "optimally" describe the target variable, it seeks to select a subset of features. The process of choosing the top features from a given initial set of features in order to improve classification performance, improve regression results, and locate clusters quickly is known as feature selection. The selection of features may be divided into two major categories: filters and wrappers [35].

2.5.3. Feature Extraction

The mapping function F is used in feature extraction to turn the original variables into new features while keeping the majority of the important data. The original attributes can be combined in a linear or non-linear way to produce this transformation. The feature space can frequently be reduced using the feature extraction technique without losing a lot of data from the original attribute space. One of its drawbacks, though, is that it frequently loses track of how the original characteristics contribute. Additionally, it is challenging to establish a connection between the original features and the new features. As a result, it is nearly impossible to analyze the new characteristics because feature extraction tools cannot derive a physical meaning for the changed data. This study covers two of the most popular feature extraction techniques [36].

2.5.3.1. Principal Component Analysis (PCA)

It is the earliest method of multivariate analysis, and Karl Pearson first proposed it in 1901, by condensing the initial feature space into a smaller space, it is an

unsupervised and non-parametric strategy that decreases the dimensionality of data from f to p, where p<f. PCA has several restrictions, which are described below [37]:

1. There is no ability to know how many PCs should be retained (the ideal number of principal components (PCs));
2. It ignores the correlation between target outputs and input features;
3. It depends on the scaling of the data (i.e., each variable is normalized to zero means);
4. It assumes that relationships between variables are linear.

The steps for PCA are described as follows [38]:

1. Let X be a PCA input matrix made up of an n-vector with an m-dimensional data set.
2. Calculate the mean data ($\overline{X}$) of each dimension using Eq. (2.2):

$$\overline{\mathrm{X}} = \frac{1}{n}\sum_{i=1}^{n} X_i \quad \ldots (2.2)$$

   Where: n is the number of samples, and X_i is the value of item i.

3. Calculate the covariance matrix (C_x) using the following Eq. (2.3):

$$C_x = \frac{1}{n-1}\sum_{i=1}^{n}(X_i - \overline{\mathrm{X}})\ (X_i - \overline{\mathrm{X}})^{T} \quad \ldots (2.3)$$

4. Calculate the eigenvectors (v_{m}) and eigenvalues (λ_{m}) of the covariance matrix using Eq. (2.4):

$$C_x v_{\mathrm{m}} = \lambda_{\mathrm{m}} v_{\mathrm{m}} \quad \ldots (2.4)$$

5. Sort the eigenvalues in descending order.
6. Principal component (PC) is a collection of eigenvectors corresponding to the sorted eigenvalues in step 5.

7. PC dimension will be reduced based on the eigenvalues.

2.5.3.2. Singular Value Decomposition (SVD)

The SVD technique detects and arranges the dimensions that fluctuate the most between data points. This relates to the third way to consider SVD: it may achieve the best estimate of the raw datasets with fewer dimensions after identifying the regions with the greatest variance. SVD shown in Fig. (2.5) may therefore be viewed as a data reduction technique [39]. One of the most essential linear algebra principles is singular value decomposition. The SVD method's goal is to diagonalizable the data matrix (X∈ $R^{p\times q}$) into three matrices, as shown in Eq. (2.5).

$$X = LSR^T = [l_1 \dots \quad l_p] \begin{bmatrix} s_1 & 0 & 0 & 0 \\ 0 & s_2 & 0 & 0 \\ 0 & 0 & \ddots & 0 \\ 0 & 0 & 0 & s_q \\ \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} -r_1^T \\ -r_2^T \\ \vdots \\ -r_q^T \end{bmatrix} \quad \dots (2.5)$$

Where L(p × p) are the left singular vectors, and S (p × q) is a diagonal matrix that depicts the singular values sorted from high to low, i.e. the biggest singular value in the upper-left index of S, thus $s_1 \geq s_2 \geq \dots \geq s_q \geq 0$, and R (q × q) depicts the right singular vectors. The orthonormal bases are the left and right singular matrices, denoted by L and R [40].

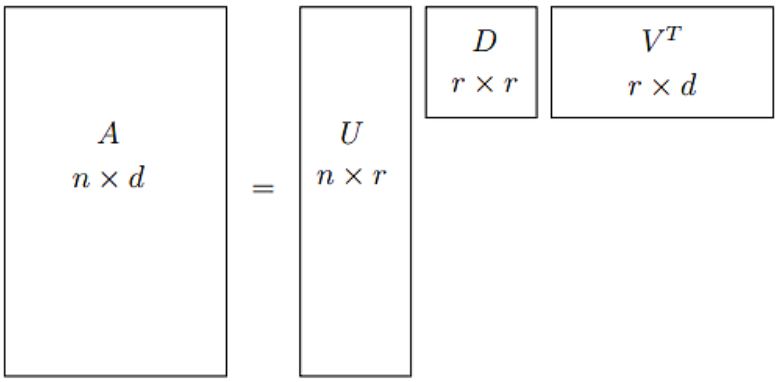

Figure (2.5). The SVD decomposition of an n × d matrix [39].

2.6. Deep Learning Models

In order to address the issues with standard machine learning in the classification of medical data, deep learning has emerged as the preferred way of analysis thanks to the ongoing advancement of technology. The deep learning approach is popularly employed in the analysis of medical images after learning from the field of computer vision. Unsupervised learning, supervised learning, and other types of learning are all part of the deep learning architecture [41].

2.6.1. Unsupervised Deep Learning

Unsupervised learning uses unlabeled training data. The objective of the training is to categorize or separate the observed values. The deep model is highly helpful for extracting information features from data in a hierarchical fashion as opposed to the shallow architecture of a well-designed feature extractor, which necessitates specialist knowledge and human design. A further division of unsupervised learning is into AE and Restricted Boltzmann Machine (RBM) [42].

2.6.2. Supervised Deep Learning

The supervised learning approach is more popular than unsupervised learning. Typical networks used in supervision techniques include Deep Neural Networks (DNN), Convolutional Neural Networks (CNN), and Recurrent Neural Networks (RNN). CNN is the most effective deep model of them all. In addition, there are numerous established CNN models, including VGGNet, AlexNet, ResNet, DenseNet, and Inception, that can conduct effective AD detection. Related researchers can set up a proprietary architecture depending on CNN [43].

2.6.2.1. Convolutional Neural Networks (CNN)

Convolutional neural networks (CNN) are a kind of artificial neural network that has dominated a number of computer vision tasks and are gaining popularity in a number of fields, including radiology. Using a variety of building pieces, including convolution layers, pooling layers, and fully connected layers, CNN is intended to automatically and adaptively learn spatial hierarchies of features through backpropagation [44].

A. Convolutional Neural Network (CNN) Architecture

Building components like convolution layers, pooling layers, and fully connected layers are all part of the CNN architecture. The recurrence of a stack of numerous convolution layers and a pooling layer, followed by one or more fully connected layers, makes up a common design. Forward propagation refers to the process where input data are turned into output through these levels Fig. (2.6). Although the convolution and pooling methods discussed in this section are for a two-dimensional (2D) CNN, a three-dimensional (3D) CNN can also execute the same activities [45].

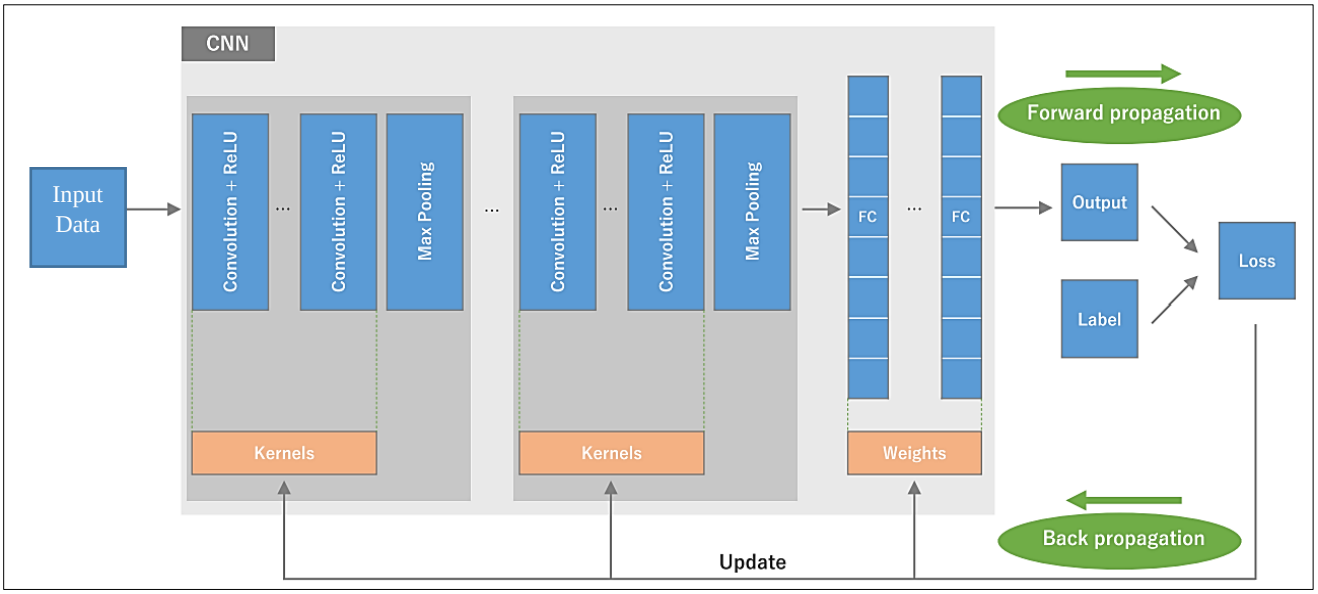

Figure (2.6). A convolutional neural network (CNN) architecture [45].

1. Convolutional Layer

The convolutional layer, which uses convolution kernels to extract features from input images, is the brain of a CNN model. With this process, the following layer's pixels can be converted into a local receptive field, which is the connected area of any convolution kernel applied to the input data. A specific kind of linear operation called convolution is used to extract features. It applies a tiny array of numbers, known as a kernel, over the input, which is an array of numbers known as a tensor. A feature map, also known as an output value in the corresponding place of the output tensor, is obtained by computing an element-wise product between each element of the kernel and the input tensor at each point of the tensor and summing it [46].

2. Nonlinearity Layer

The non-saturating activation function is used in this layer. While having no impact on the convolution layer's receptive fields, it will boost the nonlinear features of the choice function and the overall network that are desirable for multi-layer networks. The most commonly used functions are [47]:

- **Rectified Linear Unit function (ReLU):** For convolution layers, ReLU is the most popular activation function. It is defined mathematically as Eq. (2.6):

$$f(x) = max(0.x) = \begin{matrix} 0 & if\ x < 0 \\ 1 & if\ x \geq 0 \end{matrix} \quad \ldots (2.6)$$

- **LeakyReLU:** It is an attempt to solve the dying ReLU problem. Instead of the function being zero when x < 0, a leaky ReLU will instead have a small negative slope (of 0.01, or so). The function computes this shown in Eq. (2.7) where α is a small constant.

$$f(x) = 1(x < 0)(\alpha x) + 1(x >= 0)(x) \quad \ldots (2.7)$$

- **Sigmoid function:** its curve looks like an S-shape. The function varies between [0, 1], therefore it is used to predict a probability as an output. Mathematically it has the form in Eq. (2.8) as follows:

$$f(x) = \frac{1}{1+ e^{-x}} \quad \ldots (2.8)$$

- **Hyperbolic Tangent (tanh) function:** Tanh has a similar shape to Sigmoid, however, its range is [1, 1]. The benefit is that zero values will be mapped close to zero, whereas negative values will be mapped substantially negative. It is mathematically defined in Eq. (2.9):

$$f(x) = \tanh(x) = \frac{2}{1+ e^{-2x}} - 1 \quad \ldots (2.9)$$

3. Pooling layer

By lowering the spatial dimension of the convolutional outputs, it lowers the network's parameter count. The invariant representation of minor input translations is also a benefit of pooling procedures. The two primary pooling operations are detailed in the sections that follow, and an example of pooling operations employing a 2 x 2 filter is shown in Fig. (2.7) [48].

- **Max pooling:** It computes the greatest value for each input patch. By swiping the filter over the feature map, the max-pooling layer saves the maximum value of each patch. Mathematically, it takes the following form in Eq. (2.10) [49]:

$$f_{max}\,(A) \;=\; max_{n\times m}\,(A_{n\times m}) \quad \ldots (2.10)$$

  Commonly, in the max pooling layer, 2 × 2 filters are applied with a stride of 2. It downsamples the input by 2 along its dimensions and discards 75% of the convolutional outputs.

- **Average pooling:** It computes the average value for each input patch. The average pooling layer downsamples the convolutional activation by splitting it

into pooling areas and determining the average values of those regions. In Eq. (2.11), it is formally defined as follows [50]:

$$f_{max}\ (A) = \frac{1}{n+m}\sum_{i=1}^{n}\sum_{k=1}^{m}(A_{i.k}) \quad \ldots (2.11)$$

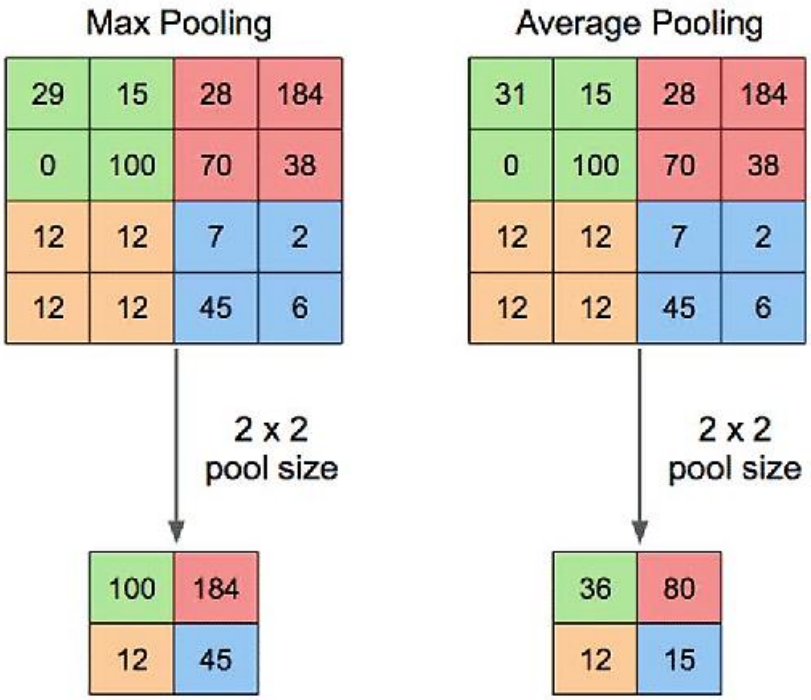

Figure (2.7). Examples of pooling operations by using 2 × 2 filters applied with a stride of 2 [48].

4. Fully Connected (FC) Layer

The convolutional stages' final output is flattened to a 1D array and joined to a fully connected layer. Like a regular neural network, fully connected layers use the results of the convolution/pooling process to categorize the image into a label (i.e., class). As a result, the activation function of the final layer (i.e., output layer) computes the final probabilities for each class and is chosen based on the task. A multi-class classification job often employs the softmax function, where each class probability value spans between [0, 1] and their total sum equals 1. Finally, each output neuron chooses a label, and the highest output value corresponds to the classification decision [51].

B. CNN-1D VS CNN-2D

According to much research, 1D CNNs are beneficial and hence superior to their 2D counterparts in dealing with 1D signals for the following reasons [52]:

1. The computational difficulties of 1D and 2D convolutions differ significantly, i.e., an image with N×N dimensions convolving with K×K kernel would have a computational difficulty, but the corresponding 1D convolution will not (with the same dimensions, N and K). This suggests that the computational complexity of a 1D CNN is much lower than that of a 2D CNN under identical conditions (same design, network, and hyper-parameters).
2. In general, most 1D CNN applications have utilized compact (with 1-2 buried CNN layers) setups with networks containing 10 K parameters, but practically all 2D CNN applications have employed "deep" architectures with more than 1 M (typically exceeding 10 M) parameters. Networks with shallow architectures are obviously easier to train and implement.
3. Deep 2D CNN training typically necessitates the use of specialized hardware (e.g. Cloud computing or GPU farms). For training compact 1D CNNs with few hidden layers (e.g., 2 or less) and neurons (e.g., 50), any CPU implementation on a typical computer is practical and relatively quick.
4. Compact 1D CNNs are highly suited for real-time and low-cost applications, particularly on mobile or handheld devices, because of their cheap processing needs.
5. Compact 1D CNNs have shown higher performance in situations with little labeled data and large signal fluctuations gathered from various sources (i.e., patient ECG, civil, mechanical, or aerospace structures, high-power circuitry, power engines or motors, etc.).

C. Advantages of Convolutional Neural Network

CNN appears to be a good fit for multidimensional imagery processing for classification and regression. These advantages are [53]:

1. It can have arbitrary rough functions with a spatial context.
2. It excels at learning spatial connections from image data, enabling the model to learn locations and scales in a wide range of structures.
3. It can be fed high-dimensional pattern images as inputs and extract complex features from the imagery data, boosting the explanatory and predictive ability of the neural network. CNN's usefulness for prediction tasks can be expanded when images are supplied at a high temporal frequency.
4. CNN is distinguished from typical multilayer visual neural networks by its capacity to learn multiscale spatial patterns from gridded input from several sources. CNN uses a convolution process as a kernel to search the input image in each dimension.
5. Because it is less costly than other nonlinear functions and has previously been demonstrated to considerably boost CNN training time, the rectified linear unit (ReLU) function is a preferred activation function for hidden CNN layers.
6. To construct sparse connections, local pattern sin images are employed.
7. Using the same filter, weights are shared throughout an entire input picture (resulting in translation of equivariant).
8. The pooling procedure results in local shift invariance.

D. Limitation of Convolutional Neural Network

The restriction of CNN has been highlighted owing to the requirement of a large amount of data. Interpreting data processing in a particular broad watershed region may be a waste of effort [54].

2.7. Performance Criteria

In the field of deep learning, classification problems, especially to measure the performance of recommendation systems, or to understand the generalization ability of models, are often verified by sensitivity (Recall), specificity, precision, accuracy, and F-score to compare different models as follows [55]:

TP = True positives are the number of cases that were anticipated to be positive but turned out to be true.

FP = False positives: the number of cases that were expected to be positive but turned out to be negative.

TN = True negatives are the number of cases that were expected to be negative but turned out to be true.

FN = False negatives: the number of cases that were expected to be negative but turned out to be positive.

2.7.1. Accuracy

The percentage of instances properly classified out of all those presented. It's computed as follows:

$$\text{Accuracy} = \frac{\text{TP+TN}}{\text{TP+TN+FB+FN}} \quad \ldots (2.12)$$

2.7.2. Precision

For all those identified as class x, the proportion of true x-class occurrences. It is computed as follows:

$$\text{Precision} = \frac{\text{TP}}{\text{TP+FP}} \quad \ldots (2.13)$$

2.7.3. Recall or (Sensitivity)

It represents the proportion of all positive samples that the model judges to be truly positive, and it measures the classifier's ability to recognize positive samples. It is computed as follows:

$$\text{Recall} = \frac{\text{TP}}{\text{TP+FN}} \quad \ldots (2.14)$$

2.7.4. F- measure

It is the harmonic mean of precision and recall. It is calculated as:

$$\text{F}_1 = 2 * \frac{\text{precision} * \text{recall}}{\text{precision+recall}} \quad \ldots (2.15)$$

2.7.5. Loss (Error rate)

An error is simply a misclassification: the classifier has presented a case, and it classifies the case incorrectly, as shown in Eq. (2.16) below:

$$\text{Error Rate} = 1 - \text{accuracy} \quad \ldots (2.16)$$

Chapter Three
The Proposed System

Chapter Three

The Proposed System

3.1. Introduction

In this chapter, the steps followed to achieve the major aims of this thesis are described. This includes preprocessing and feature reduction phase as a primary phase before the system predicts AD based on several proposed models of deep Convolution Neural Networks (CNN). The architecture of this system is depicted first. Then, data pre-processing is discussed to prepare data for subsequent processes, data (gene) selection is utilized to select a gene subset from the original data, and then the classification models are discussed subsequently.

3.2. The Proposed System Architecture

The five primary stages of the proposed system architecture (data preprocessing, data splitting, feature reduction, classification, and evaluation) are necessary to accomplish the thesis's objectives. The next sub-sections give a thorough discussion of various phases, while this part provides a basic overview. The first stage of data preparation is normalization. Second, in the feature reduction step, the Principal Component Analysis (PCA) and Singular Value Decomposition (SVD) are used.

These strategies aid in reducing the number of characteristics (genes) needed in the classification phase and identifying the most significant ones. Third, the feature extraction step entails employing the suggested CNN model to do this task. Fourth, three Convolutional Neural Networks were used to build and run the three classification models (CNN-1D). Finally, the suggested model's findings were tested using various metrics. The main structure of the suggested system is depicted in Figure (3.1)

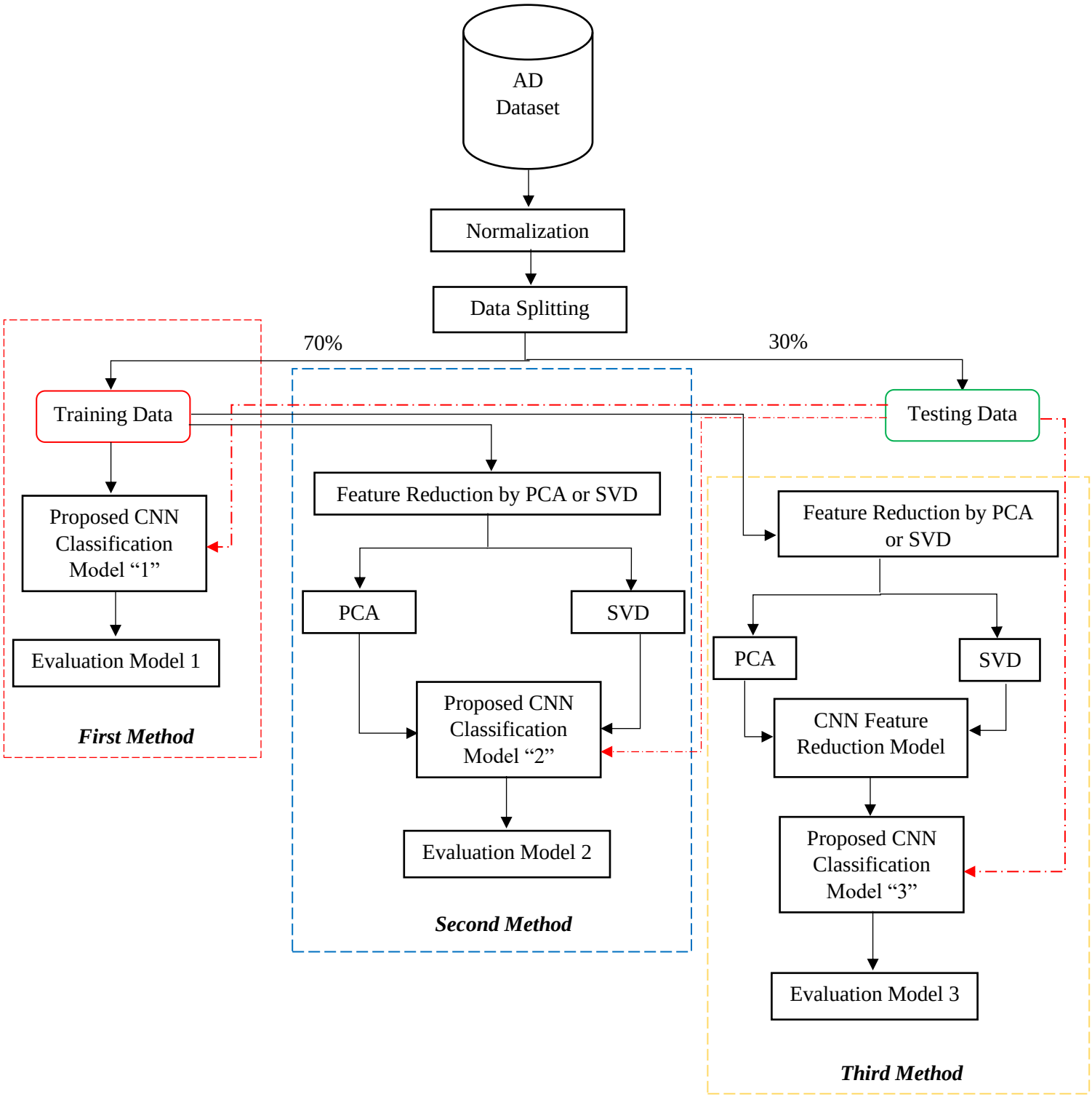

Figure (3.1). An overall diagram of the proposed system.

3.2.1. AD Data Preprocessing (1st Stage)

The first stage of the proposed system is preprocessing, which seeks to turn the raw dataset into a simple and efficient format. As a result, it is a lengthy approach with the primary purpose of producing a dataset that is reliable and suitable for deep learning algorithms. In this stage, the normalizing strategy is used to complete this level.

- **Apply the Normalization Method**

The normalizing procedure was used for all numeric gene values that would be used as input for the deep learning models. It was used on the AD dataset to exclude genes with big values that influence the computation results. The Min-Max normalization approach, described in Eq. (2.1) is utilized to normalize all gene values to a set range between zero and one. As a result, the gene values are scaled so that the smallest value for each gene will be zero and the greatest value will be one.

3.2.2. Splitting Data Stage (2nd Stage)

The Hold-out-validation method was used for ensuring correct generalization and minimizing overtraining. The AD dataset was divided into two subsets: one for training (70%) and one for testing (30%) the final model results. The details of this procedure are explained in Algorithm (3.1).

Algorithm (3.1). AD Dataset Splitting
Input: AD dataset
Output: Splitting dataset (70% training set, 30% testing set)
Begin **1:** Define sets of model parameter values to evaluate. **2:** For each parameter set do **3:** For each resampling iteration do

4:	Hold-out specific samples
5:	Fit the model on the remainder
6:	Predict the hold-out samples
7:	End for
8:	Determine the average performance of hold-out predictions
9:	End for
10:	Determine the optimal parameter set.
11:	Using the optimal parameter set, fit the final model to all of the training data.
End	

3.2.3. Feature Reduction Stage (3rd Stage)

Feature reduction or dimensionality reduction is the method of reducing the number of features in a resource-intensive calculation without sacrificing vital information. As the number of qualities drops, so does the number of variables, making the computer's task easier and faster. Feature selection and feature extraction are the two processes in feature reduction. There are several ways available for feature reduction. This system made use of the following techniques:

3.2.3.1. Principal Component Analysis (PCA)

The PCA is the most frequent unsupervised linear strategy for projecting high-dimensional data into a new lower-dimensional representation of the data that reflects as much of the variance in the data as possible with the least degree of reconstruction error. The algorithm (3.2) explains the steps of the PCA technique as follows:

Algorithm (3.2). Principal Component Analysis (PCA)

Input: Original data

Output: Reconstruct data

Begin

1: Set N data points $x_i = (x_{1i}, x_{2i}, ..., x_{Mi})$ as row vectors.

2: Put these vectors into a matrix X (which will have size N × M).

3: Centre the data by subtracting off the mean of each column, putting it into matrix B_{AD}.

4: Compute the covariance matrix $C_{AD} = \frac{1}{n} {B_{AD}}^T B_{AD}$

5: Compute the eigenvalues and eigenvectors of C_{AD}, so $V^{-1} C_{AD} V = D_{AD}$, where V holds the eigenvectors of C_{AD} and D_{AD} is the M × M diagonal eigenvalue matrix.

6: Reject those with eigenvalue less than some η, leaving L dimensions in the data.

End

3.2.3.2. Singular Value Decomposition (SVD)

The SVD method detects and arranges the dimensions that most fluctuate between data points. It can produce the best approximation of the original data points with fewer dimensions after detecting the biggest variance. As a consequence, SVD might be regarded as an effective strategy for decreasing features in the AD dataset. The algorithm (3.3) describes how this strategy works.

Algorithm (3.3). Singular Value Decomposition (SVD)

Input: An m x n matrix A, and a number k

Output: Approximate U_k, V_k and Σ_k

Begin

1: Generate an n × k Gaussian matrix G

2: Compute Y= A × G

3: Compute an orthogonal column basis Q of Y

4: Form B= $Q^T \times$ A

5: Compute eigendecomposition of B$B^T = X\Sigma^2 \times X^T$

6: $U_k = QX$, $V_k = B^T X\Sigma^{-1}$ and $\Sigma_k = \Sigma$

End

3.2.3.3. CNN Feature Reduction

In the proposed system, a new method for feature reduction based on deep learning was proposed, where a Convolutional Neural Network (CNN) feature extraction model was designed to perform this task away from traditional feature reduction methods in an attempt to prepare data in a distinctive way before entering the classification stage, for the purpose of increasing accuracy and reducing classification time. Fig. (3.2) shows the structure of this network and Algorithm (3.4) shows the layers of building this network.

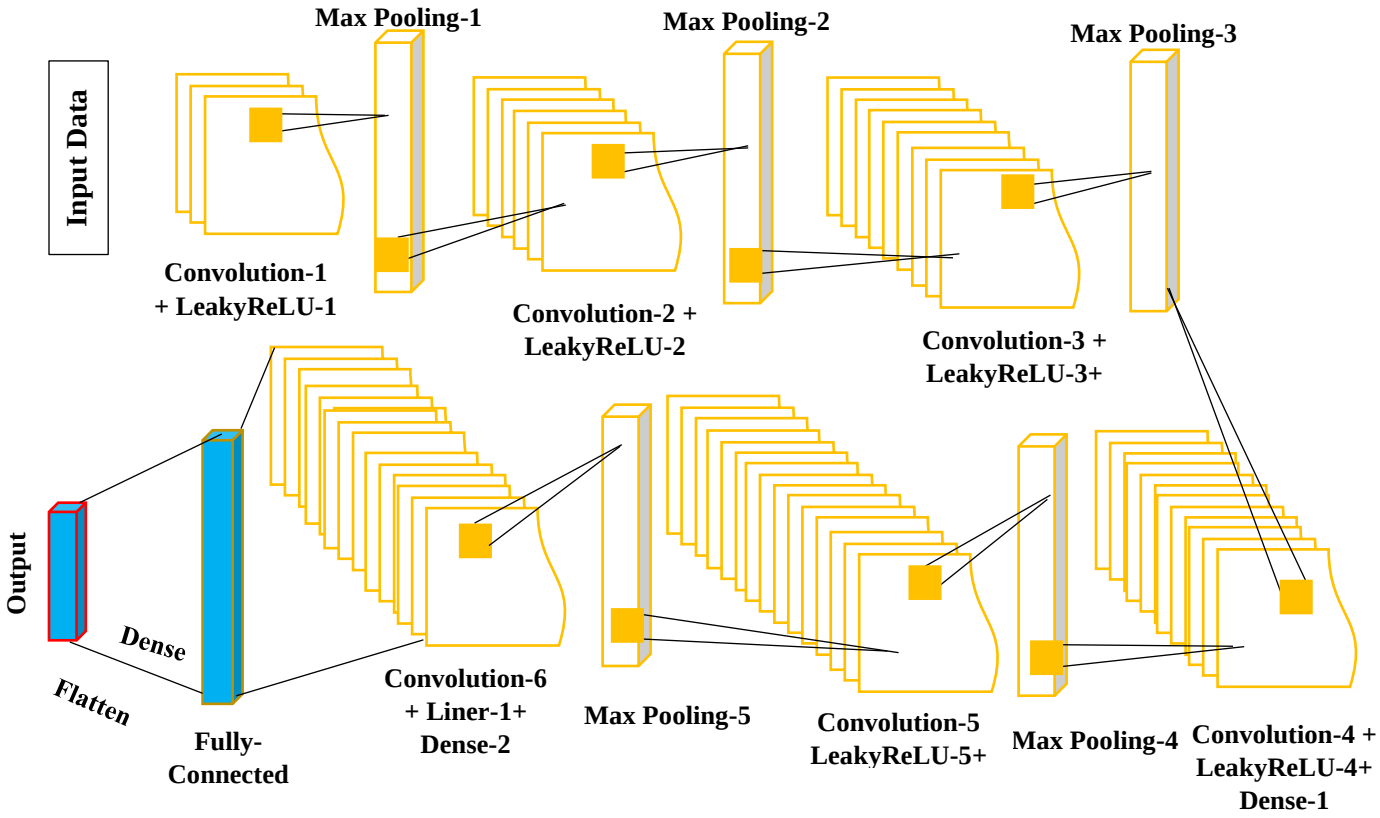

Figure (3.2). The Proposed CNN feature reduction model.

The proposed CNN feature reduction model consists of 20 layers as follows:

- Convolutional (6) layers.
- Max Pooling (5) layers.
- Leaky ReLU (5)
- Flatten (1) layer
- Dense (3) layer.

Table (3.1) explains these layers in some detail.

Table (3.1). Proposed CNN feature reduction model layers

NO.	Layer Type	Filters	Kernel Size/Stride	Activation Function
1	Convolutional	16	3/1	–
2	Max Pooling	–	1/1	–
3	Leaky ReLU	–	–	–
4	Convolutional	32	3/1	–
5	Max Pooling	–	1/1	–
6	Leaky ReLU	–	–	–
7	Convolutional	32	3/1	–
8	Max Pooling	–	1/1	–
9	Leaky ReLU	–	–	–
10	Dense	–	–	Linear
11	Convolutional	32	3/1	–
12	Max Pooling	–	1/1	–
13	Leaky ReLU			
14	Convolutional	32	3/1	
15	Max Pooling	–	1/1	–
16	Leaky ReLU	–	–	–

17	Dense	–	–	Linear
18	Convolutional	40	3/1	–
19	Flatten	–	–	–
20	Dense	–	–	Softmax

Algorithm (3.4). Feature Reduction CNN Model

Input: Data with reduced features

Output: Extracted features by CNN model

Begin

1: Set features that have been reduced using both PCA and SVD techniques.

2: Set the input shape value to 10.

3: Build the CNN model with the following layers:

- Convolution layer as the layer-1 with a kernel size is 3, the padding equals 1, the stride equals 1, and the filters no. are 16.
- Max pooling layer as the layer-2 with pool size is 1 and stride equal to 1.
- Leaky ReLU as the layer-3 with alpha=0.3.
- Convolution layer as the layer-4 with a kernel size is 3, the padding equals 1, the stride equals 1, and the filters no. are 32.
- Max pooling layer as the layer-5 with pool size is 1 and stride equal to 1.
- Leaky ReLU as the layer-6 with alpha=0.3.
- Convolution layer as the layer-7 with a kernel size is 3, the padding equals 1, the stride equals 1, and the filters no. 32.
- Max pooling layer as the layer-8 with pool size is 1 and stride equal to 1.
- Leaky ReLU as the layer-9 with alpha=0.3.
- Dense layer with liner activation function as the layer-10 with filter size=32.

- Convolution layer as the layer-11 with a kernel size is 3, the padding equals 1, the stride equals 1, and the filters no. 32.
- Max pooling layer as the layer-12 with pool size is 1 and stride equal to 1.
- Leaky ReLU as the layer-13 with alpha=0.3.
- Convolution layer as the layer-14 with a kernel size is 3, the padding equals 1, the stride equals 1, and the filters no. 32.
- Max pooling layer as the layer-15 with pool size is 1 and stride equal to 1.
- Leaky ReLU as the layer-16 with alpha=0.3.
- Dense layer with linear activation function as the layer-17 with filter size=64.
- Convolution layer as the layer-18 with a kernel size is 3, the padding equals 1, the stride equals 1, and filters no. 40 with linear activation function.
- Flatten layer as the layer-19.
- Dense layer with softmax activation function as the layer-20 with filter size=3.

4: CNN features = Return (extracted features)

End

3.2.4. Classification Stage (4th Stage)

In this section, the three proposed Convolutional Neural Networks (CNNs) for AD data classifying in this system will be explained as follows:

3.2.4.1. First Method (Pure Data)

The suggested CNN model is utilized to categorize data immediately after the data set is imported and normalized in this manner. In this method, the proposed CNN

model is used to classify data immediately after the dataset is loaded, spilled, and normalized without any other operation. The suggested CNN classification model "1" structure is similar to the structure of the CNN feature reduction model in Fig. (3.2). Algorithm (3.5) illustrates this model in more detail. The proposed CNN Classification Model "1" consists of 20 layers as follows:

- Convolutional (6) layers.
- Max Pooling (5) layers.
- Leaky ReLU (5) layers.
- Flatten (1) layer
- Dense (3) layers.

The layers of this classification method are similar to the layers mentioned in Table (3.1).

Algorithm (3.5). The proposed CNN classification model "1"

Input: Split and normalized AD dataset.

Output: Accuracy of CNN classification model "1"

Begin

1: Load AD data set.

2: Normalized dataset, call Algorithm (3.1).

3: Splitting dataset, call Algorithm (3.2).

4: **Training dataset:**

Build the classification CNN model "1" with the following layers:

- Convolution layer as the layer-1 with a kernel size is 3, the padding equals 1, the stride equals 1, and the filters no. 16.
- Max pooling layer as the layer-2 with pool size is 1 and stride equal to 1.
- Leaky ReLU as the layer-3 with alpha=0.3.

- Convolution layer as the layer-4 with a kernel size is 3, the padding equals 1, the stride equals 1, and the filters no. 32.
- Max pooling layer as the layer-5 with pool size is 1 and stride equal to 1.
- Leaky ReLU as the layer-6 with alpha=0.3.
- Convolution layer as the layer-7 with a kernel size is 3, the padding equals 1, the stride equals 1, and the filters no. 32.
- Max pooling layer as the layer-8 with pool size is 1 and stride equal to 1.
- Leaky ReLU as the layer-9 with alpha=0.3.
- Dense layer with liner activation function as the layer-10 with filter size=32.
- Convolution layer as the layer-11 with a kernel size is 3, the padding equals 1, the stride equals 1, and the filters no. 32.
- Max pooling layer as the layer-12 with pool size is 1 and stride equal to 1.
- Leaky ReLU as the layer-13 with alpha=0.3.
- Convolution layer as the layer-14 with a kernel size is 3, the padding equals 1, the stride equals 1, and the filters no. 32.
- Max pooling layer as the layer-15 with pool size is 1 and stride equal to 1.
- Leaky ReLU as the layer-16 with alpha=0.3.
- Dense layer with linear activation function as the layer-17 with filter size=64.
- Convolution layer as the layer-18 with a kernel size is 3, the padding equals 1, the stride equals 1, and the filters no. 40 with linear activation function.
- Flatten layer as the layer-19.
- Dense layer with softmax activation function as the layer-20 with filter size=3.

4: Test the remainder of the data that have been entered for training.

5: Return (accuracy)

End

3.2.4.2. Second Method (PCA or SVD with CNN)

In this method, after dividing the data, its features are reduced using one of the two methods to complete this task PCA or SVD. In this part, the steps of the second proposal are explained, which combines PCA or SVD with the proposed CNN classification model "2". Figure (3.3) depicts the suggested CNN classification model "2" structure. Algorithm (3.6) illustrates the CNN model layers in more detail.

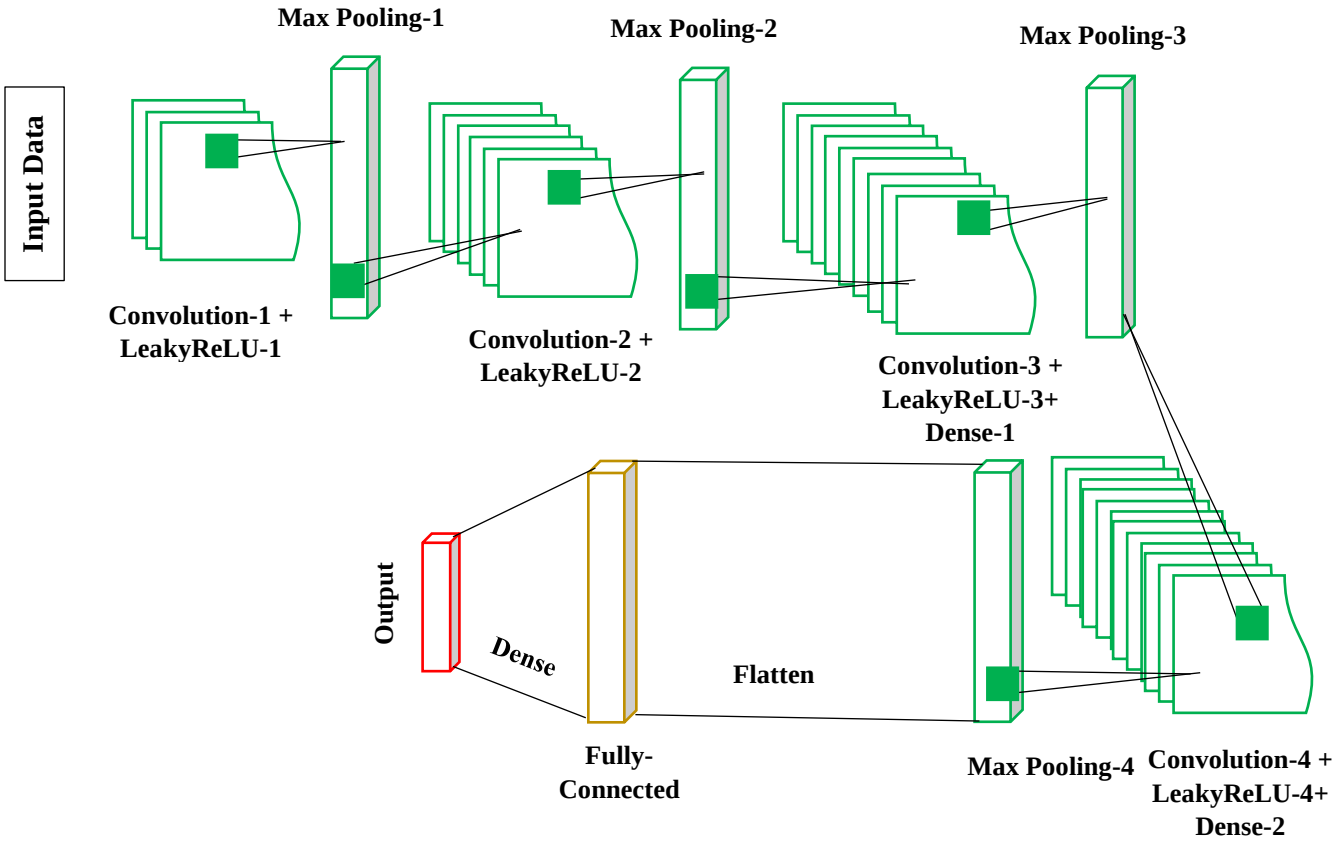

Figure (3.3). The Proposed CNN Classification Model "2"

The proposed CNN Classification Model "2" consists of 16 layers as follows:

- Convolutional (4) layers.
- Max Pooling (4) layers.
- Leaky ReLU (4) layers.
- Flatten (1) layer
- Dense (3) layer.

Table (3.2) explains these layers in some detail.

Table (3.2). Proposed CNN classification model "2" layers

NO.	Layer Type	Filters	Kernel Size/Stride	Activation Function
1	Convolutional	16	3/1	–
2	Max Pooling	–	1/1	–
3	Leaky ReLU	–	–	–
4	Convolutional	32	3/1	–
5	Max Pooling	–	1/1	–
6	Leaky ReLU	–	–	–
7	Dense	–	–	Linear
8	Convolutional	64	3/1	–
9	Max Pooling	–	1/1	–
10	Leaky ReLU	–	–	–
11	Convolutional	128	3/1	–
12	Max Pooling	–	1/1	–
13	Leaky ReLU	–	–	–
14	Dense	–	–	Linear
15	Flatten	–	–	–
16	Dense	–	–	Softmax

Algorithm (3.6). The proposed CNN classification model "2"

Input: AD dataset

Output: Accuracy of CNN classification model "2"

Begin

1: Load AD data set.

2: Normalized dataset, call Algorithm (3.1).

3: Splitting dataset, call Algorithm (3.2).

4: Reduce features using PCA, call Algorithm (3.3), or using SVD, call Algorithm (3.4).

5: Build classification CNN model "2" with the following layers:

- Convolution layer as the layer-1 with a kernel size is 3, the padding equals 1, the stride equals 1, and the filters no. 16.
- Max pooling layer as the layer-2 with pool size is 1 and stride equal to 1.
- Leaky ReLU as the layer-3 with alpha=0.3.
- Convolution layer as the layer-4 with a kernel size is 3, the padding equals 1, the stride equals 1, and the filters no. 32.
- Max pooling layer as the layer-5 with pool size is 1 and stride equal to 1.
- Leaky ReLU as the layer-6 with alpha=0.3.
- Dense layer with liner activation function as the layer-7 with filter size=64.
- Convolution layer as the layer-8 with a kernel size is 3, the padding equals 1, the stride equals 1, and the filters no. 64.
- Max pooling layer as the layer-9 with pool size is 1 and stride equal to 1.
- Leaky ReLU as the layer-10with alpha=0.3.
- Convolution layer as the layer-11 with a kernel size is 3, the padding equals 1, the stride equals 1, and the filters no. 128.
- Max pooling layer as the layer-12 with pool size is 1 and stride equal to 1.

- Leaky ReLU as the layer-13 with alpha=0.3.
- Dense layer with linear activation function as the layer-14 with filter size=64.
- Flatten layer as the layer-15 with a filter size is 128.
- Dense layer with softmax activation function as the layer-16 with filter size=3.

4: Return (accuracy)

End

3.2.4.3. Third Method (PCA or SVD with Double CNN models)

In this method, after dividing the data, the features are reduced using one of the two methods to complete this task like PCA or SVD. Then, the proposed CNN model is used to extract features (CNN feature extraction model). Another proposed CNN model used to accomplish the classification stage in this method is called CNN classification model "3". The suggested CNN classification model "3" structure is similar to the structure of CNN classification model "2" in Fig. (3.3). Algorithm (3.7) illustrates the CNN model layers in more detail. The proposed CNN Classification Model "3" consists of 16 layers as follows:

- Convolutional (4) layers.
- Max Pooling (4) layers.
- Leaky ReLU (4) layers.
- Flatten (1) layer
- Dense (3) layer.

The layers of this classification method are similar to the layers mentioned in Table (3.2).

Algorithm (3.7). The proposed CNN classification model "3"

Input: AD dataset

Output: Accuracy of CNN classification model "3"

Begin

1: Load AD data set.

2: Normalized dataset using Eq. (2.1).

3: Splitting dataset, call Algorithm (3.1).

4: Reduce features using PCA, Call Algorithm (3.2), or using SVD, Call Algorithm (3.3).

5: Extract features, Call Algorithm (3.4).

6: Build classification CNN model "3" with the following layers:

- Convolution layer as the layer-1 with a kernel size is 3, the padding equals 1, the stride equals 1, and the filter no. 16.
- Max pooling layer as the layer-2 with pool size is 2, the padding is 1, and stride is equal to 2.
- Leaky ReLU as the layer-3 with alpha=0.3.
- Convolution layer as the layer-4 with a kernel size is 3, the padding equals 1, the stride equals 1, and the filter no.32.
- Max pooling layer as the layer-5 with pool size is 2, the padding is 1, and stride equal to 1.
- Leaky ReLU as the layer-6 with alpha=0.3.
- Dense layer with liner activation function as the layer-7 with filter size=64.
- Convolution layer as the layer-8 with a kernel size is 3, the padding equals 1, the stride equals 1, and the filters no. 64.

- Max pooling layer as the layer-9 with pool size is 2, the padding is 1, and stride equal to 2.
- Leaky ReLU as the layer-10with alpha=0.3.
- Convolution layer as the layer-11 with a kernel size is 3, the padding equals 1, the stride equals 1, and the filters no. 128.
- Max pooling layer as the layer-12 with pool size is 2, the padding is 1, and stride equal to 2.
- Leaky ReLU as the layer-13 with alpha=0.3.
- Dense layer with linear activation function as the layer-14 with filter size=64.
- Flatten layer as the layer-15 with filter size is 128.
- Dense layer with softmax activation function as the layer-16 with filter size=3.

4: Return (accuracy)

End

3.2.5. Evaluation Stage (5th Stage)

In this section, the proposed methods are evaluated through the accuracy and time metrics, and the best method is selected on the basis of the lowest time and the highest accuracy. Algorithm (3.8) explain the system evaluation stage in detail.

Algorithm (3.8). The Proposed System Evaluation

Input: AD dataset

Output: An accuracy of (Method "1", Method "2", and "Method "3"), Best CNN model accuracy

Begin

1: Load dataset **// Input//**

2: Pre-Processing phase

- Normalize data using Eq. (2.1).

3: Apply Holdout splitting, Call Algorithm (3.1) // **Training Phase //**

4: Select the CNN model:

If method "1" is selected, then

Go to "5"

Else if method "2" is selected, then

Go to "6"

Else

Go to "7"

End if

5: Classify data by using the CNN model "1", Call Algorithm (3.5) then Go to "8".

6: Apply the second method:

- Perform feature reduction using PCA, Call Algorithm (3.2) or SVD, Call (3.3).
- Classify data by using the CNN model "2", Call Algorithm (3.6) then Go to "8"

7: Apply the third method:

- Perform feature reduction using PCA, Call Algorithm (3.2) or SVD, Call (3.3).
- Reduce features, Call Algorithm (3.4)
- Classify data by using the CNN model "3", Call Algorithm (3.7).

8: Test the remainder of the data.

//Testing Phase//

9: Classify Samples **// Classification Phase //**

- Results (1) = CNN_1 (Features Set, Targets)
- Results (2) = CNN_2 (Features Set, Targets)
- Results (3) = CNN_3 (Features Set, Targets)

10: Best CNN Classifier = High (Accuracy) // **Output**//

11: Calculate the time of execution for each model.

12: Best Method = Highest (accuracy) + lowest (time) //**Evaluation Phase**//

13: **End Procedure**

End

Chapter Four
Experimental Results and Discussion

Chapter Four

Experimental Results and Discussion

4.1. Introduction

The suggested system depicted in chapter three was implemented to address the research objectives depicted in chapter one. The research needs are first given, followed by a brief explanation of the AD dataset. Following that, the experimental results are detailed and discussed in this section.

4.2. The Proposed System Requirements

To achieve its goal, the suggested system requires the following hardware and software:

4.2.1. Software Requirements

The suggested system was built using the Python 3.6 programming language. It is compatible with Microsoft Windows 10.

4.2.2. Hardware Requirements

This work is done on a computer with the following specifications:

- Intel Core i7, CPU 2.20 GHz.
- RAM 16 GB.
- Operating system: 64-bit

4.3. Dataset Description

The dataset used in this study is from Gene Expression Omnibus (GEO) and is made up of two files: “the AD dataset”. This dataset was contributed by the AddNeuroMed Cohort and made available in 2015. The Alzheimer’s Disease dataset comprises multiple genes, and samples pertain to certain characteristics such as:

(tissue, status, ethnicity, age, and gender). It consists of 16382 genes and 569 samples collected from 245 Alzheimer's patients, 142 MCIs, and 182 CTLs. The first row (example) identifies each column (gene). Each row includes a label that describes the patient's current condition, such as whether or not the patient is pregnant (AD or MCI, or CTL). The remaining values in the dataset represent gene expression levels. Table (4.1) offers an overview of the dataset utilized for further information. Fig. (4.1) explains a part of these data.

Table (4.1). AD Dataset Description

Name of dataset	AD dataset
Dataset characteristics	Multivariate
Attribute characteristics	Real
Number of Instances	569
Number of Attributes	16382
Number of Class Labels	3

1	GSMID	Class	LMN_134329	MN_134329	LMN_165120	LMN_165122	LMN_165122	LMN_165122	LMN_165123	LMN_165123
2	GSM1539532	AD	1.488753	-1.896736	-0.364558	-1.236635	-0.985963	-0.627597	-0.379636	-0.780912
3	GSM1539536	AD	-1.473358	0.963272	-0.356819	-0.797054	-0.01895	0.412661	-0.448269	0.707315
4	GSM1539537	AD	-1.010106	0.53814	0.718685	0.430741	0.588384	-0.663173	0.470998	-0.137167
5	GSM1539538	AD	0.612877	-1.605426	2.01994	-0.266875	0.731437	0.05467	-0.755256	-0.780871
6	GSM1539544	AD	2.271395	-1.994185	0.725074	-0.810631	1.486522	-1.356875	-0.4046	-0.252399
7	GSM1539545	AD	-0.873313	0.315973	2.390354	-0.190767	-0.080746	-0.030908	-0.828866	0.640168
8	GSM1539548	AD	-1.211998	-0.113137	1.339286	-1.285917	1.406242	0.239321	0.163996	0.672617
9	GSM1539553	CTL	-1.350287	2.058415	-0.337033	-0.733833	-1.053122	1.178253	-0.033732	-1.132376
10	GSM1539555	CTL	2.001046	-1.171973	-0.326755	-0.527655	0.872383	-2.110761	1.676165	1.510296
11	GSM1539557	MCI	0.080408	-0.455062	-0.012725	-0.09205	1.173033	0.511202	-0.589235	1.983939
12	GSM1539558	MCI	0.629905	-0.278292	-0.034688	2.601276	2.373389	0.818056	0.536254	-0.278799
13	GSM1539559	MCI	0.116087	0.390532	2.754649	0.392867	0.804919	0.753937	0.949226	-0.787011
14	GSM1539560	AD	1.063322	0.762673	-0.696342	-0.763029	0.883528	-0.66858	-0.96863	-1.743136
15	GSM1539562	AD	-1.039711	0.405651	-0.741562	-1.498392	-0.689214	0.561863	-0.343839	-0.353071
16	GSM1539563	AD	-2.58142	1.223749	1.158262	0.032513	0.358236	2.033662	0.611212	-0.696894
17	GSM1539566	MCI	0.016991	-1.613704	-0.842708	1.366368	-0.04811	-0.656598	-0.745302	-0.432417
18	GSM1539567	MCI	0.549067	1.65635	2.005478	0.998476	1.729201	0.422911	-1.382272	-0.881006
19	GSM1539569	CTL	1.545118	1.068747	0.398316	-0.178101	-1.538981	0.253796	0.227994	0.160275
20	GSM1539570	AD	0.854958	-1.061058	1.031504	0.234918	-1.396481	-0.753536	1.931111	-0.060089

Figure (4.1). Sample of AD dataset.

4.4. Results of the proposed system

This section describes the results obtained from the proposed system for staging Alzheimer's patients. After the experiment, it turned out that the ratio of 70% for the training set and 30% for the testing set is the best method, and after pre-processing the data set and reducing the features according to the previously mentioned techniques and the classification model used as follows:

4.4.1. The First Classification Method Results

The results were calculated based on a variety of metrics mentioned in the second chapter of this thesis. In addition to the execution time of these methods. Table (4.2) shows the results and execution time and Fig. (4.2) explains the chart of these results.

Table (4.2). Results of 1st CNN classification model.

	Accuracy	Precision	Recall	F-score	Loss	Time in Seconds
CNN model "1"	95.60%	95.74%	94.04%	95.60%	4.035%	20.44 sec.

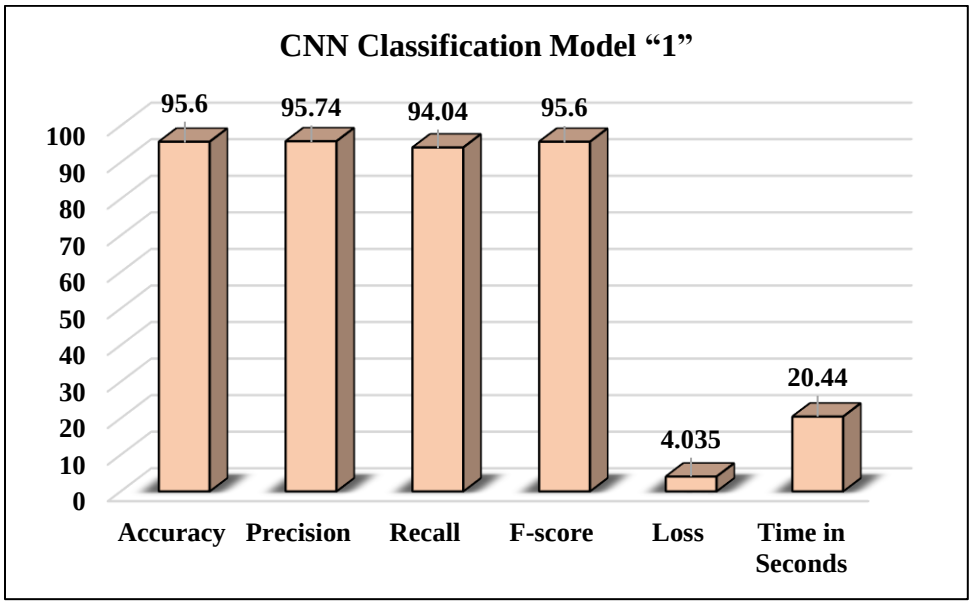

Figure (4.2). Chart of first CNN classification model results.

4.4.2. The Second Classification Method Results

This method is divided into more than one section where two techniques (PCA and SVD) are used to reduce the features in order to prepare data for the classification stage. These techniques are also used in two ways according to the values given to them, for example, PCA is used in two forms (PCA-10, and PCA-15). The SVD is also used in two different forms (SVD-10, and SVD-15). The CNN classification model is the same in the four cases mentioned. Features with its details can be known by simply copying the search key on the Google search engine, the details will appear as shown in Fig. (4.3).

Data table

ID	SPOT_ID	SEARCH_KEY	CHROMOSOME	DEFINITION
LOC23117	LOC23117	ILMN_44919	16	PREDICTED: Homo sapiens KIAA0220-like protein, transcript variant 11 (LOC23117), mRNA.
HS.575038	HS.575038	ILMN_127219		Homo sapiens cDNA: FLJ21027 fis, clone CAE07110
FCGR2B	FCGR2B	ILMN_139282		PREDICTED: Homo sapiens Fc fragment of IgG, low affinity IIb, receptor (CD32) (FCGR2B), mR
TRIM44	TRIM44	ILMN_5006	11	Homo sapiens tripartite motif-containing 44 (TRIM44), mRNA.
LOC653895	LOC653895	ILMN_38756		PREDICTED: Homo sapiens similar to protein geranylgeranyltransferase type I, beta subunit (LC
DGAT2L3	DGAT2L3	ILMN_7652	X	Homo sapiens diacylglycerol O-acyltransferase 2-like 3 (DGAT2L3), mRNA.
LOC387701	LOC387701	ILMN_35097	10	PREDICTED: Homo sapiens hypothetical LOC387701 (LOC387701), mRNA.
HS.133181	HS.133181	ILMN_77451		BX093329 Soares_parathyroid_tumor_NbHPA Homo sapiens cDNA clone IMAGp998A124183 ; I
C15ORF39	C15ORF39	ILMN_18382	15	Homo sapiens chromosome 15 open reading frame 39 (C15orf39), mRNA.
HS.545755	HS.545755	ILMN_108888		EST53225 Fetal heart II Homo sapiens cDNA 3 end, mRNA sequence
PCDHGA9	PCDHGA9	ILMN_22537	5	Homo sapiens protocadherin gamma subfamily A, 9 (PCDHGA9), transcript variant 1, mRNA.
STAMBPL1	STAMBPL1	ILMN_1387	10	Homo sapiens STAM binding protein-like 1 (STAMBPL1), mRNA.
HS.539137	HS.539137	ILMN_104330		xo15e10.x1 NCI_CGAP_Ut2 Homo sapiens cDNA clone IMAGE:2704074 3, mRNA sequence
HS.372465	HS.372465	ILMN_88273		AGENCOURT_10400114 NIH_MGC_82 Homo sapiens cDNA clone IMAGE:6616301 5, mRNA seq
HS.545796	HS.545796	ILMN_108918		he12d07.x1 NCI_CGAP_CML1 Homo sapiens cDNA clone IMAGE:2918797 3, mRNA sequence
STH	STH	ILMN_1785	17	Homo sapiens saitohin (STH), mRNA.
LOC342979	LOC342979	ILMN_36381	19	PREDICTED: Homo sapiens hypothetical LOC342979 (LOC342979), mRNA.
LOC728492	LOC728492	ILMN_34624	5	PREDICTED: Homo sapiens similar to small EDRK-rich factor 1A, telomeric, transcript variant 4
UGT2B7	UGT2B7	ILMN_138375		PREDICTED: Homo sapiens UDP glucuronosyltransferase 2 family, polypeptide B7 (UGT2B7), mI
CLEC12A	CLEC12A	ILMN_24114	12	Homo sapiens C-type lectin domain family 12, member A (CLEC12A), transcript variant 3, mRN

Figure (4.3). Description of genes.

The 15th effective features after the reduction process are shown in Table (4.3) as follows:

Table (4.3). The 15th effective features

ID_REF	Features
ILMN_1745607	7.4335366

ILMN_2106002	7.41463228
ILMN_2404917	7.63112345
ILMN_1699172	7.63499986
ILMN_2371825	7.64589661
ILMN_1688625	7.94254241
ILMN_1723521	7.35041595
ILMN_1735539	7.71563878
ILMN_2280189	7.59104841
ILMN_1686846	7.54852366
ILMN_1802537	7.36077528
ILMN_1691959	7.91093468
ILMN_1668617	7.44741716
ILMN_1668880	7.51332306
ILMN_2073157	7.47400231

The 10^{th} effective features after the reduction process are shown in Table (4.4) as follows:

Table (4.4). The 10^{th} effective features

ID_REF	**Features**
ILMN_1745607	7.4335366
ILMN_2106002	7.41463228
ILMN_2404917	7.63112345
ILMN_1723521	7.35041595
ILMN_2280189	7.59104841
ILMN_1686846	7.54852366
ILMN_1802537	7.36077528

ILMN_1668617	7.44741716
ILMN_1668880	7.51332306
ILMN_2073157	7.47400231

The 3rd effective features after the reduction process are shown in Table (4.5) as follows:

Table (4.5). The 3rd effective features

ID_REF	Features
ILMN_1745607	7.4335366
ILMN_1723521	7.35041595
ILMN_1802537	7.36077528

4.4.2.1. CNN classification model "2" with PCA-10

Table (4.6) shows the evaluation results of the second classification method with the PCA-10 feature reduction technique and the execution time of this method. Fig. (4.4) explains the chart of these results.

Table (4.6). Results of 2nd CNN classification model with PCA-10.

	Accuracy	Precision	Recall	F-score	Loss	Time in Seconds
CNN model "2" with PCA-10	98.9%	98.97%	98.8%	98.9%	1.164%	0.317 sec.

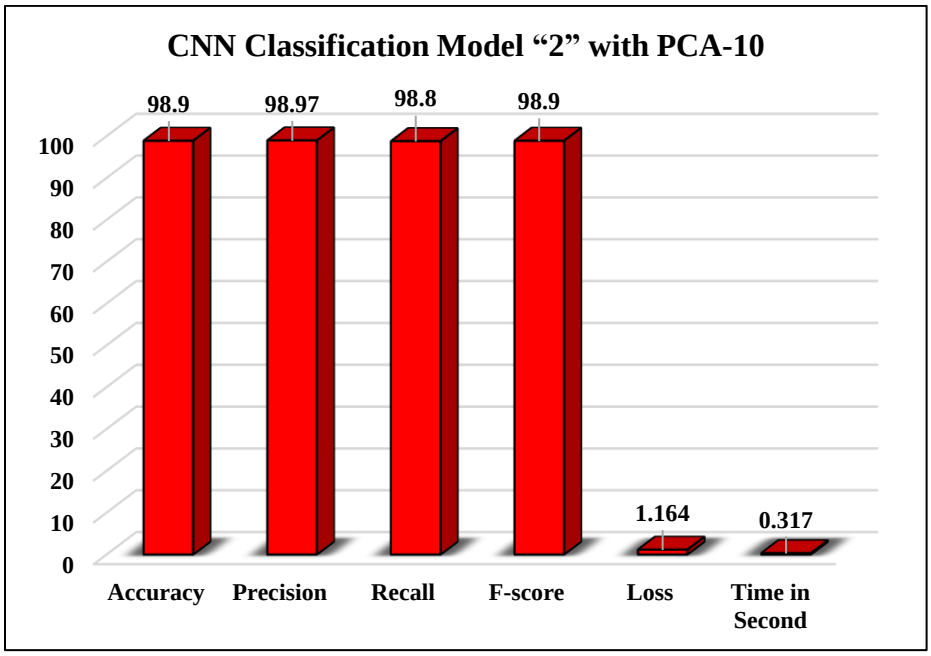

Figure (4.4). Chart of second CNN classification model with PCA-10 results.

4.4.2.2. CNN classification model "2" with PCA-15

Table (4.7) shows the evaluation results of the second classification method with the PCA-15 feature reduction technique and the execution time of this method. Fig. (4.5) explains the chart of these results.

Table (4.7). Results of 2nd CNN classification model with PCA-15.

	Accuracy	Precision	Recall	F-score	Loss	Time in Seconds
CNN model "2" with PCA-15	98.9%	98.99%	98.92%	98.9%	1.12%	0.349 sec.

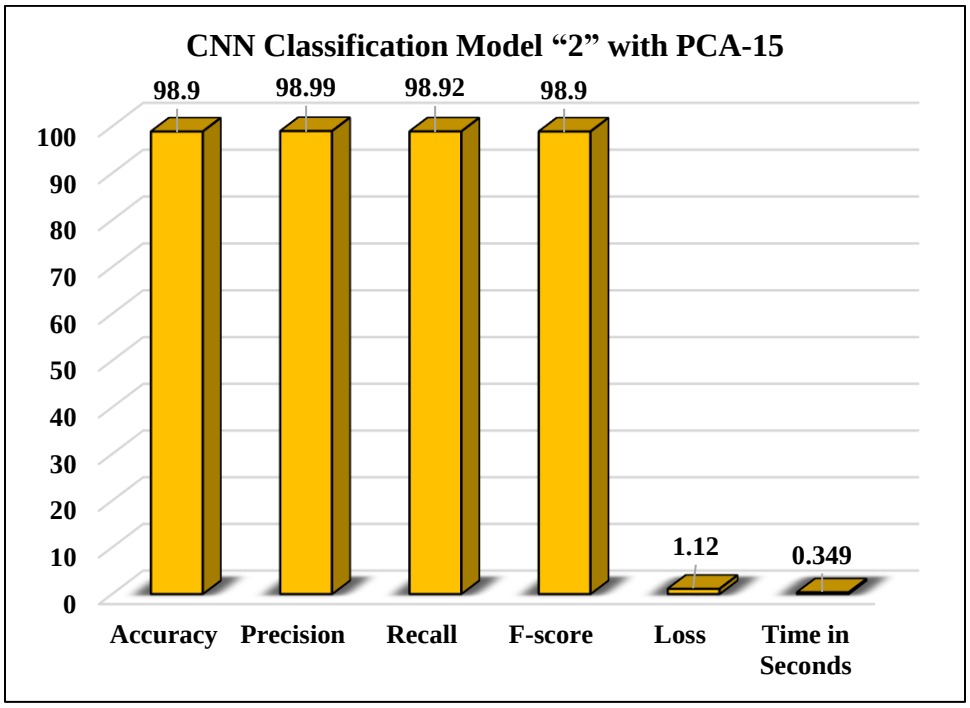

Figure (4.5). Chart of second CNN classification model with PCA-15 results.

4.4.2.3. CNN classification model "2" with SVD-10

Table (4.8) shows the evaluation results of the second classification method with the SVD-10 feature reduction technique and the execution time of this method. Fig. (4.6) explains the chart of these results.

Table (4.8). Results of 2nd CNN classification model with SVD-10.

	Accuracy	Precision	Recall	F-score	Loss	Time in Second
CNN model "2" with SVD-10	98.9%	98.93%	98.8%	98.9%	1.19%	0.368 sec.

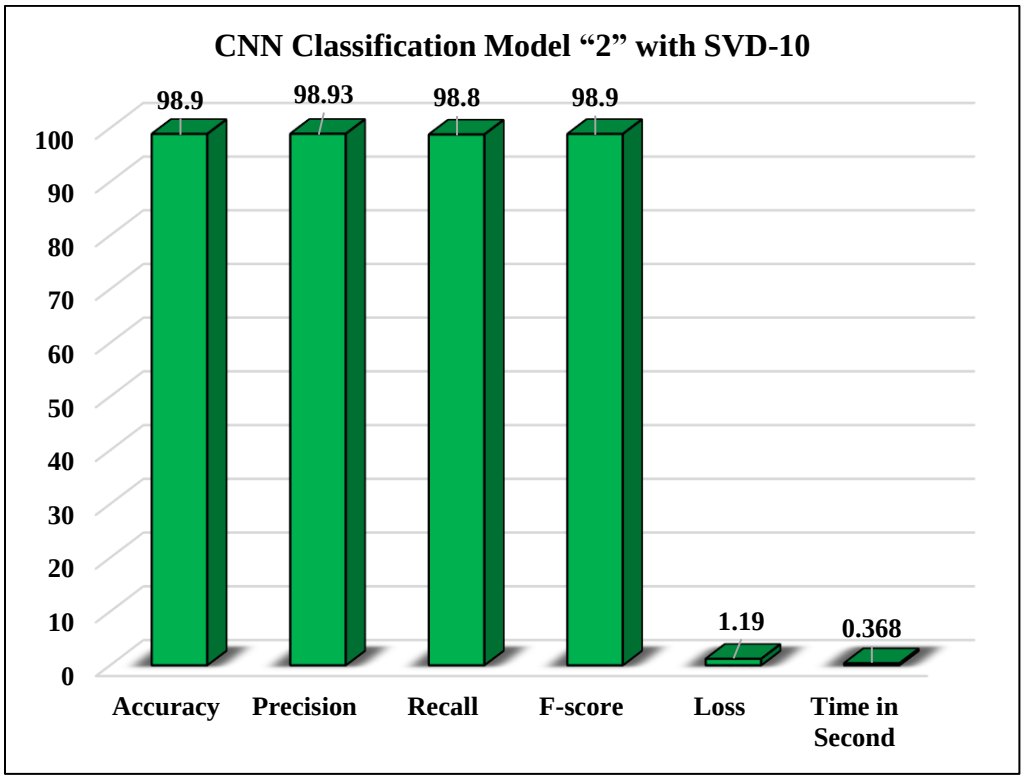

Figure (4.6). Chart of second CNN classification model with SVD-10 results.

4.4.2.4. CNN classification model "2" with SVD-15

Table (4.9) shows the evaluation results of the second classification method and the execution time of this method. Fig. (4.7) explains the chart of these results.

Table (4.9). Results of 2nd CNN classification model with SVD-15.

	Accuracy	Precision	Recall	F-score	Loss	Time in Second
CNN model "2" with SVD-15	98.9%	98.96%	98.87%	98.9%	1.1%	0.351 sec.

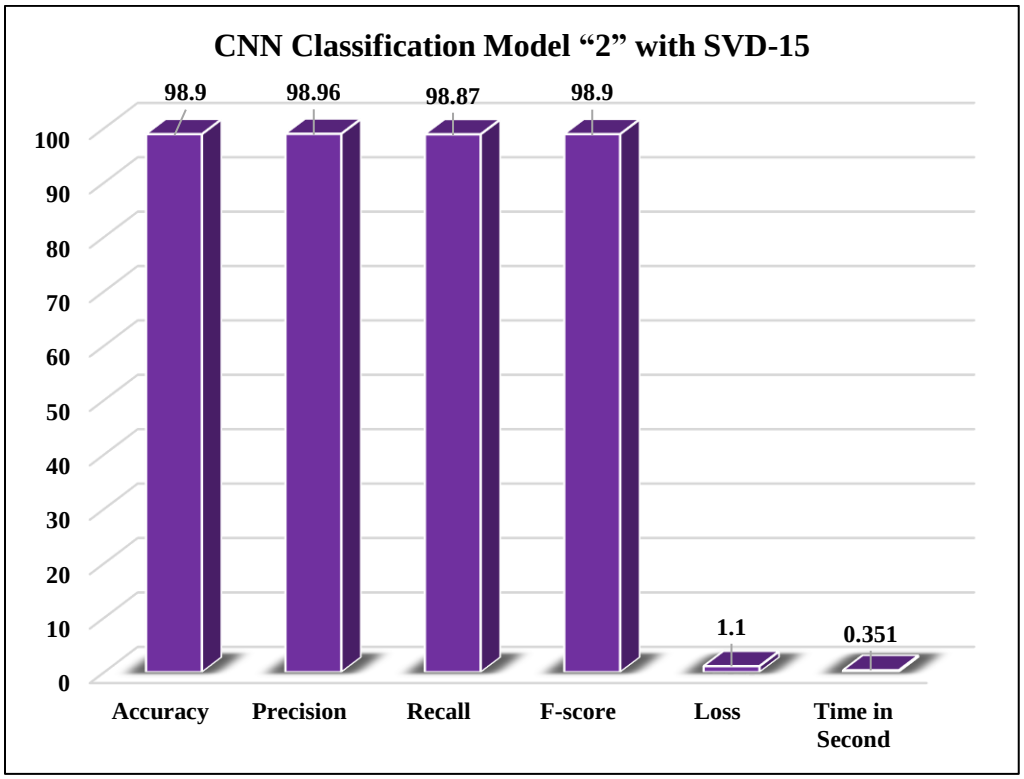

Figure (4.7). Chart of second CNN classification model with SVD-15 results.

4.4.3. The Third Classification Method Results

This method is divided into two sub-sections where two techniques (PCA and SVD) are used to reduce the features. Then, the proposed CNN features extraction model is used to extract features in order to prepare data for the classification stage. The CNN classification model is the same when using the PCA or SVD with the proposed CNN feature extraction.

4.4.3.1. CNN classification model "3" with PCA

Table (4.10) shows the evaluation results of the third classification method with the PCA-10 feature reduction technique and the execution time of this method. Fig. (4.8) explains the chart of these results.

Table (4.10). Results of 3rd CNN classification model with PCA.

	Accuracy	Precision	Recall	F-score	Loss	Time in Second
CNN model "3" with PCA	99.1%	99.6%	99.5%	99.1%	0.4	0.212

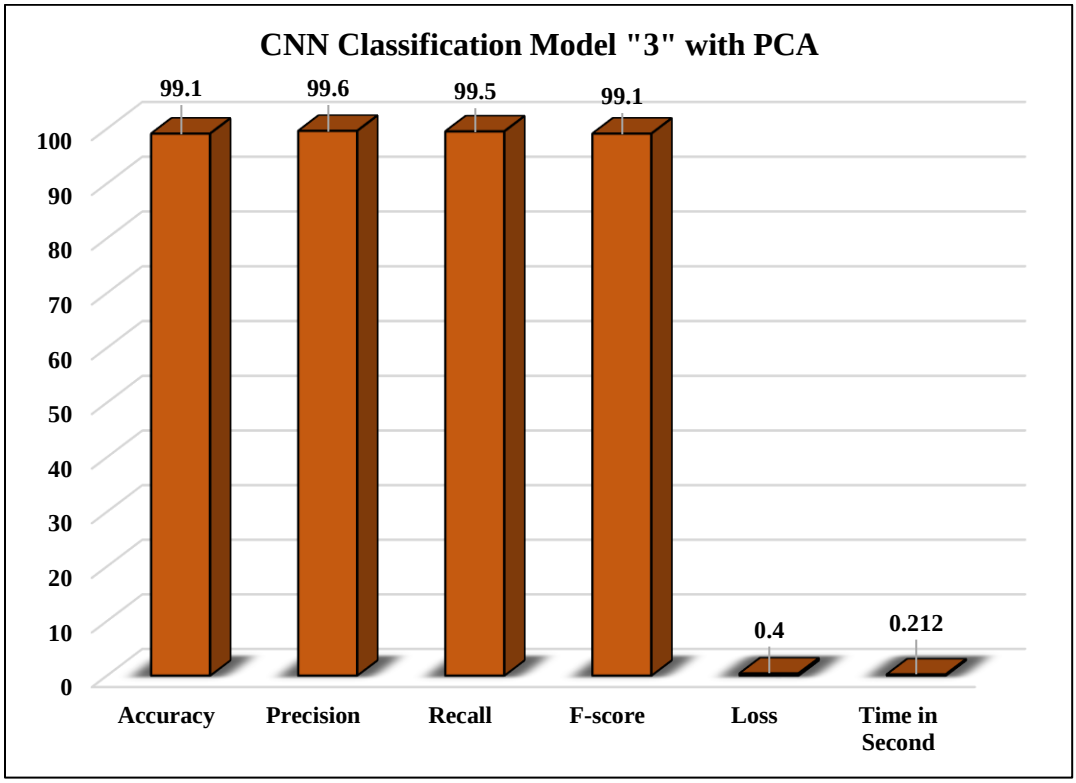

Figure (4.8). Chart of third CNN classification model with PCA results.

4.4.3.2. CNN classification model "3" with SVD

Table (4.11) shows the evaluation results of the third classification method with the SVD feature reduction technique and the execution time of this method. Fig. (4.9) explains the chart of these results.

Table (4.11). Results of 3rd CNN classification model with SVD.

	Accuracy	Precision	Recall	F-score	Loss	Time in Second
CNN model "3" with SVD	99.1%	99.4%	99.2%	99%	0.4%	0.219 sec.

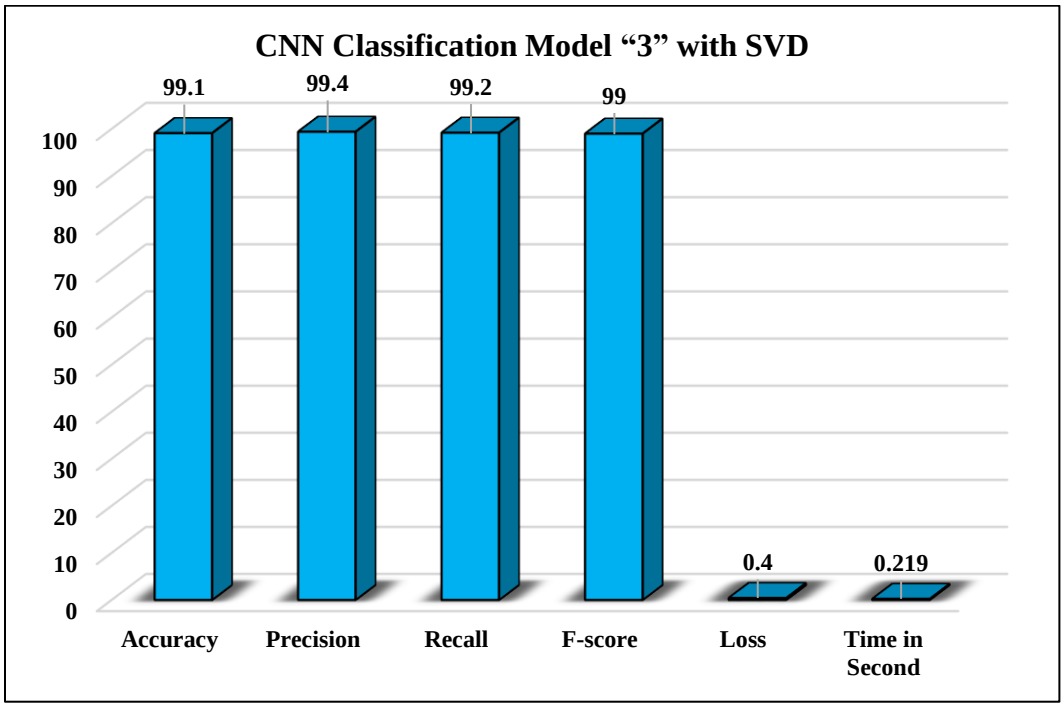

Figure (4.9). Chart of second CNN classification model with SVD results.

4.5. Discussion of Results

In this section, all the results of each of the three classification methods will be discussed, with all their ramifications, in an attempt to determine the best model and explain the reasons for this as follows:

4.5.1. Discussion of first method results

In the first method, when entering the data into the classification stage without any reduction or extraction of features, the results were perfect, and the accuracy of the

classification reached 95.60%, and the reason for this is the strength of the proposed CNN classification model "1" for classifying data in this method. As for the time, compared to other methods, it was the highest and reached 20.44 seconds.

4.5.2. Discussion of second method results

In this method, the data is prepared before it is entered into the filtering stage in order to reduce the time required for that. Two techniques were used as shown in the results section, and with each technique with values of 10 and 15 for PCA, PCA-10 results were better in terms of time with a slight difference with PCA-15. In both cases, the classification time did not exceed fractions of a second. As for the error rate, which is basically very small, it was less in the case of using PCA-15, with a little difference from PCA-10. In the case of employing SVD as a strategy to reduce features, the SVD-10 results were superior in terms of time, with just a slight difference from the SVD-15 results. The categorization time in both situations was fractions of a second. The error rate is basically very low; however, it was lower when SVD-15 was used significantly differently than SVD-10. In all cases, the accuracy was perfect, with a value of 98.9%, and with a time not exceeding fractions of a second. Therefore, this method can be considered in terms of time better than the previous one, and the reason for this is due to the use of known feature reduction techniques with the proposed CNN classification model "2" to classify the stage of Alzheimer's disease.

4.5.3. Discussion of third method results

In this method, the PCA or SVD technique is used to reduce the features, and then a special proposed CNN model is also used to perform the feature reduction task. Compared to the previous two methods, the classification time and error rate were less. Accordingly, the third method can be considered the best, and the reason for

this is that the use of feature reduction techniques with the proposed CNN technique to reduce features gave optimal results with both PCA and SVD in terms of accuracy of 99.1% with the lowest error rate equal to 0.004 and the least possible time of 0.212 seconds when using PCA and 0.219 seconds when using SVD.

Table (4.12) shows the results of all methods and Fig. (4.10) shows the chart of comparison of the proposed methods.

Table (4.12). Proposed models comparison

Method	Accuracy	Precision	Recall	F1-Score	Loss	Time in Seconds
Deep Learning (CNN)	95.60%	95.74%	94.04%	95.60%	4.035%	20.44 sec.
PCA-10 + CNN	98.9%	98.97%	98.8%	98.9%	1.164%	0.317 sec.
PCA-15 + CNN	98.9%	98.99%	98.92%	98.9%	1.12%	0.349 sec.
SVD-10+ CNN	98.9%	98.93%	98.8%	98.9%	1.19%	0.368 sec.
SVD-15+ CNN	98.9%	98.96%	98.87%	98.9%	1.1%	0.351 sec.
PCA+CNN+CNN	99.1%	99.1%	99.1%	99.1%	0.4%	0.212 sec.
SVD+CNN+CNN	99.1%	99.1%	99.1%	99.1%	0.4%	0.219 sec.

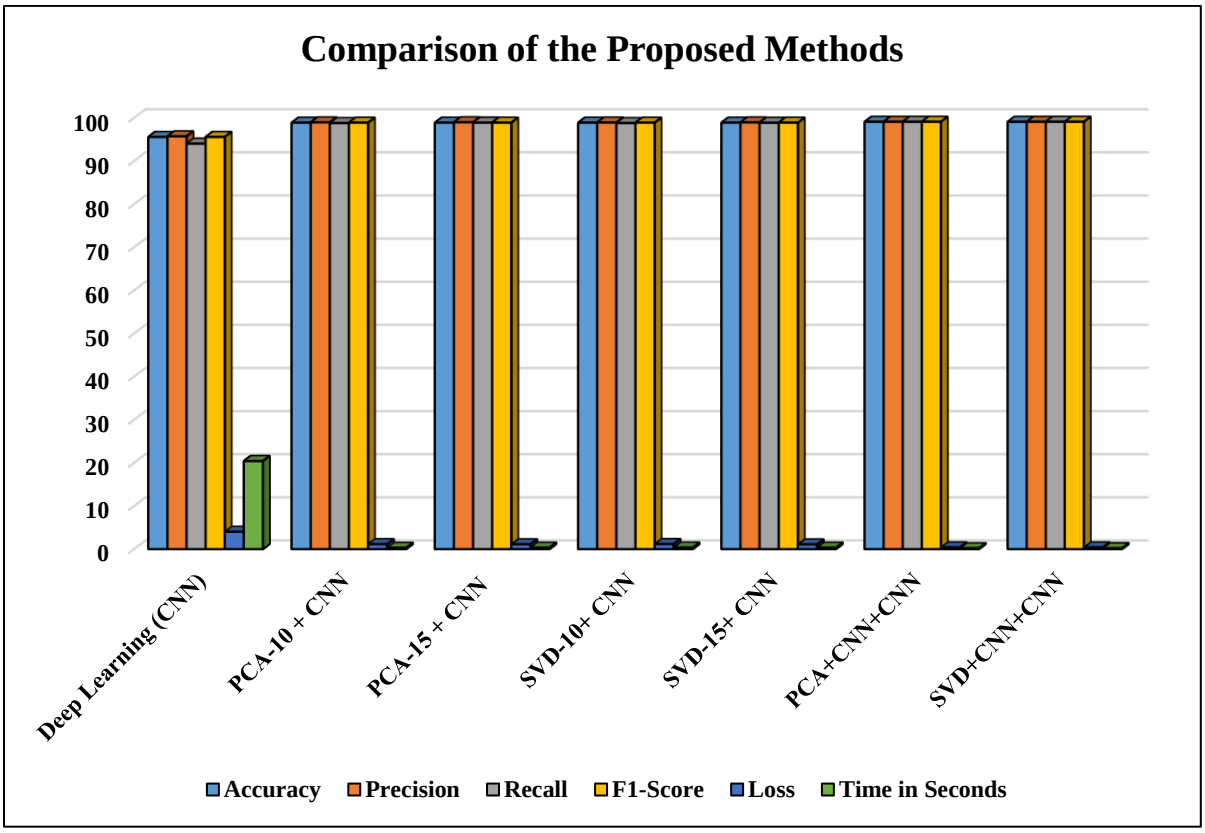

Figure (4.10). Chart of proposed models comparison

4.6. Comparison with Previous Studies

In this section, the results of the proposed system methods will be compared with the related studies mentioned in chapter one. Table (4.13) explains the accuracy values comparison of these studies and Fig. (4.10) shows the chart of these values.

Table (4.13). Comparison with related works

Ref. No.	Accuracy
[9]	95.73%
[10]	93%
[11]	86.60%
[12]	90.47%
[13]	90.91%

[14]		95%
[15]		94.53%
[16]		97.5%
[17]		97.4%
[18]		98.86%
[19]		77.80%
Our Proposed Methods	Deep Learning (CNN)	95.60%
	PCA10 + CNN	98.9%
	SVD10+ CNN	98.9%
	PCA15 + CNN	98.9%
	SVD15+ CNN	98.9%
	PCA+CNN +CNN	99.1%
	SVD+CNN+ CNN	99.1%

Through the results in Table (4.12), it can be noted that all the proposed methods are superior to the methods proposed in the studies that worked on the data of Alzheimer's patients, the details of which were clarified in chapter one. When comparing the results with reference no. [18], which worked on the same AD dataset, it is clear that the proposed method provided better results for classifying these data with 99.1% accuracy and in record time.

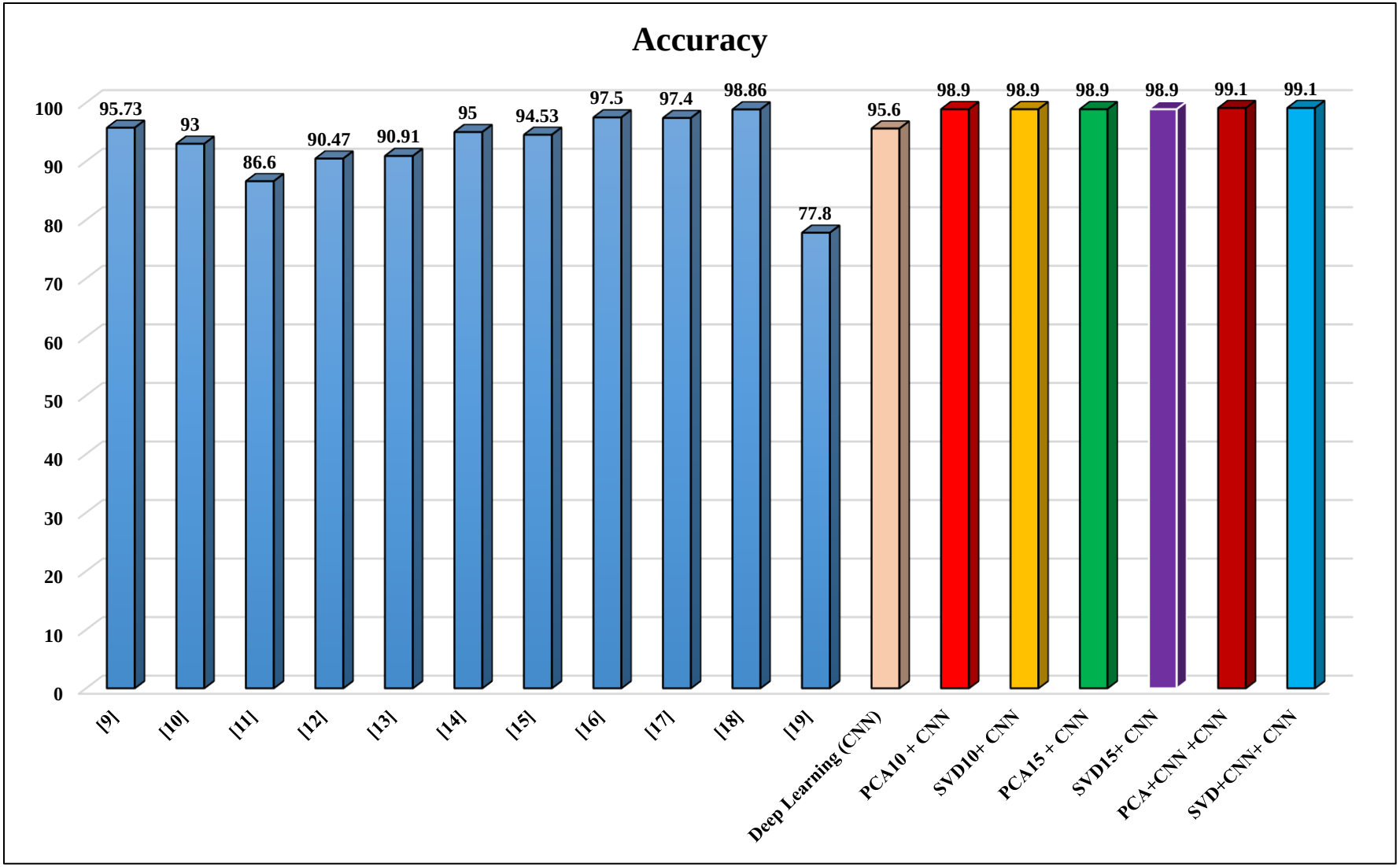

Figure (4.11). Chart of accuracy comparison with related studies.

Chapter five

Conclusions and Recommendations

Chapter Five
Conclusions and Recommendations

5.1. Conclusions

The following most significant findings are obtained during the creation and implementation of the suggested system and the achievement of its results:

1. The suggested system included a preprocessing stage represented by the normalization process which was carried out as a crucial stage since it helped to prepare the dataset for the proposed deep learning models. The Holdout cross-validation approach was adopted; this method provides adjustable propriety in data splitting in both the training and testing phases.
2. The use of feature reduction techniques PCA and SVD had a significant impact on increasing the accuracy of the proposed system in addition to reducing the execution time because it works to preserve the important features that can contribute to raising the accuracy of the proposed system and exclude features that are considered irrelevant.
3. Using the feature extraction method based on the proposed CNN model in the third method contributed to reducing the time more compared to using the feature reduction method only.

4. In many cases, the suggested deep learning classification models, which were developed using the CNN model, can outperform traditional machine learning techniques. After the important features have been found utilizing feature reduction and feature extraction, the classification model can detect patients with AD from MCI or CTL with a reduced error rate. As a consequence, the suggested

5. CNN-1D used in all proposed models is more dependable, achieving the best results, and it is unaffected by the size or kind of dataset. The attained accuracy was 99.1%, and the average execution time was 0.21 seconds.

5.2. Recommendations for Future Works

Several issues have been identified that must be addressed in the future to develop the suggested models, as follows:

1. Working on another dataset that could be medical images instead of numerical data.
2. Try classifying other diseases, such as chronic illnesses and the COVID-19 pandemic, using the suggested models.
3. Employ other techniques to reduce features such as Linear Discriminant Analysis (LDA) or Gray Level Co-Occurrence Matrix (GLCM) in the event that medical images are handled.

References

[1] Z. Breijyeh and R. Karaman, "Comprehensive Review on Alzheimer's Disease: Causes and Treatment," Molecules, vol. 25, no. 24, 2020.

[2] H. Hampel et al., "The Cholinergic System in the Pathophysiology and Treatment of Alzheimer's Disease," Brain A J. Neurol, vol. 141, pp. 1917–1933, 2018.

[3] G. Paroni, P. Bisceglia, and D. Seripa, "Understanding the amyloid hypothesis in Alzheimer's disease," Journal of Alzheimer's Dis., vol. 68, pp. 493–510, 2019.

[4] C. Saratxaga, I. Moya, A. Picón, M. Acosta, A. Moreno-Fernandez-de-Leceta, E. Garrote, and A. Bereciartua-Perez, "MRI Deep Learning-Based Solution for Alzheimer's Disease Prediction," J Pers Med. vol. 11, no. 9, 2021.

[5] M. Shahbaz, "Classification of Alzheimer's Disease using Machine Learning Techniques," In Proceedings of the 8th International Conference on Data Science, Technology and Applications (DATA 2019), pp. 296-303, 2019.

[6] S. Qiu et al., "Multimodal Deep Learning for Alzheimer's Disease Dementia Assessment," Nature Communications, vol. 13, no. 3404, 2022.

[7] Y. Wang, D. Tu, J. Du, X. Han, Y. Sun, Q. Xu, G. Zhai, and Y. Zhou., "Classification of Subcortical Vascular Cognitive Impairment Using Single MRI Sequence and Deep Learning Convolutional Neural Networks," Science Neurodegeneration, vol. 13, no. 627, 2019.

[8] I. Sarker, "Deep Learning: A Comprehensive Overview on Techniques, Taxonomy, Applications and Research Directions," SN COMPUT. SCI., vol. 2, no. 420, 2021.

[9] R. Jaina, N. Jaina, A. Aggarwala, and D. Hemanth, "Convolutional Neural Network based Alzheimer's Disease Classification from Magnetic Resonance Brain Images", Cognitive Systems Research, 2018.

[10] J. Islam, "Brain MRI Analysis for Alzheimer's Disease Diagnosis Using an Ensemble System of Deep Convolutional Neural Networks", Brain Informatics, vol. 5, no. 2, 2018.

[11] O. Kanghan, C. Young-Chul, K. Kim, K. Woo-Sung and O. Seok, "Classification and Visualization of Alzheimer's Disease using Volumetric Convolutional Neural Network and Transfer Learning," Scientific Reports, vol. 9, no.1, 2019.

[12] S. Basheera and M. Ram, "Convolution Neural Network–Based Alzheimer's Disease Classification Using Hybrid Enhanced Independent Component Analysis Based Segmented Gray Matter of T2 Weighted Magnetic Resonance Imaging with Clinical Valuation," Alzheimer's Dement. Transl. Res. Clin. Interv, vol.5, pp. 974–986, 2019.

[13] S. Bringas, S. Sergio, R. Duquec, C. Lage, and J. Monta˜na, "Alzheimer's Disease Stage Identification Using Deep Learning Models," Journal of Biomedical Informatics, July 13, 2020.

[14] Y. Fu'ada, I. Wijayanto, N. Pratiwi, F. Taliningsih, S. Rizal and M. Pramudito, "Automated Classification of Alzheimer's Disease Based on MRI Image Processing using Convolutional Neural Network (CNN) with AlexNet Architecture", Journal of Physics: Conference Series, 2021.

[15] M. Al-Adhaileh, "Diagnosis and Classification of Alzheimer's Disease by Using a Convolution Neural Network Algorithm", Soft Computing ©Springer, 2021.

[16] Y. AbdulAzeem, W. Bahgat, and M. Badawy "A CNN Based Framework for Classification of Alzheimer's Disease," Neural Computing and Applications, vol. 33, pp. 10415–10428, 2021.

[17] H. Mohammed, "Machine Learning Methods to Predict the Genes Expression that Affects Stages of Alzheimer's Disease". Master Thesis, University of Babylon, College of Information Technology, 2021.

[18] M. Odusami, R. Maskeliūnas, and R. Damaševičius, "An Intelligent System for Early Recognition of Alzheimer's Disease Using Neuroimaging", Sensors, vol. 22, no.740, 2022.

[19] Z. Baiwen, W. Jingxuan, L. Lan, and W. Shuicai, "The Effect of Smoothing Filter on CNN based AD Classification," In Proceedings of the 2019 8th International Conference on Bioinformatics and Biomedical Science", New York, NY, USA, 29–31 October 2019; pp. 54–58.

[20] S. Kumar and M. Nandhini, "A Comprehensive Survey: Early Detection of Alzheimer's Disease Using Different Techniques and Approaches," International Journal of Computer Engineering & Technology (IJCET), vol. 8, no. 4, July-August 2017, pp. 31–44.

[21] A. Porsteinsson, R. Isaacson, S. Knox, M. Sabbagh, and I. Rubino, "Diagnosis of Early Alzheimer's Disease: Clinical Practice in 2021," Journal Prev. Alzheimer Disease, vol. 3, no. 8, pp. 371-386, 2021.

[22] A. Atri, "The Alzheimer's Disease Clinical Spectrum. Medical Clinics of North America," Elsevier, vol. 103, no. 2, pp. 263–293, 2019.

[23] E. Jabason, M. Ahmad, and M. Swamy, "Classification of Alzheimer's Disease from MRI Data Using an Ensemble of Hybrid Deep Convolutional Neural Networks," In Proceedings of the 2019 IEEE 62nd International Midwest Symposium on Circuits and Systems (MWSCAS), Dallas, TX, pp. 481–484, USA, 4–7 August 2019.

[24] F. Huang, 2019. “Data Processing”. In: Schintler L., McNeely C. (eds) Encyclopedia of Big Data. Springer, Cham.

[25] A. Hassler, E. Menasalvas, F. José García-García, L. Rodríguez-Mañas, and A. Holzinger, “Importance of medical data preprocessing in predictive modeling and risk factor discovery for the frailty syndrome,” BMC Medical Informatics and Decision Making, vol. 19, no. 33, 2019.

[26] G. Kamani, R. Parmar and Y. Ghodasara, “Data Normalization in Data Mining Using Graphical User Interface: A Pre-Processing Stage”, Guj. J. Ext. Education, vol. 30, no. 2, December 2022.

[27] A. Nurhopipah and U. Hasanah, “Dataset Splitting Techniques Comparison for Face Classification on CCTV Images,” Indonesian Journal of Computing and Cybernetics Systems (IJCCS), vol.14, no.4, pp. 341-352, October 2020.

[28] Q. Nguyen et al., “Influence of Data Splitting on Performance of Machine Learning Models in Prediction of Shear Strength of Soil,” Mathematical Problems in Engineering, 2021.

[29] J. Awwalu and O. Nonyelum, “On Holdout and Cross Validation: A Comparison between Neural Network and Support Vector Machine,” International Journal of Trend in Research and Development, vol. 6, no. 2, pp. 235-239.

[30] D. Berrar, “Cross-Validation,” Encyclopedia of Bioinformatics and Computational Biology, Elsevier, 2018.

[31] J. Villarreal et al., “Secure Learning for Android Malware Detection,” Faculty of Engineering Systems Engineering Program ITCS Department ICESI University Project, 2019.

[32] O. Aouedi, K. Piamrat , and B. Parrein, “Intelligent Traffic Management in Next-Generation Networks,” Future Internet, MDPI, 2022.

[33] P. Jindal and D. Kumar, "A Review on Dimensionality Reduction Techniques," International Journal of Computer Applications, vol. 173, no.2, pp. 42-46, September 2017.

[34] S. Khalid, "A Survey of Feature Selection and Feature Extraction Techniques in Machine Learning," Science and Information Conference (SAI) - London, UK, pp. 372–378, 2014.

[35] M. Kondo, C. Bezemer, Y. Kamei, A. Hassan, and O. Mizuno, "The Impact of Feature Reduction Techniques on Defect Prediction Models," Empirical Software Engineering, vol. 24, no. 10, 2019.

[36] A. Oprea, "The Use of Principal Component Analysis (PCA) in Building Yield Curve Scenarios and Identifying Relative-Value Trading Opportunities on the Romanian Government Bond Market," Journal of Risk and Financial Management, vol. 15, no. 247, 2022.

[37] G. Chao, L. Yuan and D. Weiping, "Recent Advances in Supervised Dimension Reduction: A Survey," Machine Learning Knowledge Extraction, vol. 1, pp. 341–358, 2019.

[38] A. Tharwat, "Principal Component Analysis - A Tutorial", International Journal of Applied Pattern Recognition, vol. 3, no. 3, pp. 197- 240, 2016.

[39] S. Brunton and J. Kutz, "Singular Value Decomposition (SVD)," Data-Driven Science and Engineering, pp.3-46, 2019.

[40] S. Gaoa and D. Limaa, "A Review of the Application of Deep Learning in the Detection of Alzheimer's Disease," International Journal of Cognitive Computing in Engineering, vol. 3, pp. 1–8, 2022.

[41] J. Choi and B. Lee, "Combining of Multiple Deep Networks Via Ensemble Generalization Loss, Based on MRI Images, for Alzheimer's Disease Classification," IEEE Signal Process Lett, vol. 27, pp. 206–210, 2020.

[42] A. Viloriaa, N. Alberto, L. Zelayab, and N. Varelac, "Unsupervised Learning Algorithms Applied to Grouping Problems," 7th International Symposium on Emerging Inter-networks, Communication and Mobility (EICM), August 9-12, 2020, Leuven, Belgium, pp. 677–682.

[43] P. Sen, M. Hajra, and M. Ghosh, "Supervised Classification Algorithms in Machine Learning: A Survey and Review," Emerging technology in modeling and graphics, Springer, pp.99-111, 2020.

[44] R. Yamashita, M. Nishio, R. Gian, and K. Togashi, "Convolutional Neural Networks: An Overview and Application in Radiology," Insights into Imaging, vol. 9, pp. 611–629, 2018.

[45] N. Senan, M. Aamir, R. Ibrahim, N. Taujuddin, W. Muda, "An Efficient Convolutional Neural Network for Paddy Leaf Disease and Pest Classification," International Journal of Advanced Computer Science and Applications, vol. 11, 2020.

[46] C. Nwankpa, W. Ijomah, A. Gachagan, and S. Marshall, "Activation Functions: Comparison of Trends in Practice and Research for Deep Learning," arXiv:1811.03378v1 [cs.LG] 8 Nov 2018.

[47] Y. Bai, "RELU-Function and Derived Function Review," SHS Web of Conferences, vol. 144, no. 25, 2022.

[48] F. Sultana, A. Sufian, and P. Dutta, "Advancements In Image Classification Using Convolutional Neural Network," in Proceedings of the 2018 4th International Conference on Research in Computational Intelligence and Communication Networks (ICRCICN), pp. 122–129, IEEE, Kolkata, India, November 2018.

[49] M. Yani, B. Irawan, and C. Setiningsih, "Application of Transfer Learning Using Convolutional Neural Network Method for Early Detection of Terry's

Nail," International Conference on Electronics Representation and Algorithm (ICERA 2019).

[50] M. Valizadeh and S. Wolff, "Convolutional Neural Network Applications in Additive Manufacturing: A review," vol. 4, 2022.

[51] A. Abadea, P. Afonso, F. Flaviode, B. Vidalc, "Plant Diseases Recognition on Images Using Convolutional Neural Networks: A Systematic Review," Computers and Electronics in Agriculture, vol. 185, pp. 1–31, 2021.

[52] S. Kiranyaza, O. Avcib, O. Abdeljaber, T. Moncef, G. Daniel, and J.Inman, "1D Convolutional Neural Networks and Applications: A Survey," Mechanical Systems and Signal Processing, vol. 151, 2021.

[53] A. Tegegne, "Applications of Convolutional Neural Network for Classification of Land Cover and Groundwater Potentiality Zones," Hindawi, Journal of Engineering, 2022.

[54] H. Xu et al., "Application of Remote Sensing Fuzzy Assessment Method in Groundwater Potential in Wailingding Island," Journal of Supercomputing, vol. 76, no. 5, pp. 3646–3658, 2020.

[55] I. Markoulidakis, G. Kopsiaftis, I. Rallis, and I. Georgoulas, "Multiclass Confusion Matrix Reduction Method and Its Application on Net Promoter Score Classification Problem," Technologies, vol. 9, no. 81, 2021.

الخلاصة

مرض الزهايمر هو مرض عصبي يصيب كبار السن في المقام الأول ويؤدي بسهولة إلى تدهور الذاكرة والسلوك البشري. وبحسب التقارير ، هناك 26.6 مليون حالة إصابة بمرض الزهايمر على مستوى العالم ، بما في ذلك 14.99 مليون حالة في المراحل الأولى وكان ذلك في عام 2006. ومن المتوقع أن يلحق الضرر بواحد من كل 85 شخصًا في جميع أنحاء العالم. وفي يومنا هذا وتحديداً في العام 2022، هناك 47.5 مليون مريض بمرض الزهايمر، 58٪ منهم يقيمون في الدول منخفضة ومتوسطة الدخل. على مدى العقود القليلة الماضية ، تم استخدام مناهج التعلم الآلي لتصنيف مرض الزهايمر ، مع النتائج التي تستند إلى الميزات التي تم إنشاؤها يدويًا والمصنفات ذات التصميم متعدد الخطوات. فأنه تم استخدام شبكات الالتفاف العصبية مؤخرًا لتصنيف مرض الزهايمر بسبب تقدم التعلم العميق.

يتضمن النظام المقترح خمس مراحل: المعالجة المسبقة ، وتقسيم البيانات ، وتقليل الميزات ، والتصنيف ، ومراحل التقييم. تم إنجاز مرحلة المعالجة المسبقة باستخدام طريقة التطبيع. تم استخدام التحقق من صحة Hold-out لتقسيم مجموعة بيانات مرض الزهايمر إلى مجموعة التدريب (70٪) والتي تساوي (398 حالة) ومجموعة الاختبار (30٪) التي تساوي (171 حالة). تم تقليل الميزة باستخدام طريقتين لتقليل الميزات: تحليل المكون الرئيسي وتحليل القيمة الفردية بينما يتم استخراج الميزة من خلال استخدام نموذج استخراج الميزة القائم على الشبكة العصبية التلافيفية المقترح. بعد ذلك ، سيتم إدخال البيانات التي أجريت عليها جميع العمليات السابقة في مرحلة التصنيف ، وفي هذه المرحلة ، تم اقتراح ثلاثة نماذج من الشبكة العصبية التلافيفية أحادية البعد من أجل زيادة دقة النظام المقترح وتقليل وقت التنفيذ والخطأ معدل. المرحلة النهائية هي التقييم حيث تم استخدام بعض مقاييس التقييم لتقييم نماذج النظام المقترحة.

أظهرت النتائج التجريبية معدلات دقة تساوي 95.60٪ في الطريقة الأولى المقترحة ، و 98.9٪ في الطريقة الثانية المقترحة ، و 99.1٪ في الطريقة الثالثة المقترحة لتصنيف بيانات مرضى الزهايمر وبلغت نسبة الخطأ في طرق التصنيف الأول والثاني والثالث 4.035٪ ، (1.164٪ ، 1.12٪ ، 1.19٪ ، 1.1٪) ، و (0.4٪ ، 0.4٪) على التوالي. لا يتجاوز وقت تنفيذ جميع الطرق المقترحة بضع ثوانٍ. أقل وقت تم الحصول عليه في الطريقة الثالثة يساوي 0.212 ثانية مع تحليل المكون الرئيسي و 0.219 ثانية مع تحليل القيمة الفردية، حيث أدى استخدام تقنيات تقليل الميزات المذكورة ونموذج تقليل الميزات القائم على الشبكة العصبية التلافيفية المقترح إلى تقليل وقت التصنيف. تم إنشاء النظام المقترح باستخدام لغة برمجة Python 3.6 ، على جهاز كمبيوتر بالخصائص التالية (Intel Core i7 ، وحدة المعالجة المركزية 2.20 جيجاهرتز ، ذاكرة الوصول العشوائي 16 جيجابايت ، نظام التشغيل: 64 بت).

جمهورية العراق
وزارة التعليم العالي والبحث العلمي
جـامعـــة بـابـــل
كلية تكنولوجيا المعلومات - قسم البرمجيات

استخدام الشبكات العصبية التلافيفية لتصنيف مراحل مرض الزهايمر

رسالة مقدمة

إلى مجلس كلية تكنولوجيا المعلومات - جامعة بابل كجزء من متطلبات

نيل درجة الماجستير في تكنولوجيا المعلومات / البرمجيات

من قبل :

حسين محمد خليل الزبيدي

بأشراف:

م. د. مهند محمد الياسري

1444 هـ **2022 م**