
**Arab American University**

**Faculty of Graduate Studies**

# Diagnosis and Classification of Hypothyroidism and Hyperthyroidism Based on Machine Learning Techniques

By

**Samer Bassam Mohammad Sweileh**

Supervisor

**Prof. Dr. Mohammed Awad**

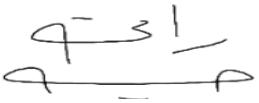
**This Thesis Was Submitted in Partial Fulfillment of the Requirements for the Master's Degree in Computer Science**

**February/ 2022**

# Thesis Approval

## Diagnosis and classification of Hypothyroidism and Hyperthyroidism based on Machine Learning Techniques

This Thesis was Defended Successfully on **10/02/2022** and Approved By:

| Committee Members | Signature |
| --- | --- |
| 1. Supervisor: **Prof. Dr. Mohammed Awad** |  |
| 2. Internal Examiner: **Dr. Rami Hadrob** |  |
| 3. External Examiner: **Dr.Yousef –Awwad Daraghmi** |  |

# Declaration

I declare that this Master dissertation has been composed by me and is based on my own work, unless stated otherwise. I confirm that this Master's thesis is my own work and I have documented all sources and material used; no other person's work has been used without due acknowledgement.

All references and verbatim extracts have been quoted, and all sources of information, including graphs and data sets, have been specifically acknowledged. To my best knowledge, this Master dissertation has not been accepted in any other previous application for a degree, in whole or in part.

Name: Samer Sweileh

Signature: 

# Dedication

I dedicate this thesis first to the steadfast Palestinian people on their land and secondly to my family, friends, and colleagues who have given me a lot of support. Special thanks to my wife for her support and encouragement for me all the time. I also dedicate this thesis to my supervisor, Prof. Dr. Mohammed Awad, thanking him for all the support and assistance.

# Acknowledgment

I would like to seize this opportunity to express my deep regards to Prof. Mohammed Awad for his advice, support, and time which he has spent reviewing my work. Prof. Mohammed provided valuable suggestions which have had a significant impact and have helped in overcoming many obstacles in writing this thesis in the best way.

# Abstract

**Diagnosis and classification of Hypothyroidism and Hyperthyroidism based on Machine Learning Techniques**

By: Samer Sweileh

Supervisor: Prof. Dr. Mohammed Awad

The thyroid gland is one of the most important parts of the body, as any imbalance in it leads to many health problems. Thyroid examinations are one of the necessary periodic examinations that must be carried out to check on health, as any trouble in it causes health complications if it is not treated. Therefore, there is an urgent need for early detection of these diseases as well as accuracy in diagnosis, so Artificial Intelligence (AI) approaches can support medical decisions to detect thyroid disease before the disease situation worsens and affects the patient with other problems. This research uses the Machine Learning (ML) Hybrid Model of Neural Networks (NNs) and Genetic algorithms (GAs) [GAs-MLPNNs], and Deep Learning (DL) Model to classify and predict thyroid diseases depending on features and medical images.

The datasets used in this research depend on different resources, we used a global dataset, also local datasets were collected from four cities in Palestine, and a global dataset for ultrasound images of the thyroid. The datasets pass-through data preprocessing and feature extraction process. The final step is the application of different AI methods to recognize the patterns in these datasets. Decision-tree (DT), Naïve Bayes (NB), Support Vector Machine (SVM), k-Nearest Neighbor (KNN), Ensemble Methods, Multi-Layer Perceptron Neural Networks (MLPNNs), GAs-

MLPNNs, and Deep learning Convolution Neural Network (CNN) implemented by VGG-16 Model were used to recognize the pattern on the datasets and improve the results of the classification of thyroid diseases with high accuracy.

In the first stage, DT, NB, SVM, KNN, Ensemble, and MLPNNs were applied to the global dataset to classify thyroid diseases, and the accuracy results were as follows: 99.5%, 93.3%, 98.2%, 95.1%, 99.6%, and 95.6% respectively. In the second stage, DT, NB, SVM, KNN, Ensemble, and MLPNNs have been applied to the local dataset in two ways, the whole data collected from the four cities, and the data for each city. All cities datasets obtained the highest accuracy when applying the Ensemble model which produces 91.10%. The best model in terms of accuracy of results for each city is DT, which obtained the accuracy; 92.40%, 88.50%, 91.90%, and 89.90% in Ramallah, Nablus, Qalqiliah, and Salfeit respectively.

In the third stage, a custom global dataset was created to compare the results of classification to the local datasets. In the fourth stage, Hybrid Model (GAs-MLPNNs) was used to improve the accuracy of the classification of thyroid diseases, the model was applied on a custom global dataset and local dataset, the results were as follows: 95% for the custom global dataset, and 96% for the local dataset. In the fifth and final stage, a VGG-16 model based on CNN was used to classify a group of ultrasound images of the thyroid gland into a malignant tumor or a benign gland tumor, and the accuracy result of the classification was 87.00%.

# Table of Contents

# List of Figures

# List of Tables

# List of Abbreviation

| | |
|---|---|
| **AI** | Artificial Inelegance |
| **ANNs** | Artificial Neural Networks |
| **ML** | Machine Learning |
| **DL** | Deep Learning |
| **DT** | Decision Tree |
| **NB** | Naïve Bayes |
| **SVM** | Support Vector Machine |
| **KNN** | K-Nearest Neighbor |
| **MLPNNs** | Multi-Layer Perceptron Neural Networks |
| **GAs** | Genetic Algorithms |
| **VGG-16** | *Visual Geometry Group-16* |
| **CNN** | Convolutional Neural Network |
| **TSH** | Thyroid-stimulating hormone |
| **FT4** | Free Thyroxine hormone |
| **FT3** | Free Triiodothyronine hormone |
| **TT4** | Total Thyroxine hormone |
| **T4U** | Thyroxine utilization |
| **FTI** | Free Thyroxine Index |
| **mIU/L** | milli-international units per Liter |
| **ng/dL** | nanograms per deciliter |
| **pg/mL** | picograms per milliliter |
| **UNRWA** | United Nations Relief and Works Agency |
| **DDTI** | Digital Database Thyroid Image |
| **FS** | Feature Selection |
| **MI** | Mutual information |
| **AUC** | Area Under Curve |
| **ROC** | Receiver Operating Characteristic Curve |
| **TP** | True Positive |
| **TN** | True Negative |
| **FP** | False-Positive |
| **FN** | False-Negative |
| **TPR** | True Positive Rate |
| **TNR** | True Negative Rate |
| **PC** | Personal Computer |
| **PSO** | Particle Swarm Optimization |

# Chapter 1

# Introduction

## 1.1 Introduction

Thyroid diseases are one of the spreads among all segments of society, one out of every twenty people in the United States of America suffers from a disease of the thyroid gland, and this percentage is seven times more in women [12]. There are many risks involved in not being diagnosed with thyroid disease or being misdiagnosed. It is possible that a person with thyroid disease does not suffer from certain symptoms or that the symptoms are simple and do not feel them, so there is a need for periodic examination for anyone who suffers or does not suffer from symptoms [7] [12]. In Europe, the prevalence of hypothyroidism, a disease of the thyroid gland prevalent around the world, in a population sample of 228613 people was 4.70%. 4.11% of them suffer from hypothyroidism and were not previously diagnosed [16].

Signs and symptoms of hypothyroidism or hyperthyroidism (they are one of the most common diseases that affect the thyroid gland) [17] depending on the extent of the problem. These symptoms develop slowly and the symptoms of thyroid disease are often confused with those of other diseases as the clinical symptoms are similar, such as aging [84] or menopause, and others.

Thyroid disorders are diagnosed based on symptoms and a blood test that measures thyroid-stimulating hormone (TSH) levels, and thyroid hormones (TT3, TT4), where any difference in the levels of these hormones from the normal level in general,

there may be a problem in the thyroid gland that needs accurate diagnosis and treatment [18]. The thyroid gland is one of the most important parts of the body, as any imbalance in it leads to many health problems. Examination of the thyroid gland is one of the most important tests that must be performed to reassure health, as any defect in it causes health complications if it is not treated [19]. The thyroid gland produces thyroid hormones along with the iodine taken from food to keep the body's metabolism functioning at a normal rate. A thyroid disorder occurs either because of an overproduction of the hormone, which causes the body to use energy faster than it should (hyperthyroidism), or not having enough hormone causes the body to use energy more slowly than it should (hypothyroidism). Among the complications that may be caused by delay in the diagnosis and treatment of thyroid disease, for example, but not limited to heart disease: Thyroid problems can lead to heart disease, such as irregular heartbeat, whether slow or rapid heartbeat, which causes weakness in the heart muscle [20][79]. It can also increase the level of harmful cholesterol, and fluid accumulation around the heart, in addition to the development of hardening of the arteries. The patient may have to undergo a heart operation if it is proven that there is a problem resulting from thyroid disorders. Delaying treatment can also cause goiter, infertility, and decreased fertility, affecting mental health, neuropathy, and many other diseases and problems [7]. Therefore, there is an urgent need for early detection of these diseases as well as accuracy in diagnosis, so Machine Learning ML can support medical decision-makers [80] [81] from doctors and researchers to detect thyroid disease before the disease situation worsens and causes other problems.

This study attempts to use machine learning technology to classify and predict thyroid diseases in the early stages to take proactive measures. The hope is that the use

of ML techniques will assist and contribute to the early detection and diagnosis of this disease. In the beyond years, IT was used to increase and develop many studies, additionally, Artificial Intelligence (AI) has presented a strong inference tool to help scientific decision-making, and different components in technological know-how and engineering [21]. Simply, AI is a manner to educate computer systems on the way to think, make selections (decision making), and act like humans, so, logically the machine ought to do matters that human beings, in the beginning, do (might be taken into consideration intelligent) [21]. AI employs the statistics accumulated from digital health records, to give you a result a good way to help the medical doctors in decision making and diagnosing diseases [22]. Classification techniques have been used in many fields to collect similar data together, which makes it much easier to treat each group similarly and used for many other purposes depending on the problem [23].

In this thesis, different AI techniques are used for the classification of thyroid datasets local and international. The first model is the Decision Tree (DT), this technique classifies data into levels by building a tree consisting of three layers: root node layer, internal nodes layer, and leaf nodes layer [58]. The second model is the Naïve Bayes (NB) it is a probability-based ML technique, used to classify classes in datasets, this technique assumes that the properties used to build the model are separate from each other, so changing the values of one property does not affect another property [59]. The third technique is Support Vector Machine (SVM), this model aims to make boundaries between classes to classify [82]. The fourth technique is K-nearest neighbor (KNN), this technique is very simple, gives a good result, and usually assumes that the similar data is close to each other, when classifying new data, it depends on the distance to each class [83]. The fifth technique the Ensemble is an aggregation of several ML

models to improve the accuracy of the results, rather than using a single model that leads to less accurate results [64]. The sixth model is an artificial neural network (ANNs), ANNs, and simulates the biological neural networks. ANNs use to solve a variety of problems in; pattern recognition, prediction, optimization, associative memory, and control [70]. In this research Multi-layer perceptron, neural networks (MLPNNs) are used [71]. The seventh technique is the Hybrid Model of Neural Networks (NNs) and Genetic algorithms (GAs) [GAs-MLPNNs]; this hybrid model is used to improve the accuracy of the classification of thyroid diseases datasets. The eighth model is (VGG-16) model, which depends on Convolutional Neural Network (CNN) architecture, it used for thyroid ultra-sound images classification [76].

In this research, the experiments were performed using six ML techniques were applied on the global dataset and on local datasets to classify thyroid diseases, DT, NB, SVM, KNN, Ensemble, and MLPNNs. Also, the local dataset is divided into 4 subsets depending on the city.

A custom global dataset was used to compare the results of classification between the local and global datasets.

To improve the classification of thyroid diseases, a Hybrid Model (GAs-MLPNNs) was applied on custom global dataset and local dataset (all cities dataset). Finally, a Deep Learning (VGG-16) model based on CNN was used to classify a group of ultrasound images of the thyroid gland into a malignant tumor or a benign gland tumor.

## 1.2 Objectives

The main objective is to improve the accuracy to determine whether the person has hyperthyroidism or hypothyroidism or does not suffer from either by using AI techniques that apply to two groups of datasets (Global & Local). In other words, examining the appropriate mechanism to analyze the data, select feature, and find a more precise method to assess thyroid disease is the primary goal of the study. Early detection and accurate diagnosis of any thyroid disorder using AI technique contribute to the treatment plan as well as avoiding many serious diseases as mentioned in the introduction.

**Specific Objectives of the Study**

Other specific objectives of the study are the following:

- Choosing the most effective variable that affects the dataset classification by using Feature Selection techniques.
- Applying a group of AI techniques for thyroid Diseases classification on two groups of datasets (Global & Local).
- Choosing the most appropriate AI techniques for the classification of thyroid Diseases on the global and local dataset.
- Performing an accuracy comparison between Local and Global datasets by applying AI techniques.
- Applying the proposed hybrid model of GAs-MLPNNs to the classification of thyroid Diseases.

- Applying Deep Learning (DL) model by using the VGG-16 model is based on Convolutional Neural Networks (CNN) technique to classify the types of thyroid cancer through thyroid roid nodules through an ultrasound images dataset.

## 1.3 Contribution

Many studies in the field of information technology have proposed many algorithms and used machine learning techniques in classifying and predicting thyroid diseases. But at the local level, no studies have been conducted in this field. So, the work is still rudimentary to automate the detection of thyroid diseases concerning hyperactivity or hypothyroidism. Therefore, the starting point was from here and to contribute to the collection of local data about laboratory thyroid examinations, and then use machine learning techniques to classify thyroid diseases at the local level.

## 1.4 Overview

In this section of the introduction chapter, we will review a summary of the rest of the upcoming chapters that we will discuss in this thesis, in order to achieve the goals and objectives of the work. The rest of the thesis organized as follows: Chapter two will give a background about the three datasets we were used (Global dataset, Local dataset, and ultrasound images dataset), then literature within the related research field will be discussed, also, it provides some other used techniques to predict and classify thyroid diseases. Chapter three will discuss in general the methodologies were used in this study in general, at first the data collection, and then preprocessing steps will be shown, which are: feature selection and dealing with missing values. Secondly, the building model phase and DT, NB, SVM, KNN, Ensemble, and MLPNNs algorithms were explained,

also GAs model to be combined with MLPNNs the general method procedure will be illustrated, and VGG-16 model based on CNN model will be explained. Finally, different performance measurements were introduced to evaluate the machine learning models. In chapter four, some experiments and the results of each dataset with different models will be discussed, also, a comparison between models showing the best one will be provided for each dataset, also a comparison between results for global and local datasets. Chapter five presents the conclusion and future work.

# Chapter 2

# Background

## 2.1 Background

Technological advances allow the different sciences to improve their processes, and incorporate new functionalities to optimize the times in diagnoses and treatment. The patients generate an enormous amount of information [25], providing the possibility of creating tools that allows us to collaborate in the diagnosis and study of different pathologies. Thyroid diseases are a problem for society because they generate a decrease in the quality of life of people, excessive tiredness, irritability, weight gain for no reason among others [26] [27].

The thyroid is a gland belonging to the endocrine, its main function is to release into the blood thyroid hormones T3 and T4, which are involved in the development of the nervous system, cognitive processes, regulate metabolism, and burning calories that influence every cell in the body organ, that is necessary to control the functions of almost all organs of the body [27].

Many studies in the field of information technology have proposed many algorithms and use machine learning techniques in classifying and predicting thyroid diseases [1][2][3][4][5][6].

In this thesis, several models were used in classification such as Decision Tree (DT), Naïve Bayes (NB), K-Nearest Neighbors (KNN), Support Vector Machine (SVM), Ensemble tree, Multi-layer perceptron Neural Network (MLP), and hybrid

model (GAs-MLPNNs). These models were applied to the global dataset and the local datasets, and then comparisons were made between the results that appeared. Also, a (VGG-16) model based on the convolutional neural network (CNN) which is one of the deep learning models, had been applied to classify images of the thyroid gland (ultra-sound images) and divide them according to the type of inflammation, benign or malignant.

## 2.2 Dataset Description

In this research, three datasets were used:

### 2.2.1 Global Dataset

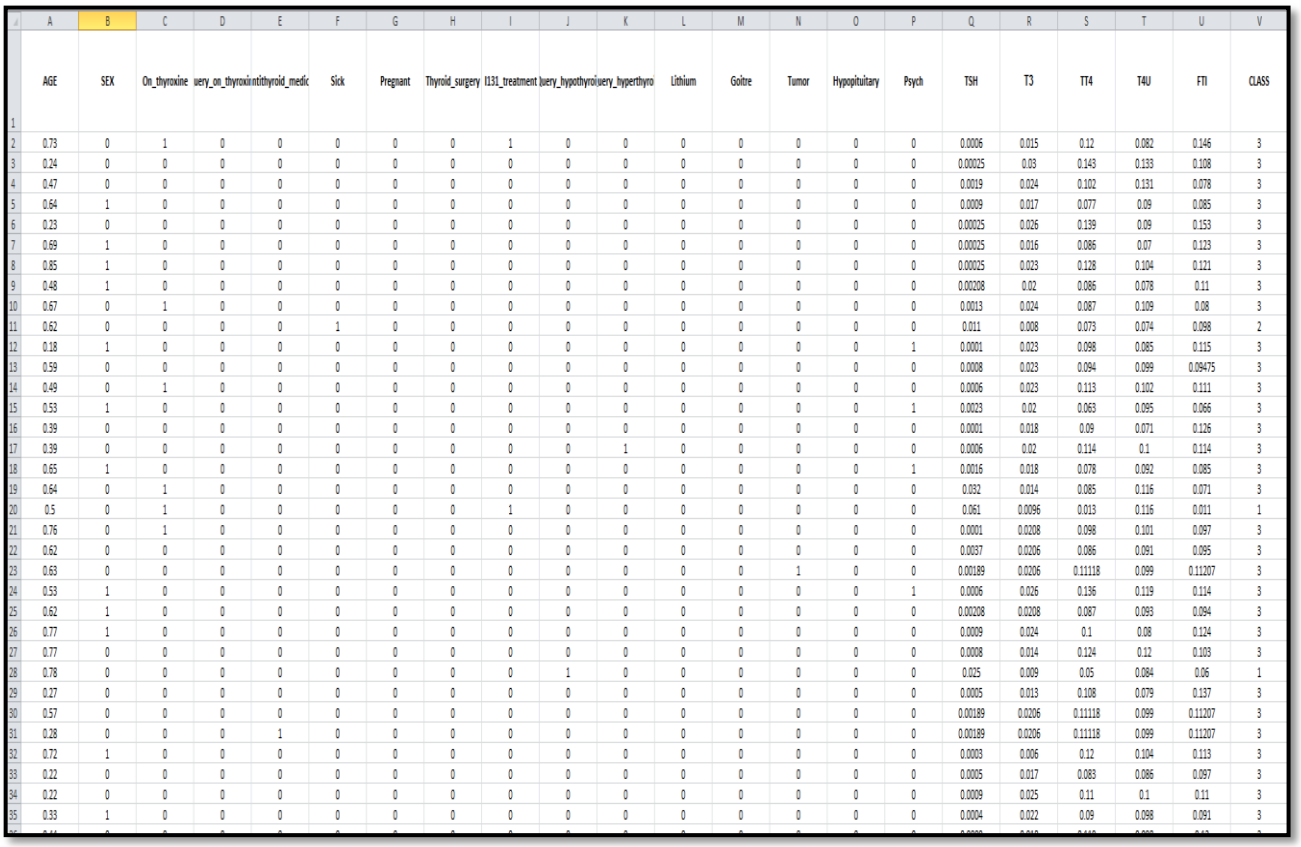
These are global data sets that exist in University of California, Irvine (UCI) Machine Learning Repository [28] about thyroid lab tests: It is a data of patients that have been examined in the laboratory and have been divided into three classes: I. Normal. II.Hyper. III. Hypo.

This dataset contains 21 variables (feature) and contains 7200 samples (a record). In the table (2.1), there are details of these variables.

**Table 2.1: Valid ranges & description for each feature (Global Dataset)**

| # | Feature Name | Description | Range of Value | Type of value |
|---|---|---|---|---|
| 1. | Age | Age of the patient | 0.01–0.97 | Continuous variable |
| 2. | Sex | Sex of the patient | 0 (Male)<br>1 (Female) | Binary variable |
| 3. | On_thyroxine | | 0 (FALSE)<br>1 (TRUE) | Binary variable |

| | | | | |
|---|---|---|---|---|
| 4. | Query_on_thyroxine | | 0 (FALSE)<br>1 (TRUE) | Binary variable |
| 5. | On_antithyroid_medication | type of treatment | 0 (FALSE)<br>1 (TRUE) | Binary variable |
| 6. | Sick | Does he/she suffer from illness? | 0 (FALSE)<br>1 (TRUE) | Binary variable |
| 7. | Pregnant | positive pregnancy result | 0 (FALSE)<br>1 (TRUE) | Binary variable |
| 8. | Thyroid_surgery | Did you do thyroid surgery? | 0 (FALSE)<br>1 (TRUE) | Binary variable |
| 9. | I131_treatment | type of treatment | 0 (FALSE)<br>1 (TRUE) | Binary variable |
| 10. | Query_hypothyroid | Request for laboratory examination to inquire about hypothyroidism | 0 (FALSE)<br>1 (TRUE) | Binary variable |
| 11. | Query_hyperthyroid | Request for laboratory examination to inquire about hyperthyroidism | 0 (FALSE)<br>1 (TRUE) | Binary variable |
| 12. | Lithium | type of treatment | 0 (FALSE)<br>1 (TRUE) | Binary variable |
| 13. | Goitre | Thyroid disease | 0 (FALSE)<br>1 (TRUE) | Binary variable |
| 14. | Tumor | Thyroid disease | 0 (FALSE)<br>1 (TRUE) | Binary variable |
| 15. | Hypopituitary | pituitary gland disease | 0 (FALSE)<br>1 (TRUE) | Binary variable |
| 16. | Psych | Does he/she suffer from mental illness? | 0 (FALSE)<br>1 (TRUE) | Binary variable |
| 17. | TSH | Thyroid-stimulating hormone | (0.0–0.53) | Continuous variable |
| 18. | T3 | Triiodothyronine hormone | (0.0005–0.18) | Continuous variable |
| 19. | TT4 | Total Thyroxine hormone | (0.002–0.6) | Continuous variable |
| 20. | T4U | Thyroxine utilization | (0.017–0.233) | Continuous variable |
| 21. | FTI | Free Thyroxine Index TT4(ug/ml) x (T3 uptake patient/T3 uptake control) | (0.002–0.642) | Continuous variable |



| | A | B | C | D | E | F | G | H | I | J | K | L | M | N | O | P | Q | R | S | T | U | V |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | AGE | SEX | On_thyroxine | uery_on_thyroxin | ntithyroid_medic | Sick | Pregnant | Thyroid_surgery | I131_treatment | uery_hypothyroi | uery_hyperthyro | Lithium | Goitre | Tumor | Hypopituitary | Psych | TSH | T3 | TT4 | T4U | FTI | CLASS |
| 2 | 0.73 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0.0006 | 0.015 | 0.12 | 0.082 | 0.146 | 3 |
| 3 | 0.24 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0.00025 | 0.03 | 0.143 | 0.133 | 0.108 | 3 |
| 4 | 0.47 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0.0019 | 0.024 | 0.102 | 0.131 | 0.078 | 3 |
| 5 | 0.64 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0.0009 | 0.017 | 0.077 | 0.09 | 0.085 | 3 |
| 6 | 0.23 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0.00025 | 0.026 | 0.139 | 0.09 | 0.153 | 3 |
| 7 | 0.69 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0.00025 | 0.016 | 0.086 | 0.07 | 0.123 | 3 |
| 8 | 0.85 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0.00025 | 0.023 | 0.128 | 0.104 | 0.121 | 3 |
| 9 | 0.48 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0.00208 | 0.02 | 0.086 | 0.078 | 0.11 | 3 |
| 10 | 0.67 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0.0013 | 0.024 | 0.087 | 0.109 | 0.08 | 3 |
| 11 | 0.62 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0.011 | 0.008 | 0.073 | 0.074 | 0.098 | 2 |
| 12 | 0.18 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0.0001 | 0.023 | 0.098 | 0.085 | 0.115 | 3 |
| 13 | 0.59 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0.0008 | 0.023 | 0.094 | 0.099 | 0.09475 | 3 |
| 14 | 0.49 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0.0006 | 0.023 | 0.113 | 0.102 | 0.111 | 3 |
| 15 | 0.53 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0.0023 | 0.02 | 0.063 | 0.095 | 0.066 | 3 |
| 16 | 0.39 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0.0001 | 0.018 | 0.09 | 0.071 | 0.126 | 3 |
| 17 | 0.39 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0.0006 | 0.02 | 0.114 | 0.1 | 0.114 | 3 |
| 18 | 0.65 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0.0016 | 0.018 | 0.078 | 0.092 | 0.085 | 3 |
| 19 | 0.64 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0.032 | 0.014 | 0.085 | 0.116 | 0.071 | 3 |
| 20 | 0.5 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0.061 | 0.0096 | 0.013 | 0.116 | 0.011 | 1 |
| 21 | 0.76 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0.0001 | 0.0208 | 0.098 | 0.101 | 0.097 | 3 |
| 22 | 0.62 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0.0037 | 0.0206 | 0.086 | 0.091 | 0.095 | 3 |
| 23 | 0.63 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0.00189 | 0.0206 | 0.11118 | 0.099 | 0.11207 | 3 |
| 24 | 0.53 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0.0006 | 0.026 | 0.136 | 0.119 | 0.114 | 3 |
| 25 | 0.62 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0.00208 | 0.0208 | 0.087 | 0.093 | 0.094 | 3 |
| 26 | 0.77 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0.0009 | 0.024 | 0.1 | 0.08 | 0.124 | 3 |
| 27 | 0.77 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0.0008 | 0.014 | 0.124 | 0.12 | 0.103 | 3 |
| 28 | 0.78 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0.025 | 0.009 | 0.05 | 0.084 | 0.06 | 1 |
| 29 | 0.27 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0.0005 | 0.013 | 0.108 | 0.079 | 0.137 | 3 |
| 30 | 0.57 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0.00189 | 0.0206 | 0.11118 | 0.099 | 0.11207 | 3 |
| 31 | 0.28 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0.00189 | 0.0206 | 0.11118 | 0.099 | 0.11207 | 3 |
| 32 | 0.72 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0.0003 | 0.006 | 0.12 | 0.104 | 0.113 | 3 |
| 33 | 0.22 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0.0005 | 0.017 | 0.083 | 0.086 | 0.097 | 3 |
| 34 | 0.22 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0.0009 | 0.025 | 0.11 | 0.1 | 0.11 | 3 |
| 35 | 0.33 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0.0004 | 0.022 | 0.09 | 0.098 | 0.091 | 3 |

**Figure 2.1: Sample of Global dataset**

### 2.2.2 Local Dataset

It is a dataset of patients who performed laboratory tests related to the thyroid gland from 4 local Palestinian cities. The patients were classified into three categories:

I. Hyper. II. Hypo. III. Normal.

This dataset contains 8 variables (feature) and contains 3775 samples (a record). In table (2.2), there are details of these variables.

**Table 2.2: Valid ranges & description for each feature (Local Dataset)**

| # | Feature Name | Description | Range of Value | Type of value |
|---|---|---|---|---|
| 1. | Age | Age of the patient | 1–82 | Continuous variable |
| 2. | Sex | Sex of the patient | 0 (Male)<br>1 (Female) | Binary variable |
| 3. | TSH | Thyroid-stimulating hormone | (0.01-52.63) | Continuous variable |
| 4. | FT4 | Free Thyroxine hormone | (0.15-4.90) | Continuous variable |
| 5. | FT3 | Free Triiodothyronine hormone | (1.10-14.28) | Continuous variable |
| 6. | Pregnant | positive pregnancy result | 0 (FALSE)<br>1 (TRUE) | Binary variable |
| 7. | Treatment | Is the person being treated with specific thyroid medication? | 0 (FALSE)<br>1 (TRUE) | Binary variable |
| 8. | City | The sample is from which city | Nablus, Qalqliah, Salfit, Ram-Allah | Numeric variable |

| | A | B | C | D | E | F | G | H | I |
|---|---|---|---|---|---|---|---|---|---|
| 1 | Age | Sex | TSH | FTI(FT4) | freeT3 | Pregnant | tretment | City | class |
| 2 | 20 | 1 | 3.84 | 0.90 | 2.79 | 0 | 0 | 1 | 3 |
| 3 | 13 | 0 | 0.88 | 1.80 | 2.13 | 0 | 0 | 1 | 3 |
| 4 | 66 | 1 | 3.87 | 1.13 | 3.43 | 0 | 0 | 2 | 3 |
| 5 | 75 | 0 | 2.80 | 1.10 | 2.78 | 0 | 0 | 1 | 3 |
| 6 | 56 | 1 | 0.77 | 0.86 | 2.93 | 0 | 0 | 1 | 3 |
| 7 | 29 | 0 | 1.46 | 1.10 | 3.41 | 0 | 0 | 2 | 3 |
| 8 | 72 | 1 | 2.48 | 0.85 | 2.25 | 0 | 0 | 2 | 3 |
| 9 | 47 | 1 | 4.04 | 1.31 | 3.68 | 0 | 0 | 2 | 3 |
| 10 | 18 | 1 | 4.07 | 1.56 | 4.48 | 0 | 0 | 1 | 3 |
| 11 | 21 | 0 | 1.00 | 1.43 | 2.73 | 0 | 0 | 2 | 3 |
| 12 | 4 | 1 | 4.09 | 0.89 | 4.78 | 0 | 0 | 3 | 3 |
| 13 | 31 | 0 | 4.04 | 0.88 | 2.08 | 0 | 0 | 4 | 3 |
| 14 | 54 | 0 | 2.24 | 1.58 | 3.30 | 0 | 0 | 4 | 3 |
| 15 | 61 | 0 | 3.44 | 1.71 | 1.81 | 0 | 0 | 2 | 3 |
| 16 | 7 | 0 | 0.94 | 1.33 | 2.64 | 0 | 0 | 2 | 3 |
| 17 | 14 | 0 | 2.00 | 0.91 | 1.80 | 0 | 0 | 4 | 3 |
| 18 | 71 | 1 | 3.88 | 1.63 | 2.52 | 0 | 0 | 4 | 3 |
| 19 | 74 | 0 | 3.41 | 1.14 | 2.34 | 0 | 0 | 4 | 3 |
| 20 | 74 | 0 | 4.05 | 1.09 | 2.82 | 0 | 0 | 3 | 3 |

**Figure 2.2: Sample of Local dataset**

The details about each feature for the local Dataset:

- Age: Ages of patients who were examined in the laboratory based on the decision of a doctor or therapist who suspected a specific problem in the thyroid gland after clinical diagnosis. The patients in the sample ranged from 1 year to 84 years old according to the collected local data, we note that those who suffer from or complain of thyroid problems are widespread among all age groups, especially the youth group. Figure (2.3) shows how the data is distributed depending on Age.

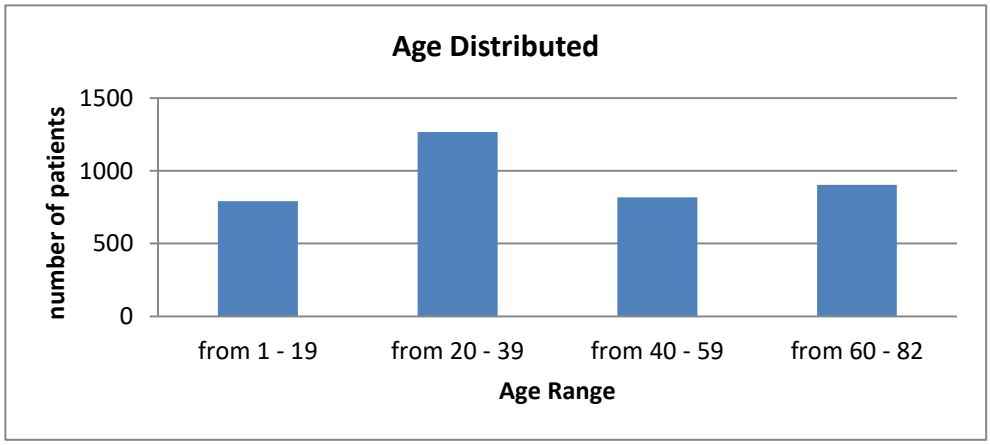



**Figure 2.3: Age distributed on the local Dataset**

- Sex: The gender of the patient who did the laboratory analysis of thyroid tests. Figure (2.4) shows the gender distribution over the entire sample (local dataset).

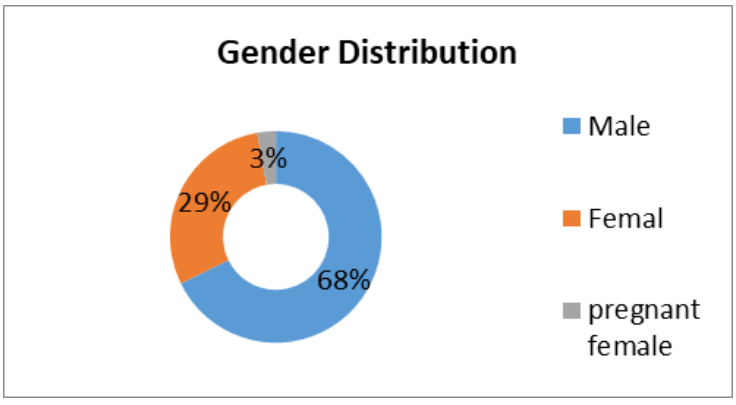



**Figure :2.4 Gender distributed on the local Dataset**

- TSH: Thyroid-stimulating Hormone, this hormone is secreted by the pituitary gland in the brain, which in turn stimulates the thyroid gland to produce its hormones into the bloodstream [29].

The TSH test indicates the levels of these hormones in the blood and detects the performance of the thyroid gland in the event of an overactive thyroid gland or hypothyroidism, this examination can also reveal the presence of any of these hormones, and detect an abnormality in the gland until Before symptoms appear on the patient, and get the necessary treatment before the disease progresses to the worst. [29][30].

TSH normal values are (0.5 to 5.0) milli-international units per liter (mIU/L); any rise in these normal percentages, the person may have hypothyroidism or be in the beginning stage of the disease. Conversely, any decrease in these percentages may be a person with hyperthyroidism or be at the beginning stage of the disease. The increase or decrease from normal levels can be caused by taking certain medications, being pregnant, or the patient is in the recovery stage from a certain surgery. [31][32].

- FT4: Free T4 or free thyroxin, thyroxine is one of the hormones produced by the thyroid gland, and there are two types of this hormone in the human bloodstream: the first is thyroxine bound to proteins and is called Total T4 (TT4) and the other is called free T4 or free thyroxin (FT4) [33], and this is measured in a thyroid blood test to detect problems that affect in the thyroid gland. It is more accurate and is what was used in this study [31]. This hormone and with the help of another hormone called triiodothyronine (T3) is responsible for the metabolic processes in the body, and any imbalance in the levels of their secretion in the blood leads to many problems.

The normal ratio of FT4 hormone is 0.8 to 1.8 nanograms per deciliter (ng/dL) [34], any imbalance in these ratios of increase or decrease can be the cause of a specific disease in the thyroid gland, but first, you must check the ratios of the rest of the hormones for accurate diagnosis.

- FT3: Free T3 or free triiodothyronine, is the other hormone that the thyroid gland secretes into the bloodstream along with thyroxine to regulate energy processes in the body, as we mentioned above. The normal range: 2.3 to 4.1 picograms per milliliter (pg/mL) [34]
- Pregnant: This variable indicates whether the woman person who did a thyroid test in the laboratory is pregnant or not. The patient is taken into consideration if she is pregnant or not; because the level of thyroid hormones in a pregnant woman is abnormal due to pregnancy and this does not mean that she has a thyroid disease, or that she has a thyroid disease temporarily [35].
- Treatment: This variable indicates that the person who did the thyroid lab tests is taking a specific drug as a treatment for a disease he suffers from, especially a disease originally related to the thyroid gland. This variable was taken into account because the period of treatment with thyroid drugs does not give real results, but rather the results are based on treatment [36].
- City: This variable refers to the city in which the test sample was registered. In our study, samples were taken from private laboratories, laboratories affiliated with the United Nations Relief and Works Agency (UNRWA), laboratories of medical relief committees, and others.

The cities from which samples were taken are in order according to the variable number:

1. Nablus. 2. Qalqiliah. 3. Salfeit. 4. Ramallah.

Note: This variable was excluded from its entry into ML models and based on it; local datasets were divided into four sub-datasets.

Figure (2.5) shows the distribution of patients or samples according to the four cities.

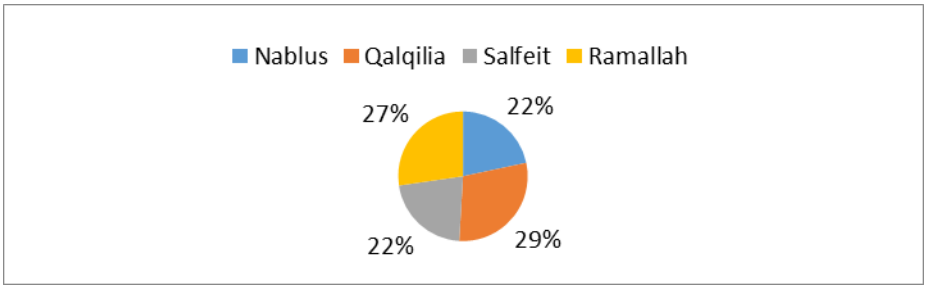



**Figure 2.5: Percentage of records (patients) in cities**

### 2.2.3 Ultra Sound Image Dataset

In this part of the study, a set of ultrasound images of the thyroid were used for patients with inflammation within the thyroid, based on the doctor's initial diagnosis. Through these images, the doctor can assess whether the inflammation is benign or malignant through a set of factors and characteristics in the image [37]. The thyroid image set was obtained from a global dataset Digital Database Thyroid Image (DDTI) [38][39]; it is a website that contains an open-access data set for scientific research purposes. The 151 medical images of the thyroid gland were used after processing between images of a thyroid gland with a malignant tumor (92 images) and a benign tumor (59 images) to train a (VGG-16) model. Figure (2.6) shows the distribution of images in the dataset

between images of malignant tumors and images of benign tumors, and this was implemented using Python code.

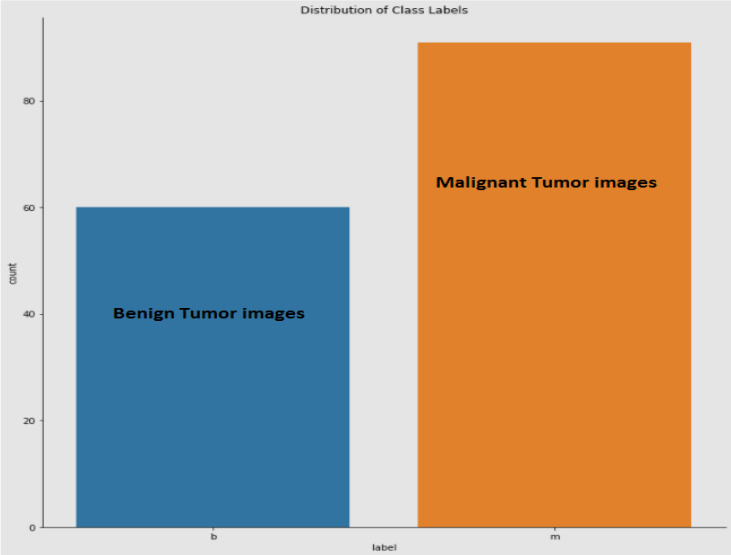



**Figure 2.6: Distribution of Tumor Types**

Figure (2.7) shows a sample from the training set of ultrasound medical images diagnosed with benign thyroiditis. This sample was taken from the training set images.

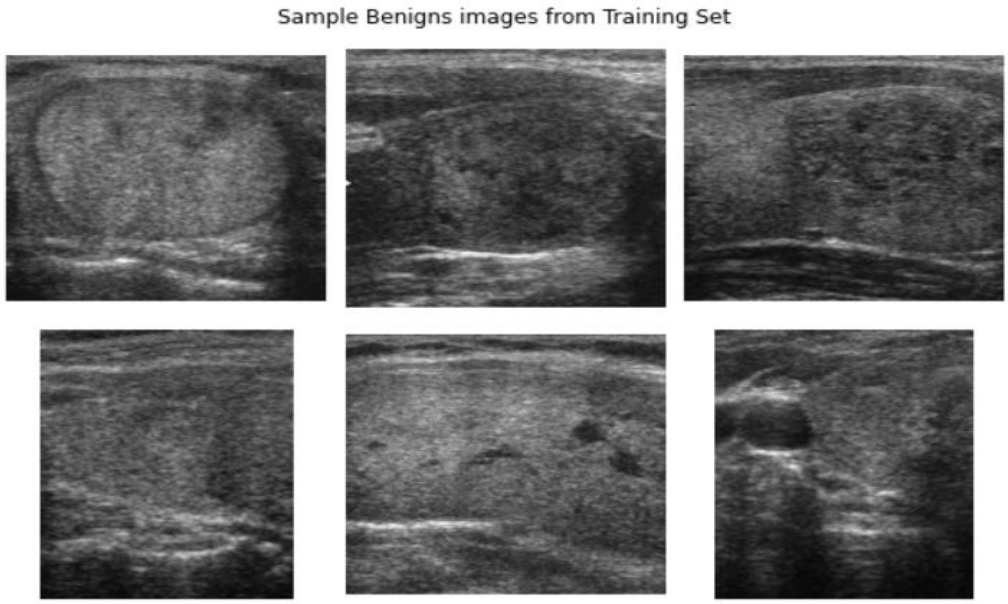



**Figure 2.7: Sample of Benign images for thyroid [39]**

Figure (2.8) shows a sample from the training set of ultrasound medical images diagnosed with maligns thyroiditis. This sample was taken from the training set images.

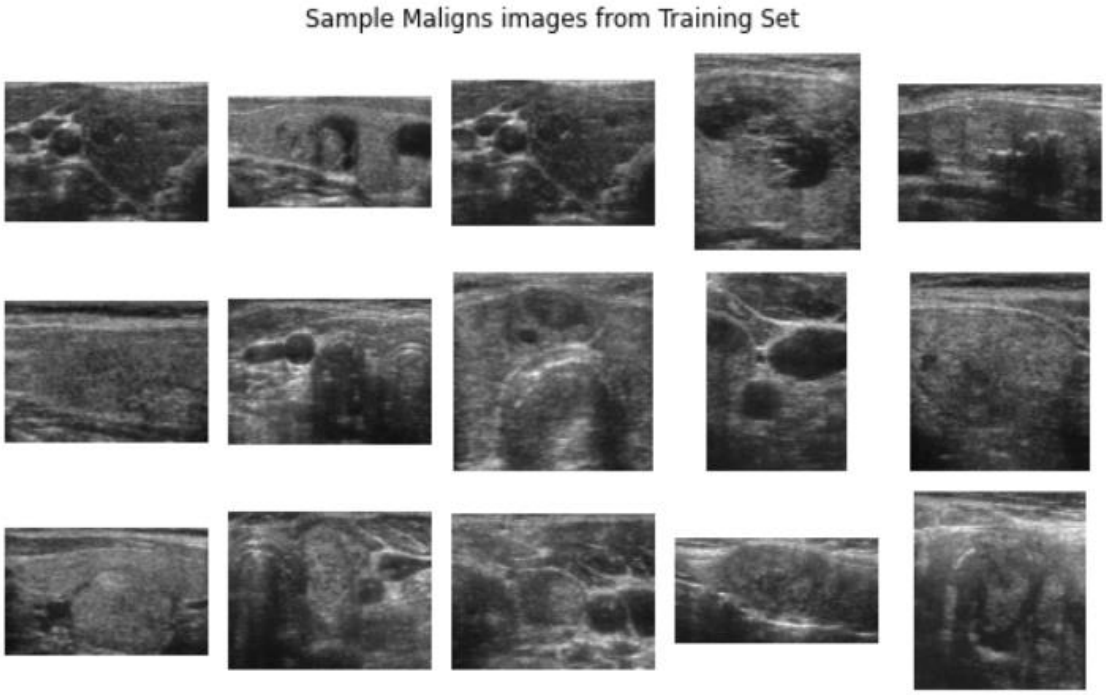



**Figure 2.8: Sample of Maligns images for thyroid gland [39]**

## 2.3 Related Work

Most of the studies that focused on the use of machine learning in terms of classification and prediction of thyroid diseases were applied to a global dataset and some of them on local researcher's communities in the researcher's country, and different inputs and variables were relied on from various points of view, most of which were based on medical laboratory analyzes, and some of them presented comparison studies of different types of machine learning algorithms to prove the accuracy of a particular technique (algorithm) on a global dataset, in this review will present the most relevant studies from newest to oldest.

- **Machine Learning Related Articles**

Ioniţă, Irina, and Liviu Ioniţă. [1] analyzed and compared four classification models (Naive Bayes, Decision Tree, Multilayer Perceptron, and Radial Basis Function Network) in two of the most common thyroid dysfunctions (hyperthyroidism and hypothyroidism) among the population in Romania state, and the result of their study was the best classification rate being that of the Decision Tree model. Umar Sidiq et al. [2] used a data set was taken from one of the recognized labs of Kashmir, and they proposed a classification model is trained using classification algorithms like K-nearest neighbor (KNN), support vector machine (SVM), decision tree (DT) and Naïve Bayes (NB) for the diagnosis of thyroid diseases. (The Decision Tree outperformed other techniques as the result of their study with accuracy (98.89%), compared with other techniques such as (KNN 91.82%), (SVM 96.52%), and (NB 91.57%)). They used the most influencing variables in disease identification or classification (age, gender, TSH, T3, and T4) as a variable to their classification model. Duggal, Priyanka, and Shipra Shukla [3] proposed three classification techniques that have been used namely Naive Bayes, Support vector machines, and Random Forest. Their results show that the Support Vector Machines are the most accurate technique (92.92%) compared with Naive Bayes (74.37%), Random Forest (78.21%). And hence this was used as a classifier to separate the symptoms of thyroid diseases into four classes namely Hypothyroid, Hyperthyroid, Sick Euthyroid, and Euthyroid (negative). They used (‘Age’, ‘ Sex’, ‘TSH’ , ‘TT4’, ‘T4U’ , ‘T3’, ‘FTI) as a variable to their classification model.

Tyagi et al. [4]. They used several techniques in the field of machine learning (support vector machine (SVM), K-NN, Decision Trees) to predict the estimated risk on a patient's chance of obtaining thyroid disease based on the information gathered from the dataset taken from UCI (University of California, Irvine Machine Learning repository). They used several variables ('Age', ' Sex', 'TSH', 'TT4', 'T4U', 'T3', 'FTI', 'TBG'), and their results shows that support vector machine and decision tree has also performed well, In their studies, they showed that the use of few variables saves time, effort and money for the patient, but undoubtedly not at the expense of accurate diagnosis of the disease. Chandel, Khushboo, et al. [5] used various classification models are used to classify thyroid disease (K-nearest neighbor, support vector machine, and Naive Bayes) based on the parameters like TSH, T4U, and goiter. And the results show that the K-nearest Neighbor technique is more accurate (94.44%) than other techniques.

Prerana, P. S., & Taneja, K. [6] They used a systematic approach for earlier diagnosis of Thyroid disease using back propagation algorithm used in the neural network, they used dataset is taken from UCI machine learning repository, consists of patients records. Each record is having 29 attributes. They proposed a predictive neural model to classify thyroid disease in MATLAB Neural Network Toolbox software. FTI (Free Thyroxin Index) values (one of 29 attributes in the dataset) have been taken as input to classify in three different classes from 1-3 values. While training the neural network with error backpropagation in conjunction with gradient-based training methods, from their experiments they conclude that the Levenberg Marquardt method has shown better performance in comparison with the simple gradient descent algorithm. In addition, their observations conclude that the error accuracy limit achieved

by the Levenberg Marquardt method is superior and it trains the models to an accuracy level of the order of 10-5 for the applied data set. Yadav, Dhyan Chandra, and Saurabh Pal. [8] they proposed two main techniques for mining the hidden pattern in the dataset. Ensemble-I and Ensemble-II both are machine learning techniques. Ensemble-I generated from the decision tree, overfitting, and neural network. Ensemble-II generated from combinations of Bagging and Boosting techniques. Their results show that (Bagging+ Boosting) ensemble-II model is the best compared to others. Their dataset was collected from some different sources, like Rahul pathology, Chandan pathology (Indian data), and some data collected from the website. The classifier to separate the symptoms of thyroid diseases into 3 classes namely Hypothyroid, Hyperthyroid, and Euthyroid (negative). They used (Clinical symptoms) as a variable in their classification model. Jamkhandikar, Dayanand, and Neethi Priya.[9] they made a comparative study using different ML Techniques to predict thyroid disorder at an earlier stage. They were analyzing naïve Bayes, k-nearest neighbor, and Support Vector Machine (SVM) for multiclass classification of thyroid dataset (global dataset from UCI). Their study shows that the KNN technique was the more accurate technique to predict thyroid disease. Raisinghani, Sagar, et al. [10] made a comparative study using different ML Techniques to predict thyroid disease. They have used decision trees, random forest, support vector machine, artificial neural network, and logistic regression on a global dataset from UCI. Their study concluded that the decision tree algorithm gives the most accurate and precise results with an accuracy of 99.46% and a precision of 0.99. Challa, Manoj. [11] The authors have deliberated the implementation of four classification approaches namely, Naïve Bayes, Decision Tree, KNN, and SVM on UCI Machine Learning Repository thyroid data set for accurate prediction of hypothyroidism and

hyperthyroidism disorder. Their study showed that the best classification approach was the decision tree approach in all the effectuated implementations.

- **Deep Learning Related Articles**

Hou, Yiqing, et al. [13] developed a model using deep learning based on the diagnosis of Hashimoto's Thyroiditis. Through the data of patients diagnosed with the disease, an ultrasound scan of the affected gland was taken. The model showed a high performance in diagnosing normal and diffuse sub-parenchymal thyroid nodules in patients with Hashimoto's thyroiditis; the model showed better performance compared to radiologists with different years of experience.

Shankarlal, B., P. D. Sathya, and V. P. Sakthivel. [14] they introduced a model capable of detecting and segmenting a tumor in the thyroid gland using machine learning and deep learning models. They used a set of thyroid images (ultrasound images) in their input data set. Their model was initially based on the use of an edge detector (Kirsch) to optimize the edge area pixels in the thyroid image and then a Dual-Tree Contourlet Transform (DTCT) was applied to the enhanced image to obtain the coefficients. Next, the features are computed from the converted thyroid image, and these features are trained and classified using the Co-Active Adaptive Neuro Expert System (CANFES) classifier. Then, the morphological segmentation method is applied to the abnormal image of the thyroid gland to divide tumor areas. Finally, the Convolutional Neural Network (CNN) algorithm is implemented on tumor areas segmented to diagnose mild, moderate, and severe. Liu, Yuejun, et al. [15] used four deep learning models (CNN, Inception, VGG16, and RNN) to classify a set of medical images of the thyroid gland (SPECT images) into hyperthyroidism, normal, and

hypothyroidism, and pre-treat the images. Their results indicated that the use of an auxiliary deep learning-based diagnostic method is more accurate than many of the other methods they mentioned in their study.

As previous studies showed that the use of machine learning models gives accurate and strong results in terms of classifying the famous thyroid diseases: hyperthyroidism and hypothyroidism, the application of machine learning models like (Decision Tree (DT), Naïve Bayes (NB), K-Nearest Neighbors (KNN), Support Vector Machine (SVM), Ensemble and Multi-layer Perceptron Neural Networks (MLPNNs) to local datasets (Palestinian datasets) will give a new addition to Scientific research in this area. Furthermore, the application of hybrid models (GAs-MLPNNs) to a local dataset (GAs-MLPNNs) also improved accuracy. Also, in this research, a VGG-16 model was used to classify medical images of the thyroid gland and categorize the tumor into benign or malignant based on a convolutional neural network (CNN).

# Chapter 3

# Methodology

## 3.1 Proposed Method

This chapter of the thesis explains the methodologies and models in Machine Learning (ML) that will be used and applied to datasets to make a comparison of classification algorithms. Then determine the appropriate model or classifier to improve classification accuracy in cases of thyroid disease. We start with the global dataset of thyroid disease from (UCI) and then work on collecting the local dataset (Palestinian dataset) and describe the preprocessing steps. After that, the deployed models will be described: Decision Tree (DT), Naïve Bayes (NB), K-Nearest Neighbors (KNN), Support Vector Machine (SVM), Ensemble and Multi-layer perceptron Neural Network (MLPNN), then comparing the classification accuracy between the results of two datasets. Also, we used a hybrid model are a (GAs-MLPNNs), where GAs as one of the evolutionary algorithms will use to find a set of optimal weights for MLPNNs. To enrich the thesis, we will turn to deep learning (DL) and build a model based on a convolutional neural network (CNN) implemented by the (VGG-16) approach to classify the ultrasound images dataset. Finally, we will explain the performance metrics used to evaluate the model's performance. Figure (3.1) shows the block diagram of the Machine Learning (ML) model, and figure (3.2) shows the block diagram of the Deep Learning (DL) model.

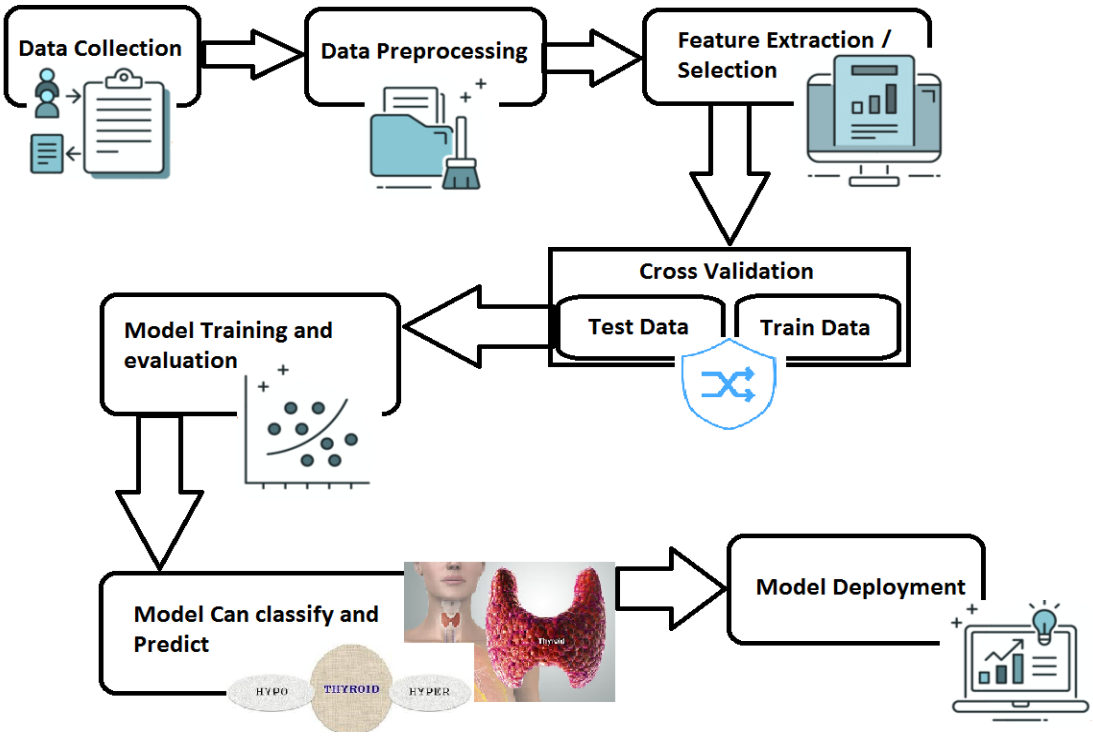



**Figure 3.1: Block diagram of the Machine Learning (ML) models**

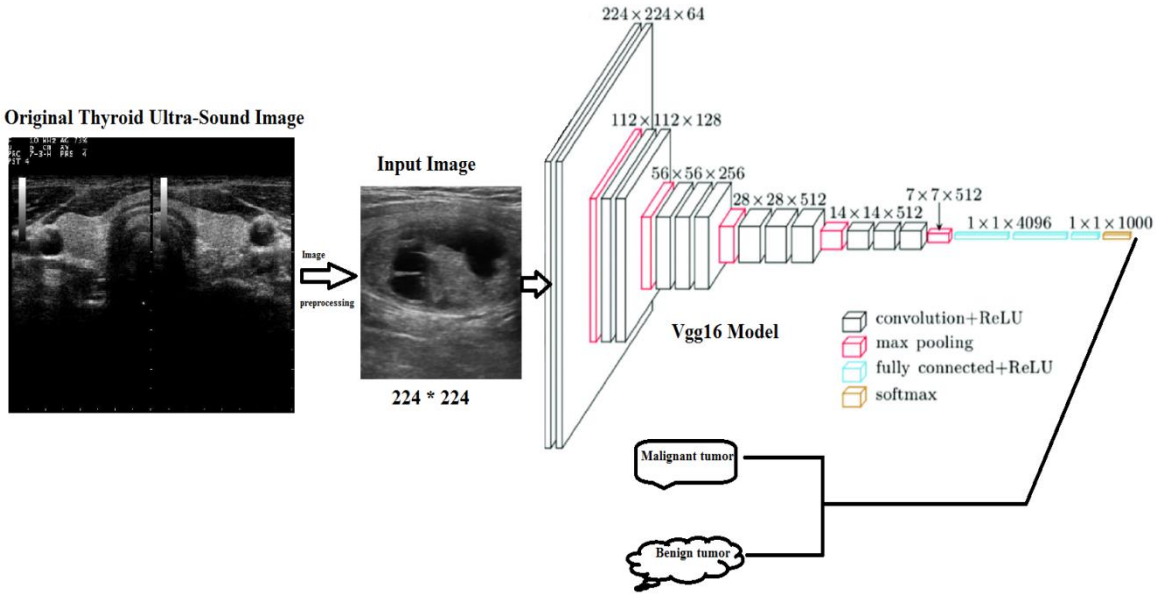



**Figure 3.2: Block diagram of the VGG-16 model [76]**

## 3.2 Data Collection

Three datasets will be used; one global dataset form (UCI), the other is a local dataset, and a global thyroid image dataset from (DDTI). In this phase, our interest was in collecting local datasets; because the data was not easily available, as data were collected from private medical centers, government hospitals, and others affiliated with the United Nations Relief and Works Agency (UNRWA) for Refugees. The local dataset data were collected from four Palestinian cities, which are: Nablus, Qalqiliah, Salfeit, and Ramallah.

## 3.3 Data Preprocessing Phase

This stage is considered a very important stage to prepare the data before training it in different models in machine learning or deep learning [46]. For the preprocessing step exist many operations such as aggregation, normalization, feature selection, data cleaning, and dealing with missing values. In this thesis, we used the technique of dealing with missing values, as well as the feature selection technique. This section will describe these steps in detail.

### 3.3.1 Feature Selection (FS)

The Mutual information (MI) technique was applied to arrange the variables most related to the target. Mutual information (MI) is the degree of the quantity of information that one random variable has about another variable [47]. This explanation is useful within the context of feature selection because it gives a way to quantify the

relevance of a feature subset with respect to the output vector C. MI is defined as follows [48]:

$$I(x;y) = \sum_{i=1}^{n}\sum_{j=1}^{n} p\big(x(i),y(j)\big).log\left(\frac{p\big(x(i),y(j)\big)}{p\big(x(i)\big).p\big(y(j)\big)}\right), \quad (3.1)$$

Where MI is 0 when x and y are statistically independent. P(x(i), y(j)) is the joint probability mass function of x and y, and P(x(i)), P(y(j)) are the marginal probability density functions of x and y respectively.

The feature selection was applied to local as well as global datasets, and the results of this (FS) appear as shown in table (3.1) and table (3.2) respectively. The (FS) technique was implemented by python code using (MI) method.

**Table 3.1: The Selected Feature and its Weights for Local dataset**

| Variable | Weight |
|---|---|
| TSH | 0.904767 |
| FT4 | 0.407761 |
| FT3 | 0.367434 |
| Age | 0.045358 |
| City | 0.005577 |
| pregnant | 0.003485 |
| Treatment | 0.000000 |
| sex | 0.000000 |

**Table 3.2: The Selected Feature and its Weights for Global dataset**

| Variable | Weight |
|---|---|
| TSH | 0.226309 |
| FTI | 0.106900 |
| TT4 | 0.094447 |
| T3 | 0.091485 |
| T4U | 0.014352 |
| On_thyroxine | 0.009191 |
| Query_hyperthyroid | 0.003890 |
| AGE | 0.003344 |

| | |
|---|---|
| SEX | 0.001868 |
| Sick | 0.001650 |
| Query_hypothyroid | 0.001083 |
| Pregnant | 0.000339 |
| Thyroid_surgery | 0.000000 |
| I131_treatment | 0.000000 |
| Lithium | 0.000000 |
| Goitre | 0.000000 |
| Tumor | 0.000000 |
| Hypopituitary | 0.000000 |
| Psych | 0.000000 |
| On_antithyroid_medication | 0.000000 |
| Query_on_thyroxine | 0.000000 |

### 3.3.2 Dealing with Missing Values

There are many methodologies used to deal with missing values in datasets, this is one of the great challenges faced by workers in the field of machine learning research, data science, and statistics; this is because making the right decision in this issue would result in powerful data models [49]. The methods that most researchers in this field use are: deleting the records that contain missing value/s, replacing the missing value/s with (mean/median/mode) of the variable, assigning a unique category, predicting the missing value, and/or using algorithms which support missing values [50]. While collecting data, we encountered some missing values and processed them using the (mean/median/mode) method. This was according to the location of the missing element in the data (in any variable) and the choice of the method closest to finding the true value.

## 3.4 Building Models Phase

In this part of the methodological chapter, we will highlight on the different models that were applied to the different datasets used in this thesis. The models were built and

applied to the data in several parts. The first part: applying several models of machine learning models to global and local datasets like Decision Tree (DT), Naïve Bayes (NB), K-Nearest Neighbors (KNN), Support Vector Machine (SVM), Ensemble and Multi-layer perceptron Neural Network (MLPNN), and then making a comparison between the classification accuracy between them on the same models. The second part: applying the hybrid model (GAs-MLPNNs) on the local and global datasets to get more enhancing in the accuracy result for classification, and then making a comparison between the results. The third part: application of the (VGG-16) model to classify ultrasound images dataset.

- **First Part**

In this part, we discuss the machine learning algorithms that we used to classify local and global data, the figure (3.3) shows the scheme of the algorithms in general.

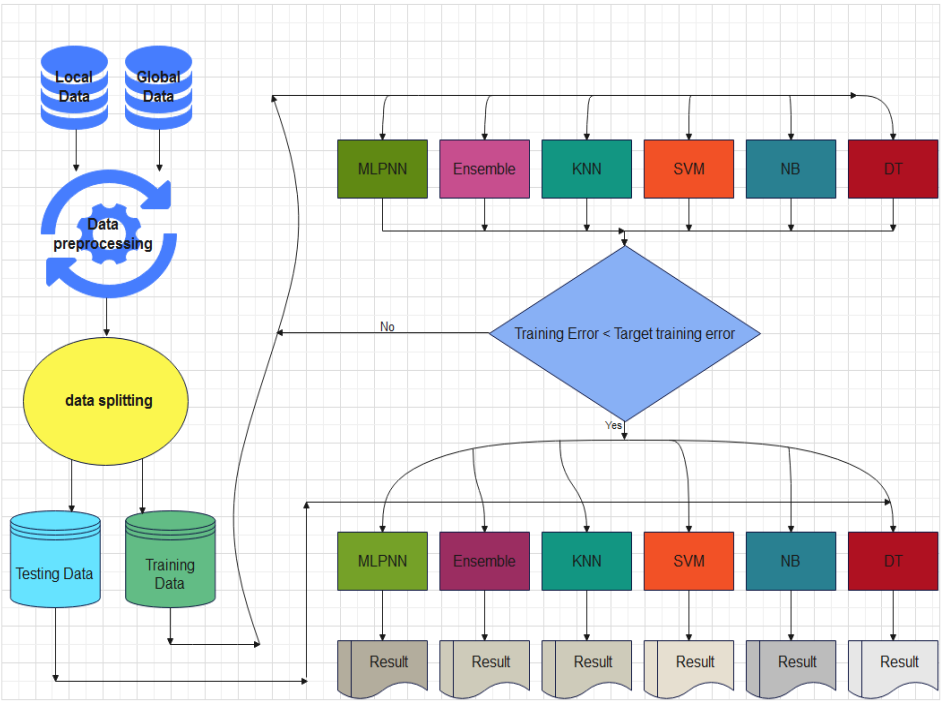



**Figure 3.3: The general Block diagram for first part (classification the datasets by all Models)**

### 3.4.1 Decision Tree (DT)

The decision tree algorithm falls into the category of supervised learning algorithms [51]. DT algorithm can be used to solve regression and classification problems as well [52]. The DT algorithm attempts to solve the problem using a tree representation. Each inner node of the tree corresponds to an attribute, and each leaf node corresponds to a class label [54]. A DT is used to start from observations about a particular element and to conclude about the value of that element represented by the leaves of the tree, while the same element is represented by the branches of the tree. DT algorithms produce a set of (Decision Rules) through which the class (test result) is found. It is based on the principle of dividing into nodes from top to bottom, then dividing the data set into smaller and smaller subgroups until the target nodes (class) are reached [55]. It starts from the root node, which contains all the records, and then divides according to the (Class Label) column -it is the column on which the classification is built-. In general, this process is used to categorize some populations into branches or stages by constructing a tree consisting of nodes and branches as shown in figure (3.4) [56].

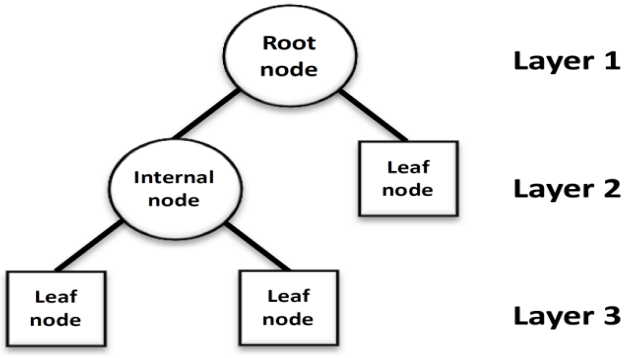



**Figure 3.4: Decision Tree general Structure**

The procedure of building a DT in general:

1- Splitting: It is the process of dividing the data set at each node into two (two sets) or more according to the attribute descriptors (column) at the next level. The division process is based on two scales:

a. A measure of randomness "entropy": This concept was invented by one of the pioneers of information theory Claude Shannon [57] and the basic concept of entropy is a measure of the amount of chaos in a system, and here the measure indicates the amount (probability) of variance in a node (that is, does a node contain more than one class?). Since it is a probability, it ranges between [0, 1], and if the entropy is 0, then the sub-data set in this node is pure (that is, it is of only one class), but if it is equal to or close to 1 then this means that the randomness is high (that is It contains data of more than one category and still needs to be divided). Its equation is:

$$E = -\sum_{i=1}^{N} pi \log_2 pi \qquad (3.2)$$

Where (Pi) is simply the frequentist probability of an element/class 'i' in our data

b. Information Gain: This metric is calculated to determine the best attribute (column) to split the data in a node so that this attribute is the most pure. Divide the data from others for this branch, its equation:

$$Information\ Gain\ =\ 1 - Entropy \qquad (3.3)$$

2- Stopping: To not make a model go in a complicated direction; we must take this step. There are some rules to stop which include minimizing: the number of

records in the leaf, the depth or number of levels (number of steps from the root node to the leaf node), and the number of records before the splitting.

3- Pruning: As we have said before, the decision tree method takes into account all records during fitting, and for this reason, it is subject to (overfitting) that the data may contain noise; that is, because it processes all records, the accuracy of training becomes very high, but when testing we get a big error because the model Become trained and fit on the training data only, as the best tree is the one with high precision with simpler installation and lower depth (in short, fewer knots with high accuracy), so we prune to prune the tree [58].

### 3.4.2 Naïve Bayes (NB)

It is a probability-based machine learning method. It is used to classify data, this method assumes that the properties used to build the model are separate from each other, so changing the values of one property does not affect another property. It is one of the very fast algorithms when doing classification, and therefore it is used in the real-time classification of data [59]. It is easy to build a naive Bayesian model, without complex iterative parameter prediction, which makes it particularly useful for very large datasets [59]. Despite its simplicity, the Naive Bayesian classifier often works surprisingly well and is widely used because it often outperforms more complex classification methods. The equation for the (NB) classifier is based on Bayes' Theorem is stated as:

$$P(h|d) = \frac{(P(d|h) * P(h))}{P(d)} \qquad (3.4)$$

Where P(h|d) is the possibility of hypothesis h given the data d. This is called the posterior probability. P(d|h) is the possibility of data d given that hypothesis h was true.

P(h) is the possibility of hypothesis h being true (regardless of the data). This is called the prior probability of h.P(d) is the probability of the data (regardless of the hypothesis).

### 3.4.3 Support Vector Machine (SVM)

SVM used for both classification and regression tasks. But mostly the SVM is used for classification. It is based on the idea of finding a (hyper-plane) [60] that divides the data set into two classes as the simplest level of classification in the best way, as shown in figure 3.5, is a hyper-plane divides data into two classes, and the two points closest to the hyper-plane are the supporting vectors.

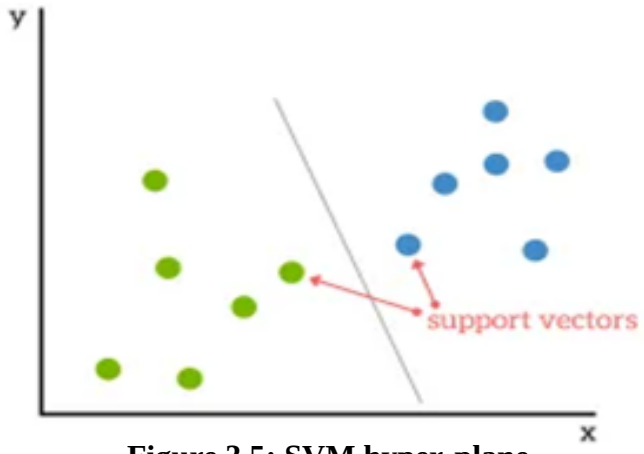



**Figure 3.5: SVM hyper-plane**

To understand the mechanism of action SVM, it is necessary to address some of the terms:

- **Support Vectors**

Supporting vectors are the data points closest to the hyper-plane and are the points that, if removed from the data set, will change the location of the hyper-plane that divides the data. So these points can be considered as the important elements in the data set.

❖ **Hyperplane**

As a simple example, let's take a classification process for a data set with only two properties figure (3.5). As a figure shown, the hyper-plane as the line that linearly separates and classifies the data set. Intuitively, the further away the points are from the hyper-level, the more confident we are that the points are ranked correctly. So we want the points to be as far from the hyper-plane as possible keeping in mind that they stay on the right side of the dividing line. So when we add new data to test it will be categorized based on which side it is on for the hyper-plane.

The distance between the hyper-plane and the nearest point of any data set is known as the margin. The goal is to choose a hyper-plane with the largest margin between it and any point in the training data set, increasing the likelihood that any new data will be correctly classified. Figure (3.6) shows the margin, which is the distance between the hyper-plane and its nearest points.

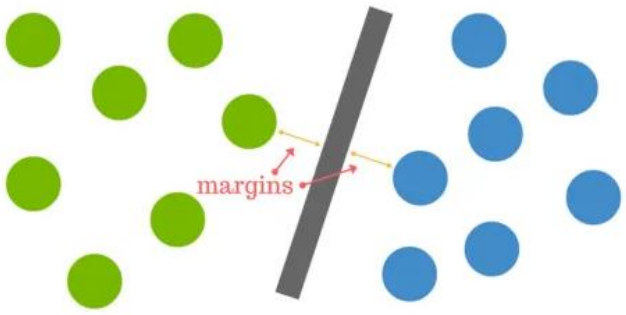



**Figure 3.6: SVM mechanism of classification & margin distance**

To understand hyper-plane equations and how to create it, given training dataset:

$$(x_1,y_1), \ldots, (x_n,y_n),\ x_i \epsilon\ R^d \text{ and } y_i\ \epsilon\ (-1,1) \quad (3.5)$$

Where $X_i$ is the symbol of the feature vector and $Y_i$ is the class label (1 or 0) of training compound i. Hence, to express the optimum hyperplane equation:

$$wx^T + b = 0 \quad (3.6)$$

Where W is the weight vector, X is the feature vector and b is the bias. b and w must satisfy the variances for the elements of the training dataset:

$$wx_i^T + b \geq +1 \text{ if } y_i = 1 \quad (3.7)$$

$$wx_i^T + b \leq -1 \text{ if } y_i = -1 \quad (3.8)$$

The objective of the training dataset on SVM is to determine the suitable B and W to separate the data with optimum hyper-plane and the max margin distance $1/\| w \|^2$ [61]. Vectors For $x_i$ that $|yi| (wx_i^T + b) = 1$ (will be called support vector). In the end, we will talk about the pros and cons of the vector support machine. The Precision, works well on small and refined data sets, and it can be more efficient because it uses part of the training data, all are advantages of the SVM model. In contrast, not suitable for large datasets as the time required to train the data is high and it is considered less effective with the noisiest and overlapping data between classes.

### 3.4.4 K-Nearest Neighbor (KNN)

KNN is considered one of the most important and simplest directed ML algorithms - which work with a supervisor - and the Nearest Neighbor algorithm is considered to be one of the descriptive and predictive classification algorithms. It can deal with anomalous data [62].

The principle of work of this algorithm depends on calculating the Euclidean distance equation (3.9) between points (It is one of the equations for measuring the distance between points) and other equations [62], where the less the distance between two points, the greater the possibility of the point belonging to each other, hence the name of the algorithm, and the letter K refers to the cases that will be classified based on the distances between them (neighbors). As in figure 3.7.

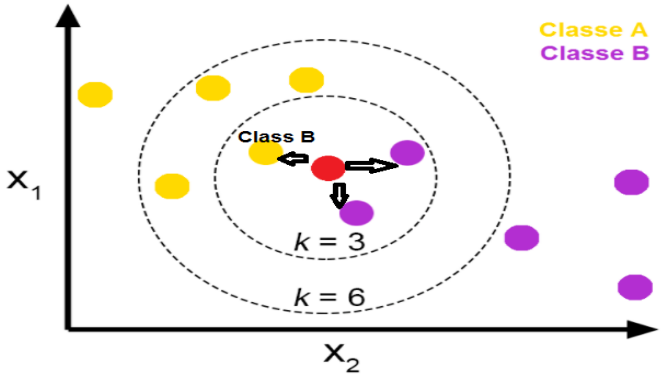



**Figure 3.7: KNN mechanism in classification**

The Euclidean distance's equation:

$$d(p,q) = d(q,p) = \sqrt{(q1-p1)^2 + \dots\dots + (qn-pn)^2} \quad (3.9)$$

Where (p,q) are points one of them is a K point and another is the data point. Figure 3.8 shows the flowchart of the KNN classifier procedure [63].

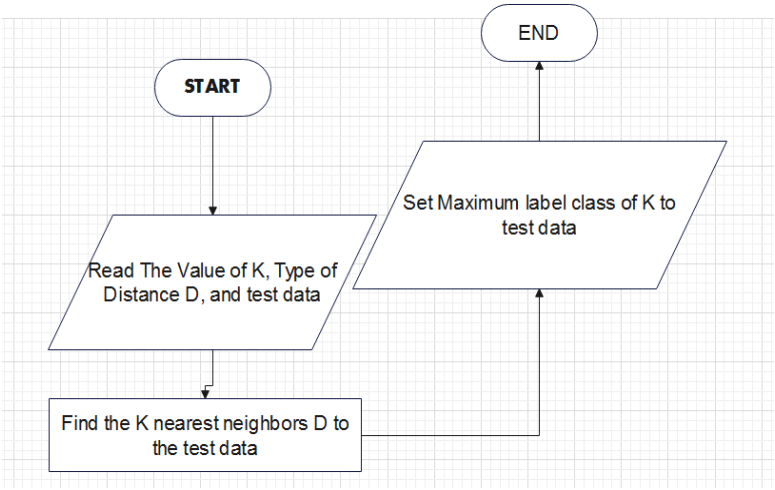



**Figure 3.8: The flowchart of KNN classifier procedure**

### 3.4.5 Ensemble Algorithm

Ensemble algorithms are an aggregation of several models to improve the accuracy of the results, rather than using a single model that leads to less accurate results. Studies indicate that compiling several ML models that are identical in their work and simple in their performance achieve the performance of a complex model in less training time for data [64]. Ensemble methods depend on Boosting and Bagging techniques which are proven to be effective but require frequent resampling of training data [65]. Apply of ensemble algorithms makes the model more valuable and more accurate because it acquires the features of more than one classifier and combines the features of the different classifiers into one classifier [66].

- ***Bagging Ensemble Algorithm***

The working principle of this technique from the ensemble method is that by making a group from (weak learner models) together they become (strong learner model). Bagging grows many decision trees [67]. Where each decision tree is an individual (weak learner), while a group of collective decision trees is a (strong learner). Figure (3.9) shows this technique.

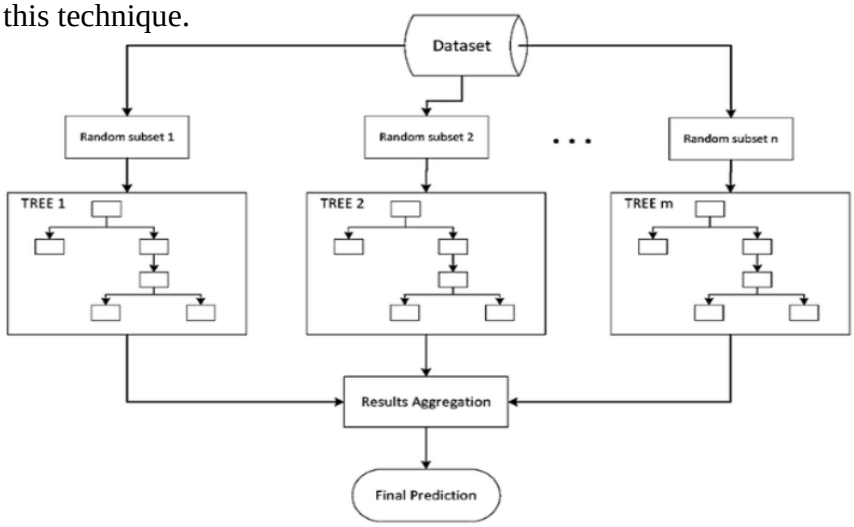



**Figure 3.9: Bagging Ensemble architecture**

- ***Boosting Ensemble Algorithm***:

The principle of its work is based in the same way as the “Bagging” by strengthening the weak learner, but in it, the wrong results are taken and trained until the appropriate model is reached. It is in contrast to the Bagging technique which is a parallel aggregation method; the boosting methods are sequential aggregation algorithms where the weights are modified to reduce the error rate [68].

- ***Random Forests Ensemble Algorithm:***

A random forest is a collection of many individual decision trees which is provided to be diverse by using random samples to construct each tree in the forest [69].

### 3.4.6 Multi-Layer Perceptron Neural Networks (MLPNNs)

Artificial neural networks (ANNs). It consists of an interconnected structure of artificially produced neurons that act as pathways for transmitting data [70]. Multi-layer perceptron neural networks (MLPNNs), the most widely used type of NNs [71]. MLPNNs consist of an input layer, an output layer, and one or more hidden layers. Each MLPNNs layer contains one or more neurons that are directly connected to the neurons of the previous layer and the next. Each connection has a value called “weight”, which is determined using the learning process. Figure (3.10) shows the general structure of MLPNNs.

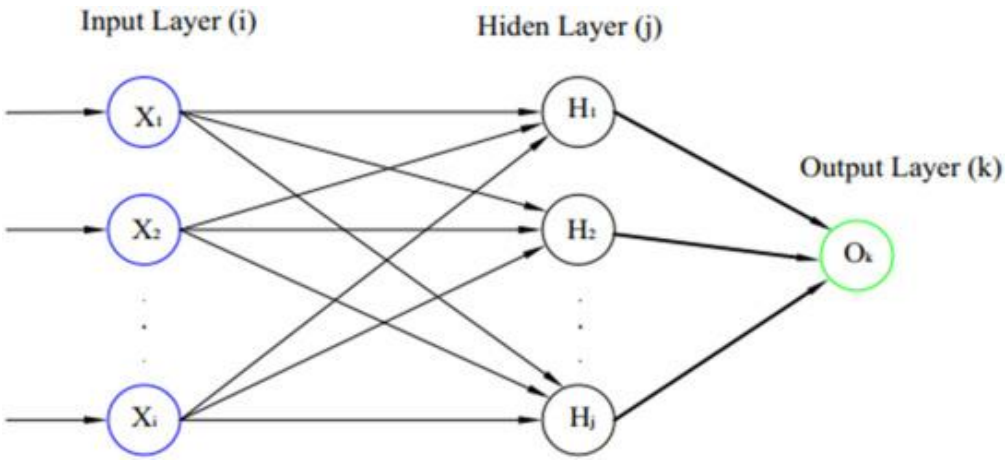



**Figure 3.10: General structure of MLPNN**

MLPNNs have two stages: the first stage is forward. In this stage, the output is predicted and it calculates the then sent the error back to the backward prorogation stage. The second stage is backward propagation, through this stage; the error is propagated back through the network to modify the values of the weights and to make reduce the error value in the output layer.

We can explain how MLPNN work clearly through these main steps:

1. Data entry: In the first step we enter the training data for which we already know the correct output, after which it is processed through the hidden layers of the neural network.
2. Data processing by neurons (weights): At first, the input values are multiplied by the weights, according to this equation:

$$Y_{ij} = f(\sum_{i=1}^{n} Xi * w_{ij}) \qquad (3.10)$$

Where $w_{ij}$ : is the connection weight between the *i*th node in the input layer and the *j*th node in the hidden layer, and$x_i$: is the *i*th input, and *f* is the activation function. It is

worth noting that in each cell the layer is connected to all the cells in the next layer. The activation function *f* calculated as in the following equation:

$$Y_j = X_k = \frac{1}{1+e^{-y}} \quad (3.11)$$

Where, $X_k$ is the input to the next layer' node.

3. Repeat the process with the other hidden layers: These operations are repeated with the other layers in the NNs until the last layer, which is the output layer.
4. Determine the output at the end: The output values are ultimately determined by the type and function of the NNs, so we get the results and compare them with the real output.
5. Calculate the error: To break the process of training, there is a certain threshold θ is set depending on the error of the MLPNN which represents the difference between the desired and actual output. The error is calculated using the following equation:

$$MSE = \frac{1}{2}\sum_{i}^{n}(y_d - y_j)^2 \quad (3.12)$$

Where $y_d$ is desired output, $y_j$ is the actual output. So, by comparing the output with the results we already know.

6. Modify the weights to obtain more accurate results using the following equation:

$$\Delta w_{i+1} = \alpha . E . x_i \quad (3.13)$$

7. The back-propagation algorithm always starts from the last layer (output layer) and propagates backward to update the weights. In the output layer, the

difference value between the actual and the target value is$\Delta_k$, where the actual value of the node k is Yk and the target value is tk.

$$\Delta_k = t_k - Y_k \quad (3.14)$$

$$\delta_k = \Delta_k Y_k (1 - Y_k) \quad (3.15)$$

Where $\delta_k$: The error signal of the output layer is calculated by $\Delta_k$ and the derivative of the sigmoid function, the actual value of the node k is $Y_k$ and the target value is $t_k$.

8. The change in the weight between node j and node k is done by multiplying the error at node k by the output of node j by using the delta rule.

$$\Delta w_{jk} = l\, \delta_k X_k \quad (3.16)$$

$w_{jk}$: The weight between node j and k, where $l$ is the learning rate, so, to update it by the following formula:

$$w_{jk} = w_{jk} + \Delta w_{jk} \quad (3.17)$$

9. To calculate, the error signal $\delta_j$for node j in the hidden layer, $\delta_j$: The error signal for node j in the hidden layer is calculated by the following formula:

$$\delta_j = (t_k - Y_k) Y_k \sum w_{jk}\, \delta_k \quad (3.18)$$

wij is the weights between the input node i and the node j can be updated by using 15 and 16 so

$$\Delta w_{ij} = l\, \delta_j X_j \quad (3.19)$$

$$w_{ij} = w_{ij} + \Delta w_{ij} \quad (3.20)$$

The back-propagation algorithm repeats until the error on the output node is minimized.

- **Second Part**

In this part, we will explain a hybrid-mode called (GAs-MLPNNs), genetic algorithms (GAs) model used to find a set of optimal weights for neural networks, while Multi-Layer Perceptron Neural Networks (MLPNNs) model used to classify the data to increase the classification accuracy as much as possible.

### 3.4.7 Genetic Algorithms (GAs)

A genetic algorithm uses techniques inspired by evolutionary biology such as selection, mutation, inheritance, and recombination to solve a problem [73]. This search algorithm starts from a set of solutions Randomization ends with choosing one of the "sons" of solutions.

*Process of genetic algorithms:*

1. The population of chromosomes: This represents the search set or the search space, which is a set of solutions to the problem.
2. Selection: Choosing the appropriate chromosomes as "parents" to carry out the process of mating between them, but the process of this selection is not random, but rather depends on the efficiency of the chromosome (fitness).
3. Fitness: Coefficients that give each chromosome a certain value that indicates how efficient the chromosome is (its closeness to the solution) and based on it (the fitness coefficient) the chromosomes are selected.
4. Crossover: After selecting the appropriate chromosomes from the first generation, crossing over occurs through which chromosomes formed new (new offspring) are based on maternal chromosomes.

5. Random Mutation: After the formation of the new offspring, mutations are made (changes in its chromosome formula), and this helps to reach the solution faster

The genetic algorithms flowchart is illustrated in figure (3.11).

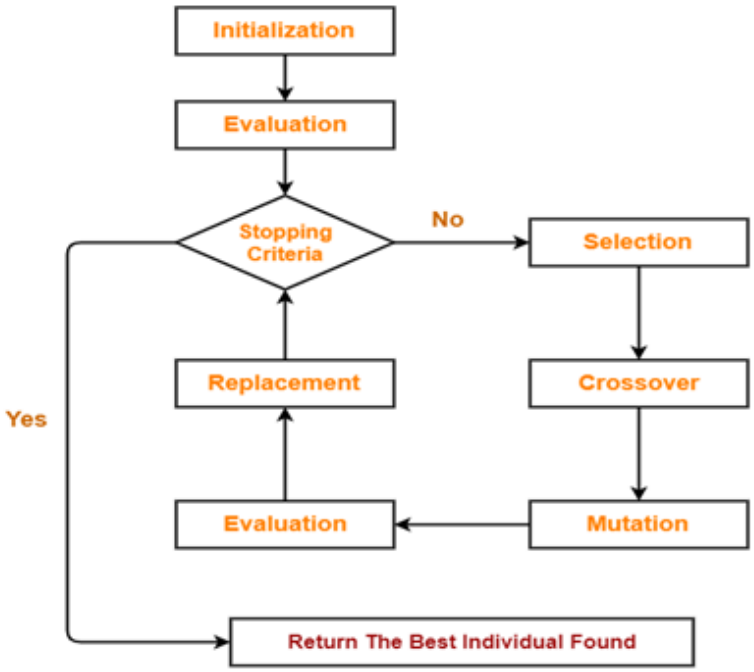



**Figure 3.11: Flow Chart for Genetic Algorithm Process**

- **Third Part**

In this part we will explain a model based on a convolutional neural network (CNN) implemented by the (VGG-16) approach, it is a part of Deep Learning (DL).

### 3.4.8 Visual Geometry Group-16 Model (VGG-16)

VGG-16 is a simple and widely model used CNN Architecture used for (ImageNet), which is a large visual database project used in visual object recognition software research. The VGG-16 Architecture was developed and introduced by Karen Simonyan and Andrew Zisserman from the University of Oxford, in the year 2014, through their

article “Very Deep Convolutional Networks for Large-Scale Image Recognition.” [76]. VGG-16 model which gave the best results in the 2014 competition (*ImageNet* Large Scale Visual Recognition *Challenge* ‘ILSVRC’, and it is a deep model consisting of 16 trained layers that can be used to display filters and feature maps because it has a simple unified structure consisting of layers convolution, maxing, and fully connected layers are arranged sequentially, and it has a very good performance [76], this means that the resulting filters will extract useful features. Figure (3.12) shows the basic architecture of the VGG-16 model.

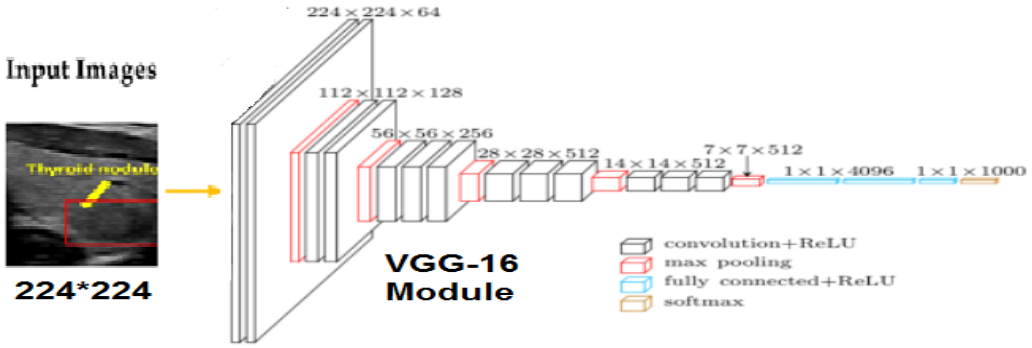



**Figure 3.12: VGG-16 Architecture**

Before starting to explain the VGG-16 model that is used in this thesis, we must talk about the basis on which this model is built, which is the convolutional neural network (CNN). CNN is a type of Deep Learning (DL) and is usually used in computer vision applications and visual scene analysis; it is characterized by the presence of one or more hidden layers that extract features in images or videos and a fully connected layer to produce the required output, and it depends on the application of neuroscience theories related to the mechanism of action the brain recognizes patterns [75] [77]. Figure (3.13) shows the general structure of CNN.

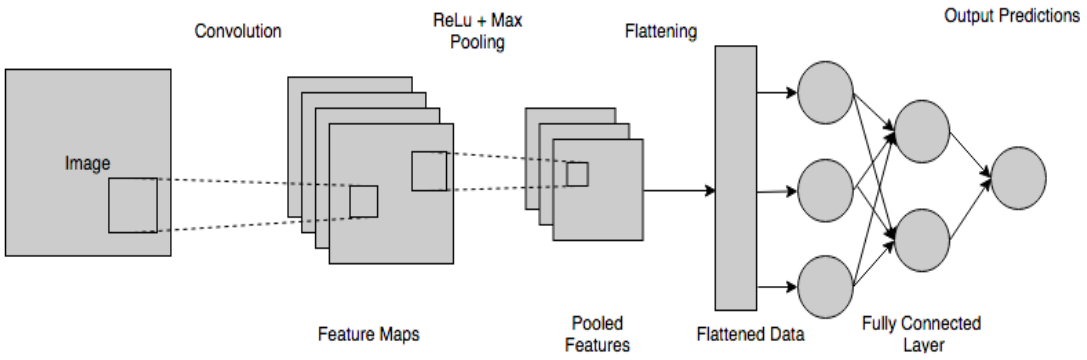



**Figure 3.13: General Structure of CNN**

*The General Steps of the CNN model:*

### 1. Convolution

The convolution process relies on a small matrix, which is the convolution kernel [24]. The pixel matrix of each layer above is continuously scanned in stages, the number of scans is multiplied by the number corresponding to the convolution kernel, and then the sum is calculated, each scan will get a value, and a new matrix will be generated when all scans are completed. As shown below in figure (3.14).

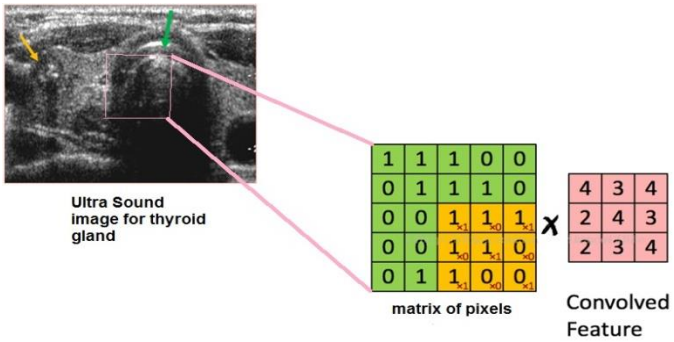



**Figure 3.14: Convolution Process on subset of image pixels**

Convolution kernels are regulated by the size and number of convolution nuclei of a CNN. The number of convolution layers is initially determined by a random initial value. When training the network, the network will constantly update parameter values through back-propagation until the best parameter value is found. It is evaluated by the

loss function. Convolution process equivalent to feature extraction and Convolution kernel equivalent to filter to extract the features we need.

- **Padding**

Before each convolution, a layer of zeros is added to the outside of the original array; this is to ensure that the filter passes over pixels in the image boundary the same number of times as it passes over inner pixels. Figure (3.15) shows the padding on the image.

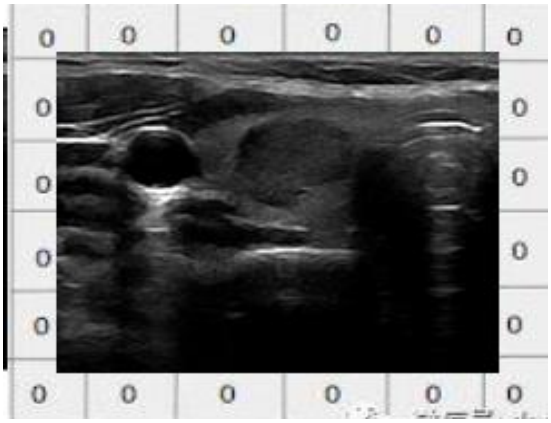


**Figure 3.15: Zero Padding added to the image**

- **Activation Function**

It is a nonlinear function that is a bridge to the next layer in a CNN. Its benefit is to reduce the number of completed accounts by not activating all the points of the feature maps at the same time, and in other words, it is to activate the points that represent the features and exclude the points that do not represent them, and it has several types, the most famous of which are: Sigmoid, TanH, and ReLU [53]. In this thesis, we used the most effective function among them, which is called ReLU or Rectified Linear Unit. Its equation is [53]:

$$f(x) = \begin{cases} x, & x > 0 \\ 0, & x \leq 0 \end{cases} \qquad (3.21)$$

When the value of x is negative, the result is f(x) = 0, while when the value of x is positive, it will be f(x) = x, i.e. we take the same value see figure (3.16).

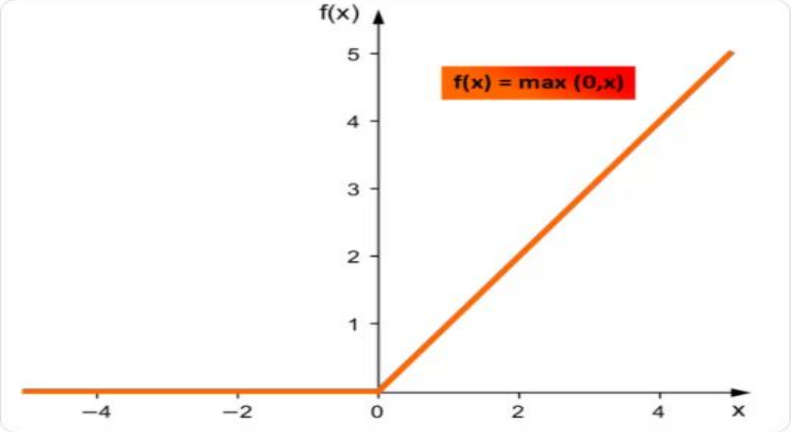



**Figure 3.16: ReLU activation function**

## 2. Polling

After the convolution process, we extracted a lot of feature information. Adjacent regions have similar feature information, which can be exchanged for each other. If all this feature information is preserved, there will be duplication of information, which increases the difficulty of computation. At this time, grouping is equivalent to reducing dimension operations. Pooling is taking the maximum or average value of the area in a small matrix area to replace the area [74]. The size of the micro-array can be determined when building the network. Figure (3.17) shows the Polling process.

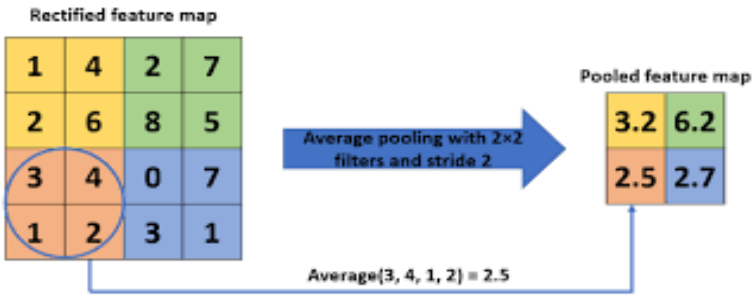



**Figure 3.17: Polling in CNN**

### 3. Flattening

Opening a multidimensional array and converting it to a one-dimensional vector; to enter this data into an artificial neural network later on. Figure (3.18) shows the process of flattening.

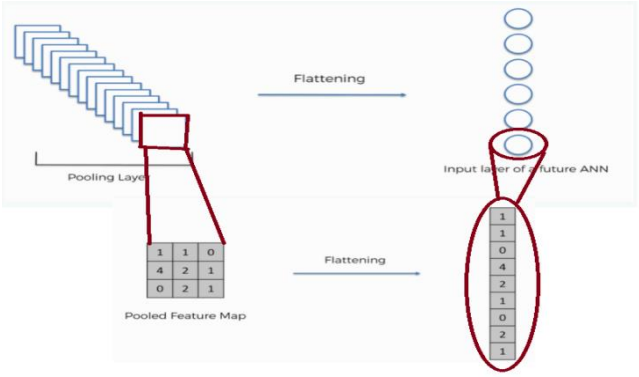



**Figure 3.18: Flatting in CNN**

### 4. Fully-Connected Layer

The role of is fully-connected layer is to take the data and combine the features into a wider variety of attributes that make the convolutional network more capable of classifying images, which is the whole purpose of creating a CNN. In this layer, every neuron is connected to every neuron in the next layer so that the output is based on the whole image, in this layer, the final classification of the images takes place. Figure (3.19) illustrates the fully-connected layer.

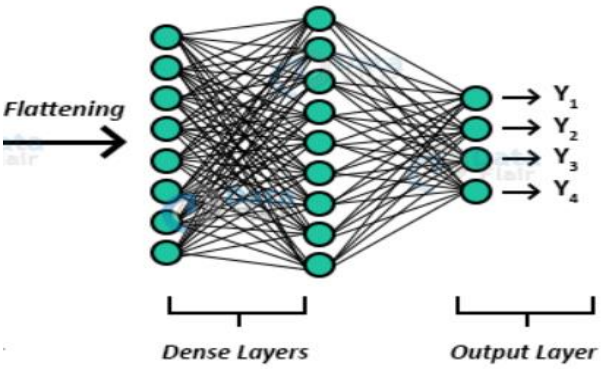



**Figure 3.19: Fully-connected Layer in CNN**

VGG-16 based on the CNN model is a pre-trained network based on the ImageNet image library of a large number of real images. The keras library corresponding to the VGG-16 model was trained. For us, in this thesis, we have transferred the acquired VGG16 weights to our CNN as initial network weights, so that our network does not need to start from scratch. A large amount of data is used for training to improve training speed. VGG16 is an overlay of 13 convolutional layers + 3 fully connected layers. The model can be simply divided into 5 stages, (this idea comes from the 5-layer convolution of AlexNet [78]) each layer two convolution kernels are grouped, and finally, three layers of fully- connected are used for classification.

Figure (3.20) shows the stages of a VGG-16.


**Figure 3.20: Stages of VGG-16 model**

Table (3.3) shows the details of the VGG-16 model for five stages.

**Table 3.3: Stages Implementation of VGG-16 model**

| Stage | Layer | Convolution | Output Dimension | Pooling | Output Dimension |
|---|---|---|---|---|---|
| First | Layer 1&2 | Convolution Layer of 64 Channel of (3*3) Kernel With padding = 1, Stride = 1 | 224*224*64 | Max Pool, Stride = 2 , Size = 2*2 | 112*112*64 |
| Second | Layer 3&4 | Convolution Layer of 128 Channel of (3*3) kernel With padding = 1, Stride = 1 | 112*112*128 | Max Pool, Stride = 2 , Size = 2*2 | 56*56*256 |
| Third | Layer 5,6,7 | Convolution Layer of 256 Channel of (3*3) kernel | 56*56*256 | Max Pool, Stride = 2 , | 28*28*512 |

| | | | | | |
|---|---|---|---|---|---|
| | | With padding = 1, Stride = 1 | | Size = 2*2 | |
| Fourth | Layer 8,9,10 | Convolution Layer of 512 Channel of (3*3) kernel With padding = 1, Stride = 1 | 28*28*512 | Max Pool, Stride = 2 , Size = 2*2 | 14*14*512 |
| Fifth | Layer 11,12,13 | Convolution Layer of 512 Channel of (3*3) kernel With padding = 1, Stride = 1 | 14*14*512 | Max Pool, Stride = 2 , Size = 2*2 | 7*7*512 |

Then flatting the resulting matrix to three fully-connected layers, the one and two-layer have 4096 neurons with activation function "ReLU", the last (output) layer has a number of classes neurons with activation function "softmax".

## 3.5 Performance Metrics Selection

In supervised ML techniques that we used in this study, we provide the required input and target, in addition to providing the accuracy of the predictions while training the algorithm. Once the algorithm has finished learning, it will apply what it learned to new data [40]. Some main performance metrics like; the accuracy, confusion matrix, and area under curve "AUC" were shown for all algorithms D-tree, NB, KNN, SVM, Ensemble, and MLPNNs. Before showing the results, the performance metrics should be clarified as follow:

- **Accuracy**: It is one of the simplest performance metrics we can use, is an index that measures the signal-to-noise ratio (SNR) of a given retrieval system, i.e. the percentage of related records detected to the total records detected in the sample [41]. It can be calculated according to this equation.

$$\text{Accuracy} = \frac{Correct\ Expectations}{Total\ Expectations} \qquad (3.22)$$

- **Confusion Matrix:** Also known as error matrix is a standard format for accuracy assessment, expressed as a matrix containing n rows and n columns. It is a correlation between model predictions or classification and actual category labels, or actual classification of data points [42]. In AI, a confusion matrix is a visualization tool, especially for supervised learning. In the confusion matrix, there are four different sets of expected and actual values, and here they are explained as follows:

  - ✓ True positive (TP): The true class of the sample is a positive example, and the result predicted by the model is also a positive example.
  - ✓ True Negative (TN): The true class of the sample is an example negative, and the model predicts it as an example negative.
  - ✓ False Positive (FP): The true category of the sample is negative, but the model predicts it as positive.
  - ✓ False-negative (FN): The true class of the sample is a positive example, but the model predicts it as a negative example.

If you want your model to be intelligent, the model must predict correctly. This means your true positives and true negatives should be as high as possible and at the same time, you need to reduce your errors for which your false positives and false negatives should be as low as possible. Also in terms of ratios, your TP ratio and TN ratio should be very high while FP ratio and FN ratio should be very low. Figure (3.21) shows the confusion matrix with four combinations [43].

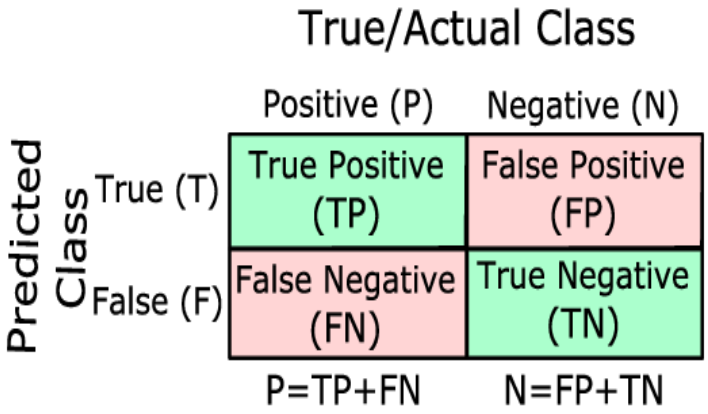



**Figure 3.21: Confusion matrix structure**

- **Sensitivity or Recall:** is a measure that quantifies the probability of actual positives. In other words, this test feature mainly focuses on identifying the members of the sample who are positive towards the tested characteristic. It can be calculated by this equation:

$$Sensitivity = \frac{TP}{TP+FN} \qquad (3.23)$$

- **Specificity**: Also known as a true negative rate (TNR), is a measurement that quantifies the probability of actual negatives. Privacy can also be easily put into an equation as follows:

$$Specificity = \frac{TN}{TN+FP} \qquad (3.24)$$

- **Precision:** This is the ratio of the number of retrieved records to the total number of retrieved records. It can be calculated by this equation:

$$Precision = \frac{TP}{TP+FP} \qquad (3.25)$$

- **Receiver Operating Characteristic Curve (ROC):** is curve is a performance metric for classification problems. It is a probability curve that plots the TPR against FPR at various threshold values and essentially separates the ‘signal’ from the ‘noise’. It is a

comprehensive indicator that reflects the continuous variables of sensitivity and specificity, each point on the roc curve reflects sensitivity to the same signal stimulus

In figure 3.21 “TPR” on the y-axis and False Positive Rate “FPR” on the x-axis fig where:

$$TPR = \frac{TP}{TP+FN} \qquad (3.26)$$

$$FPR = \frac{FP}{FP+TN} \qquad (3.27)$$

- **The Area under Curve (AUC):** It is easy to understand that when the FPR is relatively low, the TPR is relatively high, so there will be a better model, and the area under the corresponding curve will also be larger, so the area under the ROC can be used as an indicator to measure the classification algorithm. So the AUC denotes the areas under the ROC curve, which means the ability of the model to classify classes. Figure (3.22) shows AUC with the blue area and ROC curve.

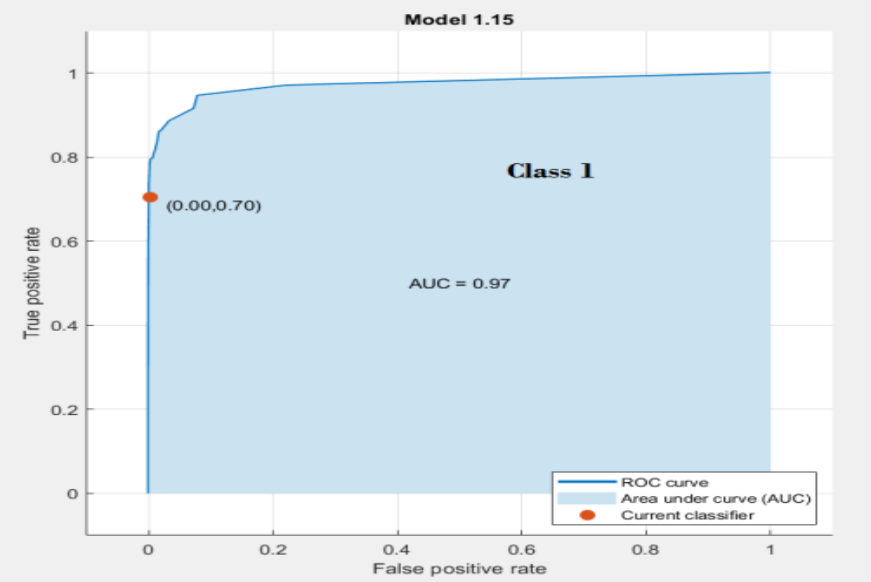



**Figure 3.22: AUC and ROC**

# Chapter 4

# Experiments and Results

## 4.1 Experiments and Results

In this chapter, we will apply the all classification models which we explained in the previous chapter to our datasets. Then, we review and discuss the Results. This chapter is divided into five parts: First, practical experiment on global datasets (UCI Dataset) and applying classification models (DT, NB, SVM, KNN, Ensemble, and MLP), the accuracy of each model will be measured. Also the sensitivity, specificity, and precision for each model will be calculated. The Area Under Curve (AUC), and Receiver Operation Curves (ROC) for best classification model will be calculated, based on the previous performance metric results that will be mentioned; the best classification model will be determined for global dataset. Second, practical experiment on local datasets (Palestinian Dataset) and applying classification models (DT, NB, SVM, KNN, Ensemble, and MLP), the accuracy of each model will be measured. Also the sensitivity, specificity, and precision for each model will be calculated. The Area Under Curve (AUC), and Receiver Operation Curves (ROC) for best model will be measured. This will be done based on each region (governorate) dataset, and the above will be measured within each region and a comparison will be made between each governorate dataset, then we determined the best model for classification based on the performance metric results for each region. Third, choosing the common variable of the global dataset (Custom global dataset) with the local dataset and deleting the variable is not common, then making a comparison between the two datasets (Custom and local) and applying classification techniques (DT, NB, SVM, KNN, Ensemble, and MLP), the

accuracy of each model will be measured, and the sensitivity, specificity, and precision for each model also will be measured. Then we determined the best model for classification based on the performance metric results for each dataset. Then, the comparison will be made between the two datasets (Local Dataset, Custom Global Dataset). Forth, (MLPNNs-GAs) model will also be applied on the local dataset and custom global dataset), and then choose a certain number of neurons within the model and compare the accuracy or performance for each choice. And a comparison will be made between the two datasets (Local Dataset, Custom Global Dataset) based on classification accuracy results. Fifth, a classification of ultrasound images of the thyroid gland and its classification in terms of the type of inflammation, malignant or benign, using a (VGG-16) model, it is a deep learning model based on convolution neural network (CNN), where the model had been built using Python.

## 4.2 Computing Environment

The Computing Environment in this work was done by ASUS X455 SnonicMaster Intel, Corei7-4510U CPU 2.00GHz 2.60GHz, RAM: 8.00GB, HDD: 1 TB with windows 8.1 pro. For applying experiments MATLAP_R2019b and Jupyter Notebook were used.

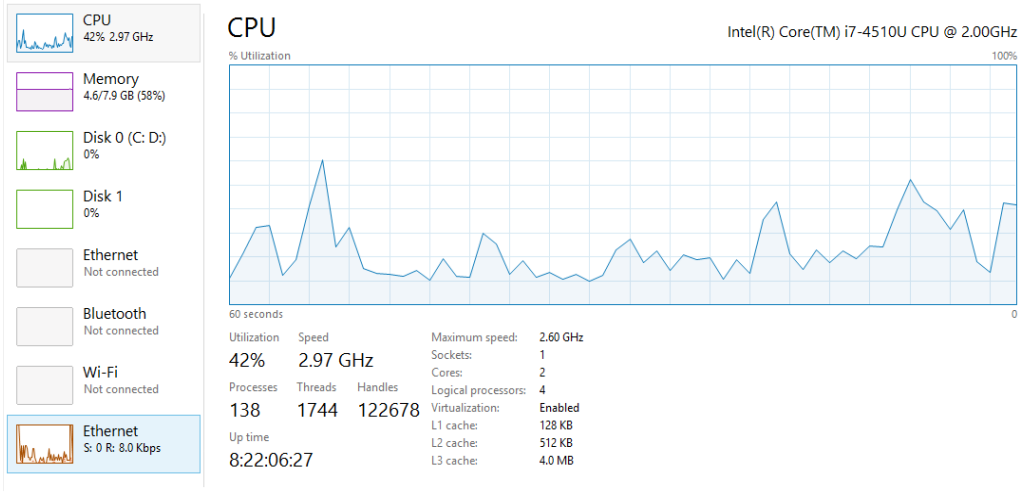



**Figure 4.1: The performance of ASUS device while running MATLAB nprtool**

## 4.3 Machine Learning Practical Experiment

### 4.3.1 Classification Results for Global Dataset

*First Experiment:* In this part of the result, we apply five classification algorithms to the global dataset (UCI dataset). They are as follows: decision tree (DT), Naïve Bayes (NB), support vector machine (SVM), K-nearest neighbor (KNN), and Ensemble. In this experiment, a K-fold cross-validation methodology was used to divide the dataset into a test set and training set to reach a good evaluation of the model. Cross-validation is used to evaluate the prediction performance of the model, especially the performance of the model trained on new data, which can reduce over-fitting to a certain extent [44]. You can also get as much effective information as possible from limited data. In the recent experiments, we used 5-fold cross-validation is used, 7200 records each fold 1440 records are obtained, 1 fold for testing randomly, and 4 folds for training.

Table (4.1) shows the results of applying the five models that we mentioned above on global; dataset, each model has four performance metrics: overall accuracy, sensitivity, specificity, and precision.

**Table 4.1: Classification results for different model on all variable global dataset**

| Model Type | Accuracy | Sensitivity | Specificity | precision |
|---|---|---|---|---|
| Decision Tree | 99.50% | 97.88% | 98.97% | 96.80% |
| Naive Bayes | 93.30% | 55.00% | 74.21% | 80.50% |
| SVM | 98.20% | 87.10% | 95.05% | 92.36% |
| KNN | 95.10% | 63.86% | 96.58% | 90.30% |
| Ensemble | 99.60% | 99.56% | 99.80% | 96.50% |

It was concluded that Ensemble is the best model type for classification of this dataset due to higher accuracy 99.60% and it is close to Decision Tree accuracy result of 99.50%; here 5-fold cross-validation was used, which (it) means that 20% of testing

data and 80% for training and the best accuracy obtained here is 99.60%. Also, the best sensitivity is 99.56% and specificity is 99.80%. Figure (4.2) illustrates the Comparison between performance metrics for all Classification Models.

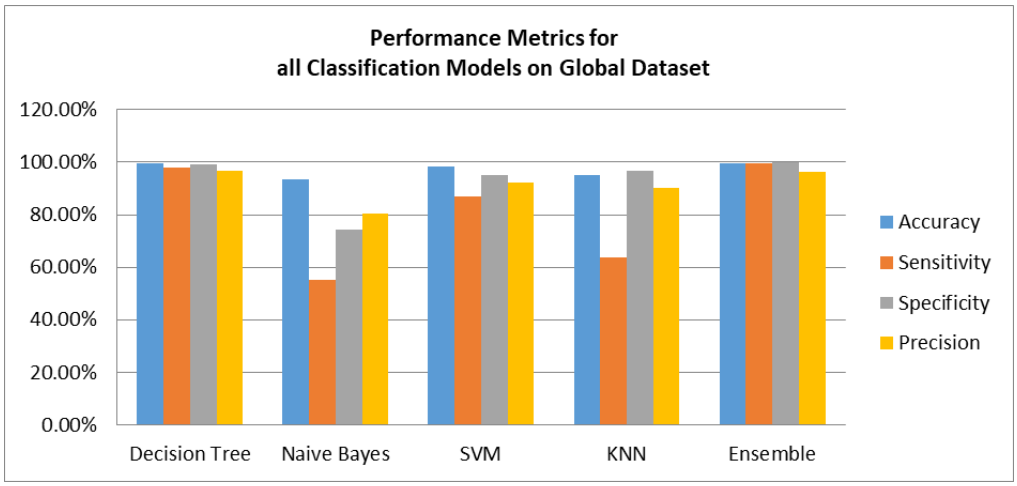



**Figure 4.2: Chart of Comparison between performance metrics for all Classification Models**

In the summary, we conclude that the highest model in terms of overall accuracy is the Ensemble model 99.6%, and achieved high results compared to the rest of the models. In the figure (4.3) are the whole lines with blue color representing the AUC and ROC curves for Ensemble model results [best accuracy], which represents sensitivity for three classes (class 1 is normal, class 2 is hyper, and class 3 is hypo); the sensitivity was (99.40%, 99.70%, and 99.58%). The specificity was (99.97%, 99.80%, and 99.63%). And the precision was (92.70%, 96.83%, and 99.95%) for classes 1,2 and 3 respectively. These results are good results when the points were in the upper-left corner; here area under the curve occupies 100% of these curves for three classes, and the current classifier equals (0.00,0.99), (0.00,0.99), and (0.01,1.00) for class 1,2 and 3 respectively.

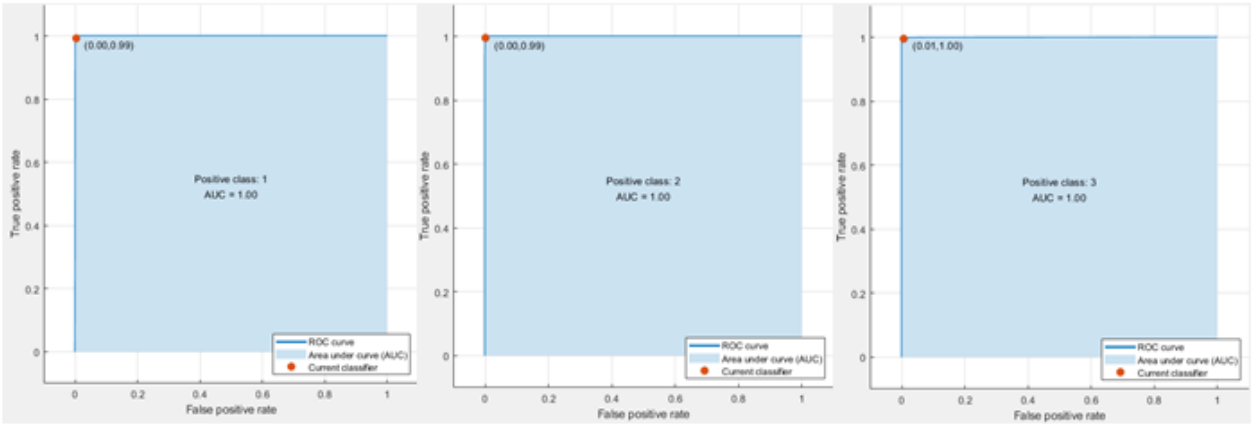



**Figure 4.3: AUC & ROC curves for global dataset (Class1, 2, and 3) by Ensemble model**

*Second Experiment:* Classification Global dataset using MLPNN model; the NNs are one of the most powerful methods for classifying and recognizing data patterns [45]. We build an NNs with two-layer feed-forward, with sigmoid and softmax output neurons. Many tests were done to get the best results. For each test of 10, 20, 30, 40, and 50 neurons were used in the hidden layer. We divide the data like the following: 70% for training 5040 records, 15% for testing 1080 records, and 15% for validation 1080 records. Figure (4.4) shows the neural network structure used in the model to classify the dataset.

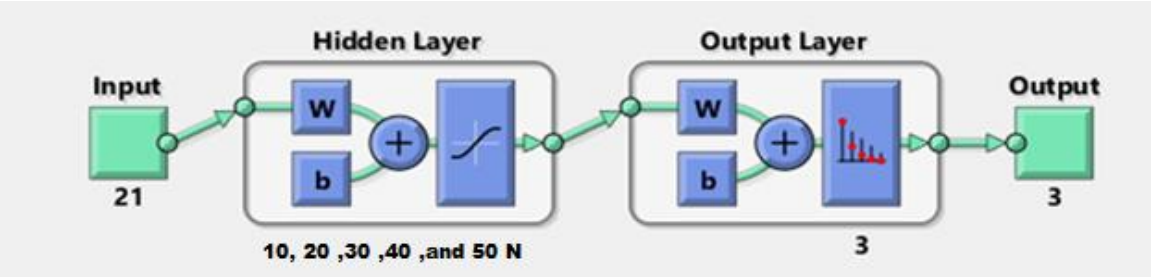



**Figure 4.4: General Neural Network structure for classification Global dataset**

We used scaled conjugate gradient back propagation for training the data. After experimenting with training the model, the results that we obtained with changing the number of neurons in the hidden layer were according to table (4.2).

**Table 4.2:MLPNNs Classification Results on Global Dataset**

| Number of neurons in the hidden layer | Train Accuracy | Test Accuracy |
|---|---|---|
| 10 | 94.30% | 93.20% |
| 20 | 94.60% | 93.10% |
| 30 | 94.50% | 93.90% |
| 40 | 94.60% | 94.10% |
| **50** | **95.70%** | **95.60%** |

Table (4.2) represents the MLPNNs model results. The model achieved the best performance with N=50 where the performance parameters of the 50 Neurons in the hidden layer represented as 95.70%, 95.60%, for training accuracy, test accuracy respectively. Figure (4.5) illustrates the train and test accuracy for the MLPNNs model applying to the global dataset with changing the number of neurons in the hidden layer. We also note that the higher the number of neurons, the more accurate the classification; and this seems to be due to the large data size of 7200 records.

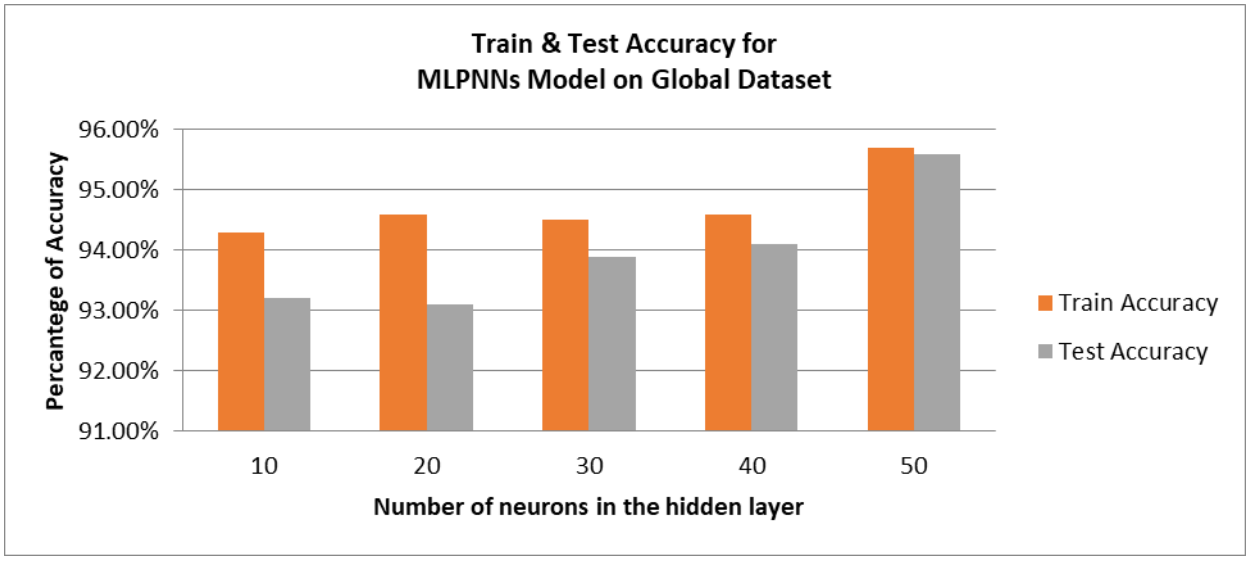



**Figure 4.5: Chart comparing different numbers of neurons for MLPNNs classification model in terms of training and testing accuracy**

Figure (4.6) shows the increase of test accuracy for the classification model when increasing the number of neurons in the hidden layer.

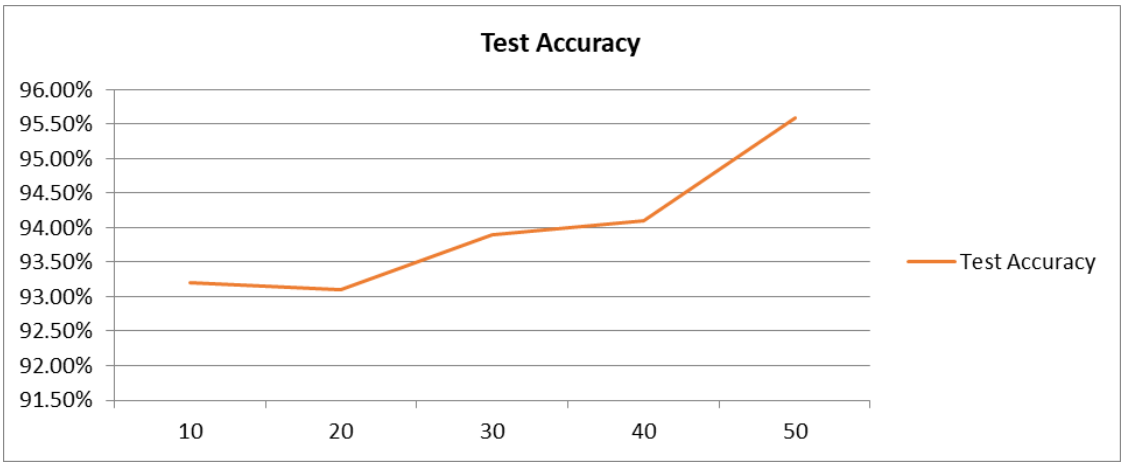



**Figure 4.6: Chart showing the test accuracy increasing with increasing the numbers of neurons for MLPNN classification model.**

In figure (4.7) shows the confusion matrices and accuracy percentages training, validation, and testing, the percentages were 95.7%, 95.0%, and 95.6% respectively. These figures represent the accuracy (94.60% overall accuracy) was obtained from the global dataset when using the MLPNNs model for classification when N=50 (the best result from experiments). Figure (4.8) shows the receiver operating characteristic (ROC) curves for the global dataset. The three lines with blue, green, and red colors for all figures represent the ROC curve for training, validations, and testing datasets. The ROC curve represents sensitivity versus (1- Specificity) for the global dataset. The sensitivity is 88.20%, the Specificity is 99.4%, and the precision is 76.5%, these results for class 1 in All ROC. The sensitivity is 77.30%, the Specificity is 96.5%, and the precision is 32.3%, these results for class 2 in All ROC. The sensitivity is 96.20%, the Specificity is 90.3%, and the precision is 99.6%, these results for class 3 in All ROC.

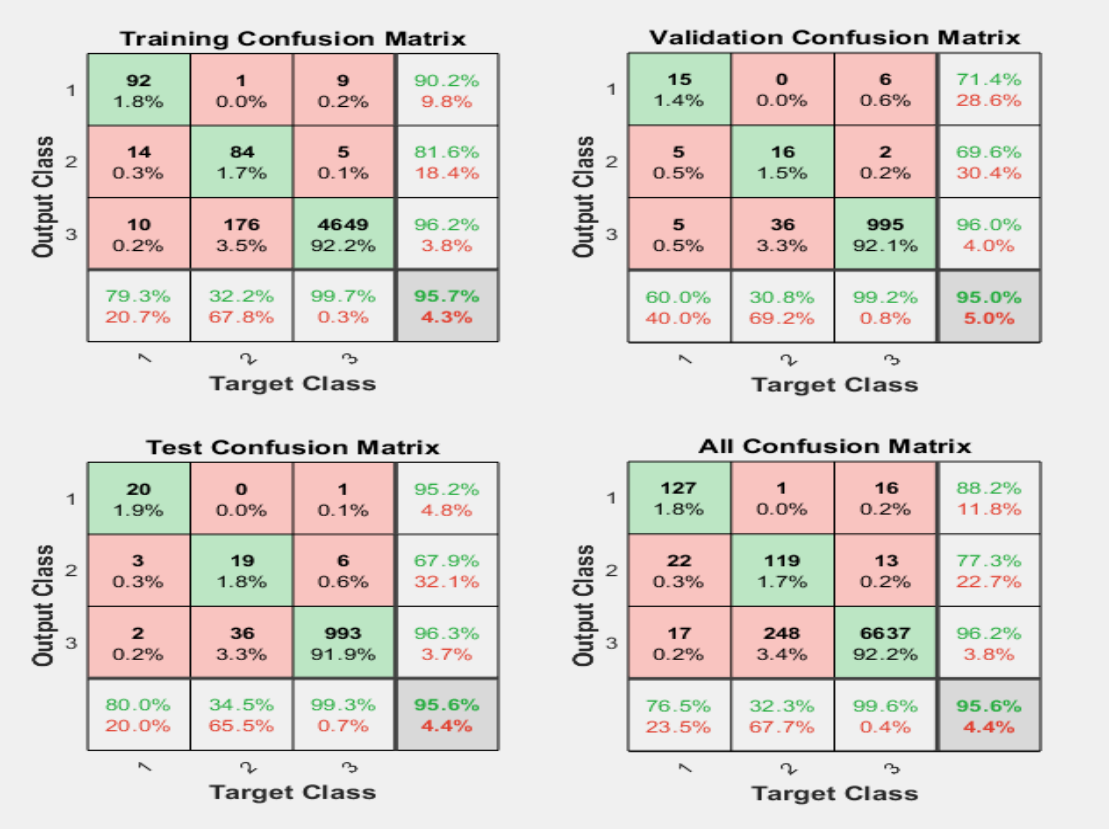



**Figure 4.7: Confusion Matrices for classifying Global dataset using MLPNNs when N=50.**

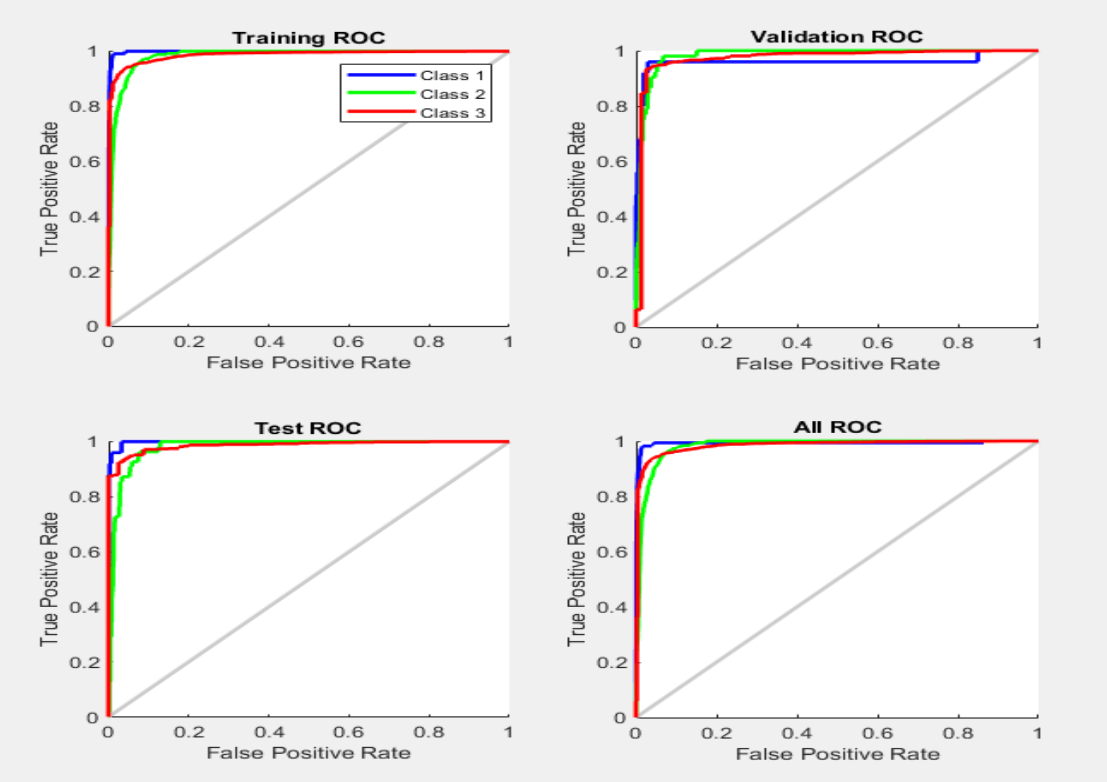



**Figure 4.8: ROC curves for classifying Global dataset using MLPNNs when N=50**

And they are very well in network performance and good when the points are in the upper-left corner. Table (4.3) & figure (4.9) conclude the result for all classification models on global dataset.

**Table 4.3: Table summarizing of the overall accuracy of all classification models for the global dataset**

| Model Type | Overall Accuracy |
|---|---|
| Decision Tree | 99.50% |
| Naive Bayes | 93.30% |
| SVM | 98.20% |
| KNN | 95.10% |
| Ensemble | 99.60% |
| MLPNN, N=50 | 95.60% |

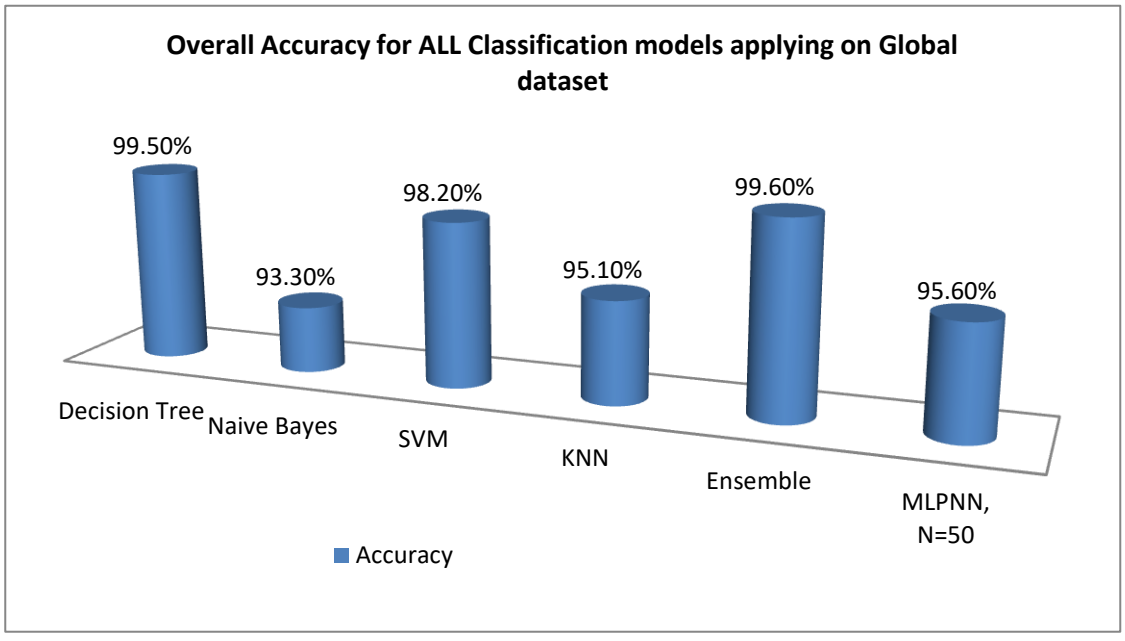



**Figure 4.9: Chart of summary of the overall accuracy of all classification models for the global dataset**

It was concluded that Ensemble is the best model type for classification of this dataset due to higher accuracy of 99.6% and it is close to the Decision Tree accuracy result of 99.5%.

### 4.3.2 Classification Results for Local Datasets

*First Experiment:* In this experiment we applied five classification algorithms they are as follows: decision tree (DT), Naïve Bayes (NB), support vector machine (SVM), K-nearest neighbor (KNN), and Ensemble. In this experiment, a K-fold cross-validation methodology was used to divide the dataset into a test set and a training set. In the recent experiments, we used 5-fold cross-validation from 3775 records each fold 755 records are obtained, 1 fold for testing randomly, and 4 folds for training. The experiments were done on the local dataset through two stages: The first stage was the removal of the variable (city) from the variables in the dataset, where the classification was applied to all records from all cities (whole data (3775) records). The second stage was to aggregate all the records of each city separately without the variable (city) and apply the classification to the records coming from one city. In other words, make each city its own dataset and apply classification models to it, of course removal (city) variable. All experiments into each stage we were used 5-fold cross validation.

The table 4.4 shows and summarizes the results for all local datasets for two stages.

**Table 4.4: Classification results for different model on each city local dataset & all city local dataset without (city) variable**

| City | Model Type | Overall Accuracy | Sensitivity | Specificity | Precision |
|---|---|---|---|---|---|
| All-City Dataset | Decision Tree | 90.60% | 89.58% | 95.14% | 90.75% |
| | Naive Bayes | 89.70% | 89.44% | 94.86% | 88.00% |
| | SVM | 90.40% | 90.00% | 95.18% | 89.40% |
| | KNN | 86.70% | 86.60% | 96.50% | 88.22% |
| | **Ensemble** | **91.10%** | **90.20%** | **95.39%** | **89.7%** |
| Nablus Dataset | **Decision Tree** | **88.50%** | **88.38%** | **94.00%** | **87.61%** |
| | Naive Bayes | 85.50% | 86.30% | 92.44% | 85.70% |
| | SVM | 87.00% | 87.40% | 93.37% | 88.06% |
| | KNN | 80.00% | 80.54% | 89.80% | 81.81% |
| | Ensemble | 85.80% | 86.18% | 92.77% | 84.32% |
| Qalqiliah Datast | **Decision Tree** | **91.90%** | **89.52%** | **95.25%** | **91.39%** |
| | Naive Bayes | 91.80% | 90.68% | 95.65% | 90.74% |
| | SVM | 90.20% | 89.14% | 94.97% | 88.59% |
| | KNN | 85.50% | 83.76% | 94.38% | 83.52% |
| | Ensemble | 90.70% | 88.78% | 94.78% | 89.72% |
| Salfeit Data set | **Decision Tree** | **89.90%** | **89.72%** | **94.92%** | **90.33%** |
| | Naive Bayes | 88.10% | 88.33% | 94.05% | 88.37% |
| | SVM | 87.20% | 87.35% | 93.60% | 87.33% |
| | KNN | 83.60% | 83.79% | 95.99% | 90.20% |
| | Ensemble | 89.30% | 89.41% | 94.60% | 89.52% |
| **Ramallah Dataset** | **Decision Tree** | **92.40%** | **91.05%** | **96.18%** | **91.97%** |
| | Naive Bayes | 92.00% | 90.68% | 96.32% | 91.32% |
| | SVM | 91.60% | 90.27% | 96.10% | 90.93% |
| | KNN | 87.80% | 87.00% | 94.30% | 87.84% |
| | Ensemble | 92.30% | 91.67% | 96.48% | 91.58% |

It was concluded that Ensemble is the best model type for classification this dataset in case all cities (whole local dataset record) due to higher accuracy 91.10%, and it is close to Decision Tree accuracy result 90.6%; Due to the procedures of the Ensemble model, which is the use of more than one model, and the data is divided into sub data and

passing to more than one model, and then the highest accuracy is voted on and approved as the total accuracy of the model.

We can also conclude that the Decision Tree (DT) model was the best model for the classification of the data on each city alone. Here, the preference of the decision tree model over the Ensemble model appeared due to the small size of the data on each city dataset, despite the presence of convergence between them.

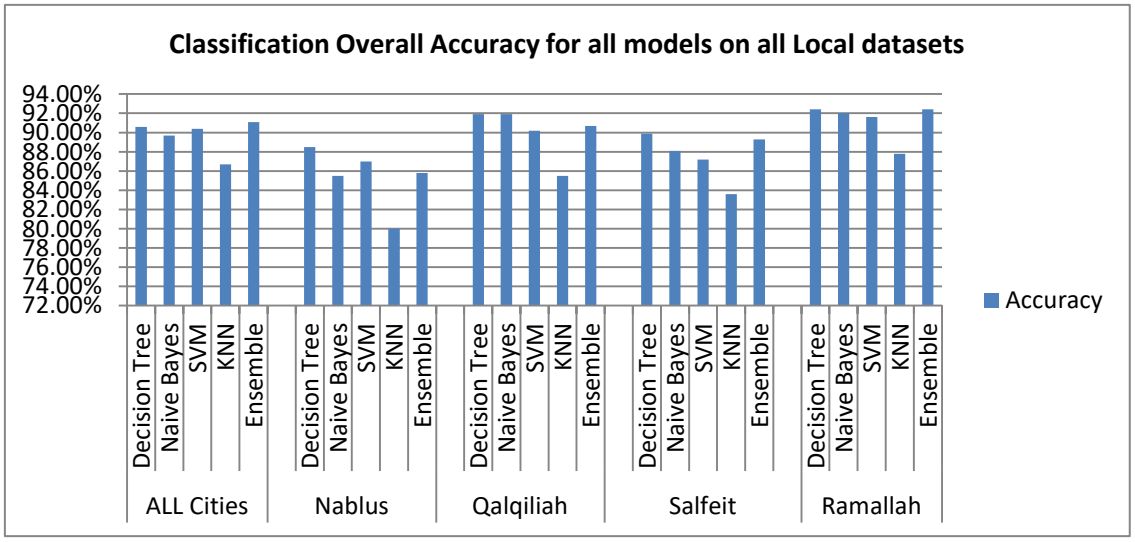



**Figure 4.10: Chart of summary for Classification Accuracy of all models on five Local Datasets**

**Results From First Stage:**

After the end of the experiments from the first stage (practical experiments on the all cities dataset (local dataset). Figure (4.11) illustrates the Comparison between performance metrics for all classification models on the Local dataset (all cities dataset).

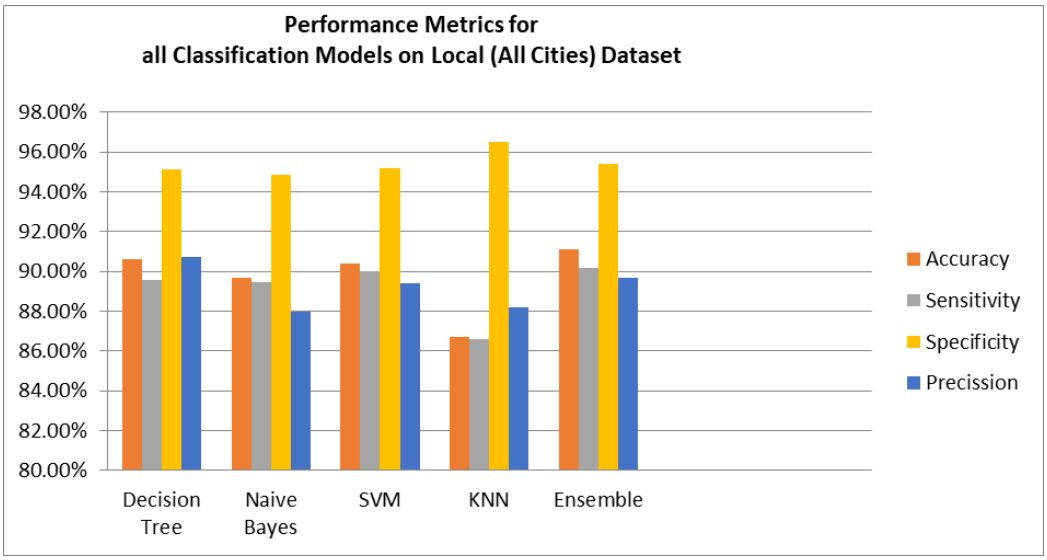



**Figure 4.11: Chart of Comparison between performance metrics for all Classification models on Local dataset**

we obtained the highest model in terms of overall accuracy is the Ensemble model 91.10%, and achieved high results compared to the rest of the models.

In the figure (4.12), are the whole lines with blue color representing the ROC curves, which represents sensitivity for three classes; the sensitivity was 94.97%, 97.00%, and 78.64% for classes 1,2 and 3 respectively. The specificity was 97.10%, 93.00%, and 96.07% for class 1,2 and 3 respectively. The precision was 93.51%, 97.00%, and 78.64% for class 1,2 and 3 respectively. These results are good results when the points were in the upper-left corner; here area under the curve occupies 99% of these curves for class 1 and class 2, 97% for class 3, and the current classifier equals (0.03,0.95), (0.07,0.97), and (0.04,0.79) for class 1,2 and 3 respectively.

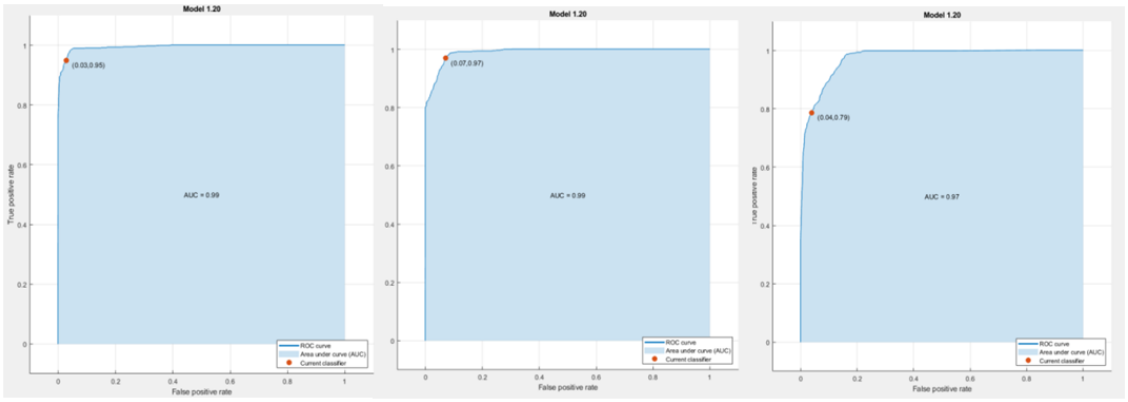


**Figure 4.12: AUC & ROC curves for Local dataset (all cities dataset) (Class 1, 2, and 3) by Ensemble model**

**Results From Second Stage:**

After the end of the experiments from the Second stage (practical experiments using the five models (DT, NB, SVM, KNN, and Ensemble) on each city dataset (Nablus, Qalqiliah, Salfeit, and Ramaalah datasets), The results in each city's data set were as follows:

1. Nablus dataset: Figure (4.13) illustrates the Comparison between performance metrics for all classification models on the (Nablus) dataset.

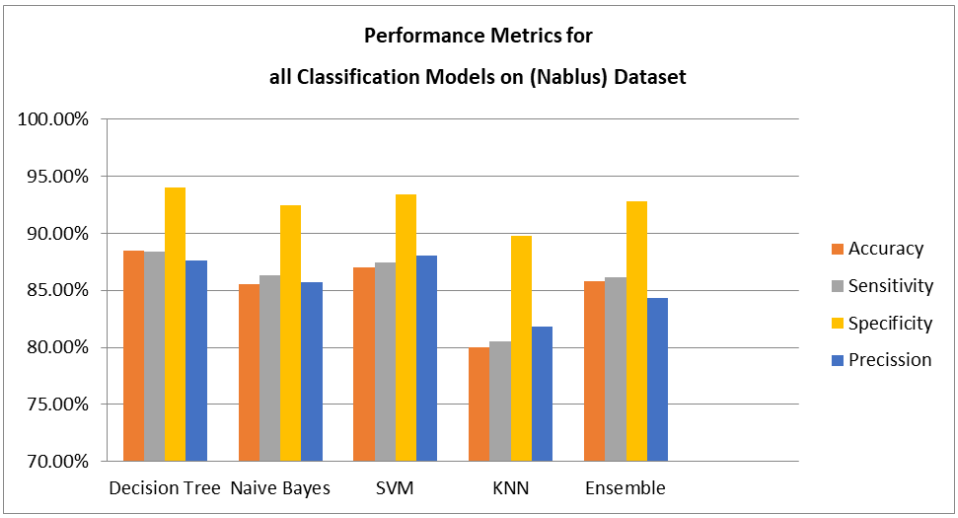



**Figure 4.13: Chart of Comparison between performance metrics for all Classification Models on (Nablus) dataset**

We conclude that the highest model in terms of overall accuracy is the Decision Tree (DT) model 88.5% and achieved high results compared to the rest of the models.

In the figure (4.14), are the whole lines with blue color representing the ROC curves, which represents sensitivity for three classes; the sensitivity was 89.86%, 94.93%, and 80.35% for classes 1,2 and 3 respectively. The specificity was 98.83%, 90.24%, and 92.87%. The precision was 96.53%, 85.96%, and 80.35% for class 1,2 and 3 respectively. These results are good results when the points were in the upper-left

corner; here area under the curve occupies 97%, 93%, and 88% of these curves for class 1, class2, and class 3 respectively, and the current classifier equals (0.01,0.90), (0.10,0.95), and (0.07,0.80) for class 1,2 and 3 respectively.

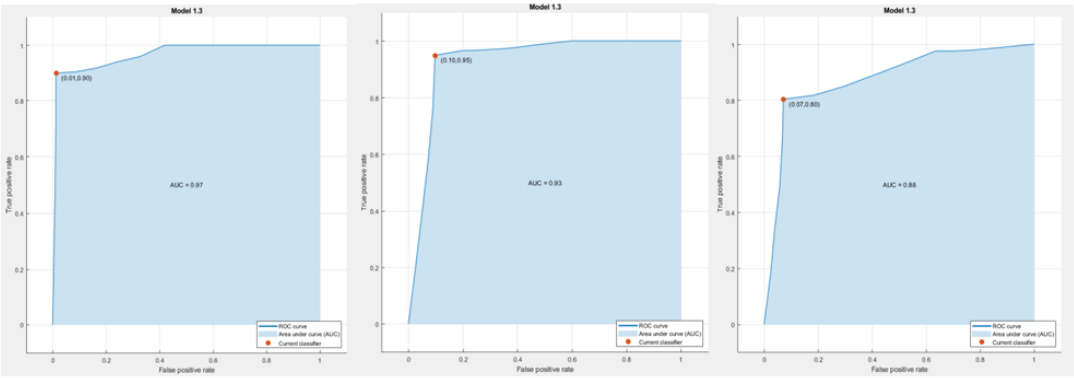


**Figure 4.14: AUC & ROC curves for (Nablus) dataset (Class 1, 2 and 3) by DT model**

2. Qalqiliah dataset: Figure (4.15) illustrates the Comparison between performance metrics for all classification models on the (Qalqiliah) dataset

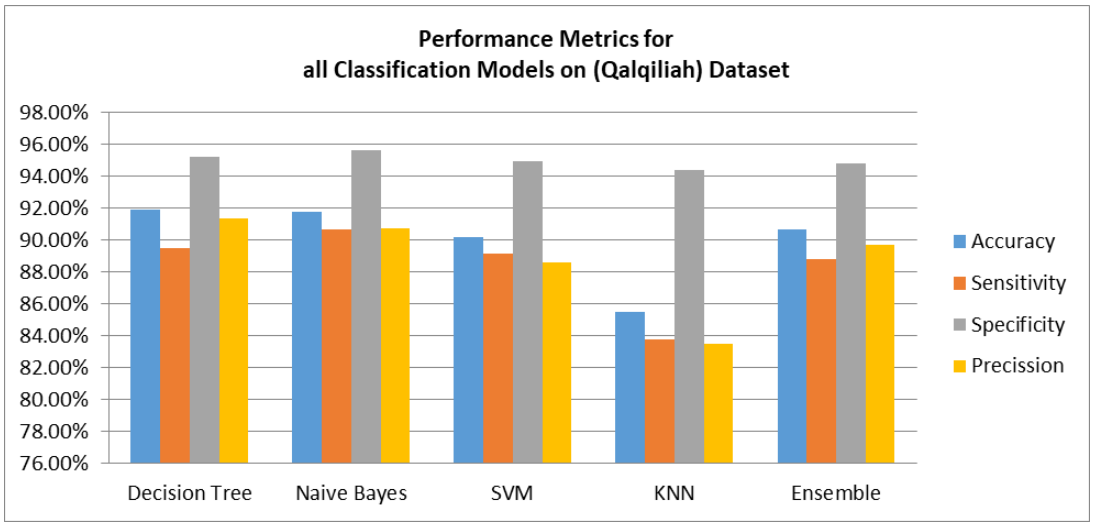



**Figure 4.15: Chart of Comparison between performance metrics for all Classification Models on (Qalqiliah) dataset**

In the summary, we conclude that the highest model in terms of overall accuracy is the Decision Tree (DT) model 91.9%, and achieved high results compared to the rest of the models. In the figure (4.16), are the whole lines with blue color represents the ROC curves, which represents sensitivity for three classes; the sensitivity was 94.24%, 97.36%, and 76.95% for class 1,2 and 3 respectively. The specificity was 98.49%,

90.87%, and 96.47%. The precision was 94.63%, 92.77%, and 86.78% for class 1,2 and 3 respectively. These results are good results when the points were in the upper-left corner; here area under the curve occupies 99%, 95%, and 90% of these curves for class 1, class2, and class 3 respectively, and the current classifier equals (0.02,0.94), (0.09,0.97), and (0.04,0.77) for class 1,2 and 3 respectively

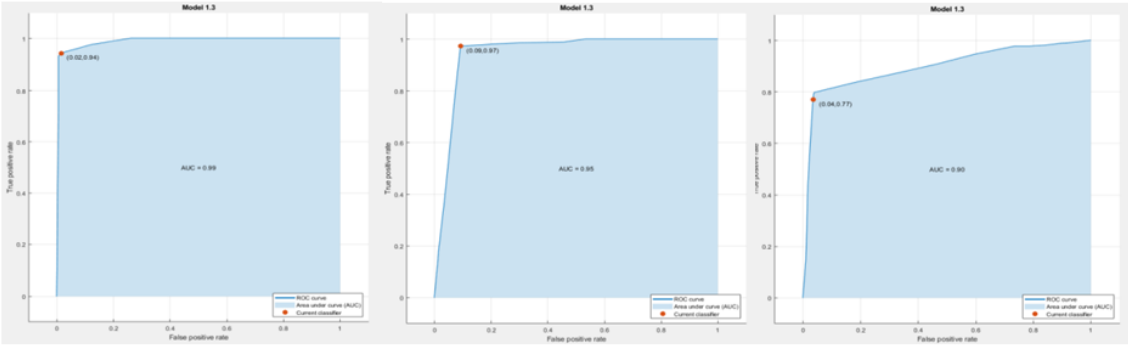


**Figure 4.16: AUC & ROC curves for (Qalqiliah) dataset (Class 1, 2, and 3) by DT model**

3. Salfeit dataset: Figure 4.17 illustrates the Comparison between performance metrics for all classification models on the (Salfeit) dataset.

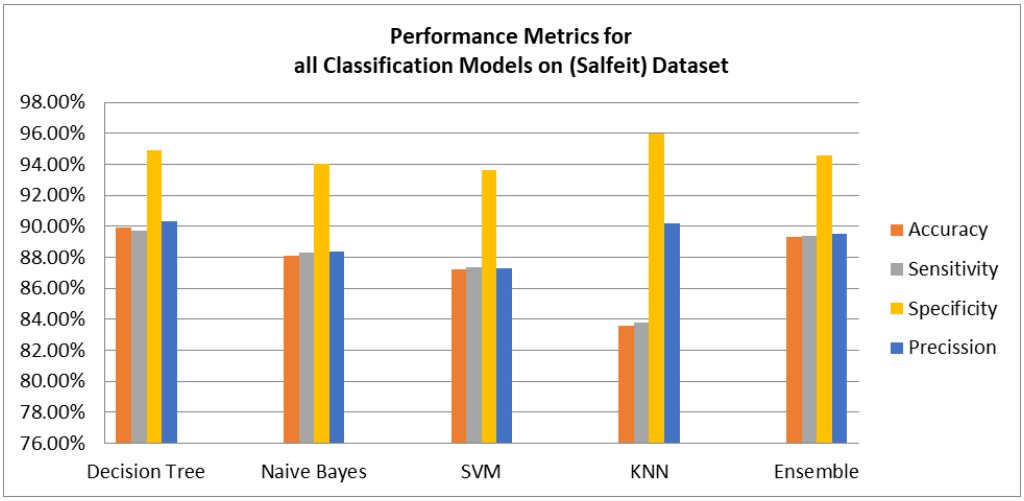



**Figure 4.17: Chart of Comparison between performance metrics for all Classification Models on (Salfeit) dataset**

We conclude that the highest model in terms of overall accuracy is the Decision Tree (DT) model 89.9%, and achieved high results compared to the rest of the models. In the figure (4.18), are the whole lines with blue color represents the ROC curves, which

represents sensitivity for three classes; the sensitivity was 93.47%, 98.02%, and 77.66% for class 1,2 and 3 respectively. The specificity was 97.40%, 91.12%, and 96.23%. The precision was 93.85%, 86.55%, and 90.60% for class 1,2 and 3 respectively. These results are good results when the points were in the upper-left corner; here area under the curve occupies 96%, 94%, and 90% of these curves for class 1, class2, and class 3 respectively, and the current classifier equals (0.03,0.93), (0.09,0.98), and (0.04,0.78) for class 1,2 and 3 respectively.

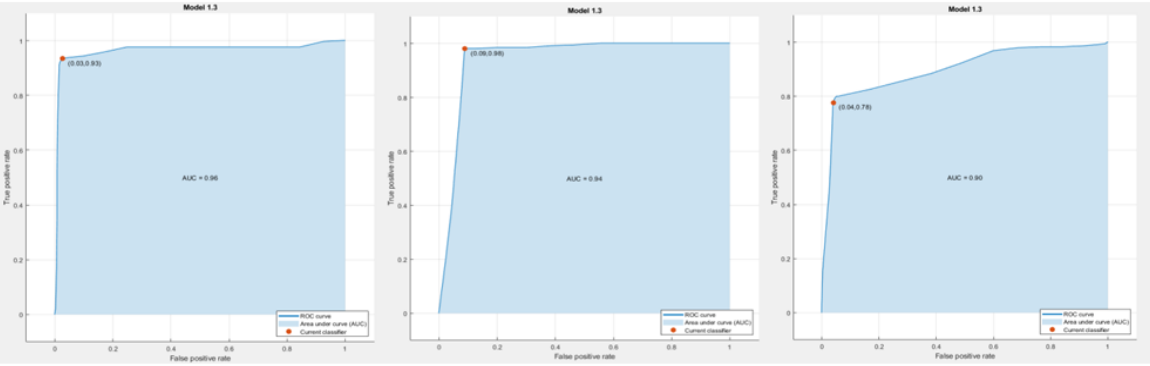


**Figure 4.18: AUC & ROC curves for (Salfeit) dataset (Class 1, 2, and 3) by DT model**

4. Ramallh dataset: Figure (4.19) illustrates the Comparison between performance metrics for all classification models on (Ramallah) dataset.

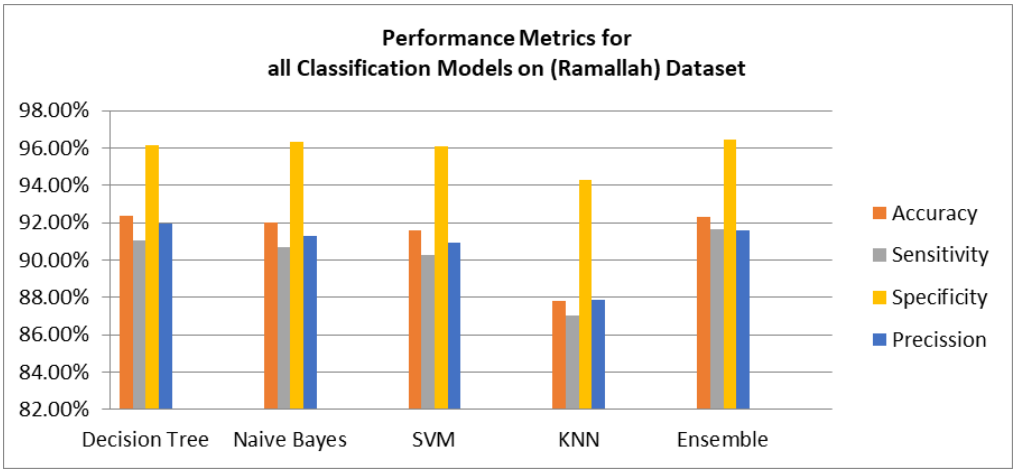



**Figure 4.19: Chart of Comparison between performance metrics for all Classification Models on (Ramallah) dataset**

In the summary, we conclude that the highest model in terms of overall accuracy is the Decision Tree (DT) 92.4%, and achieved high results compared to the rest of the models. In the figure 4.20 the whole lines with blue color representing the ROC curves, which represents sensitivity for three classes; the sensitivity was 98.66%, 95.06%, and 79.42% for classes 1,2 and 3 respectively. The specificity was 95.87%, 95.45%, and 97.21%. The precision was 94.86%, 89.75%, and 91.29% for class 1,2 and 3 respectively. These results are good results when the points were in the upper-left corner; here area under the curve occupies 99%, 97%, and 94% of these curves for class 1, class2, and class 3 respectively, and the current classifier equals (0.04,0.99), (0.06,0.95), and (0.03,0.79) for class 1,2 and 3 respectively.

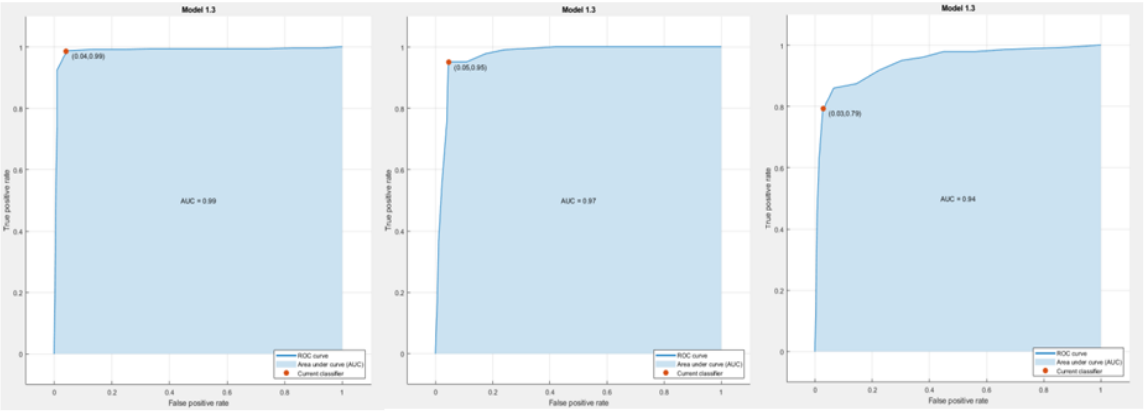



**Figure 4.20: AUC & ROC curves for (Ramallah) dataset (Class 1, 2, and 3) by DT model**

*Second Experiment:* Classification Local datasets using MLPNN model; we build a neural network model with two-layer feed-forward, with sigmoid and softmax output neurons. Many tests were done to get the best results. For each test of 5, 10, 15, and 20 neurons were used in the hidden layer on each dataset. We use the same strategy that we followed previously in terms of dividing the work in the local data into two stages. We divide the practical experiment on the whole local dataset into five datasets:

1. All records from all cities (whole local dataset) 3775 records without (city) variable.
2. Nablus city records (Nablus Dataset) 818 records without (city) variable.
3. Qalqiliah city records (Qalqiliah Dataset) 1105 records without (city) variable.
4. Salfeit city records (Salfeit Dataset) 822 records without (city) variable.
5. Ramallah city records (Ramallah Dataset) 1030 records without (city) variable.

We divide the records as the following for the five above datasets: 70% for training, 15% for testing, and 15% for validation. Figure (4.21) shows the neural network structure used in five local datasets to classify the dataset by using MLPNN:

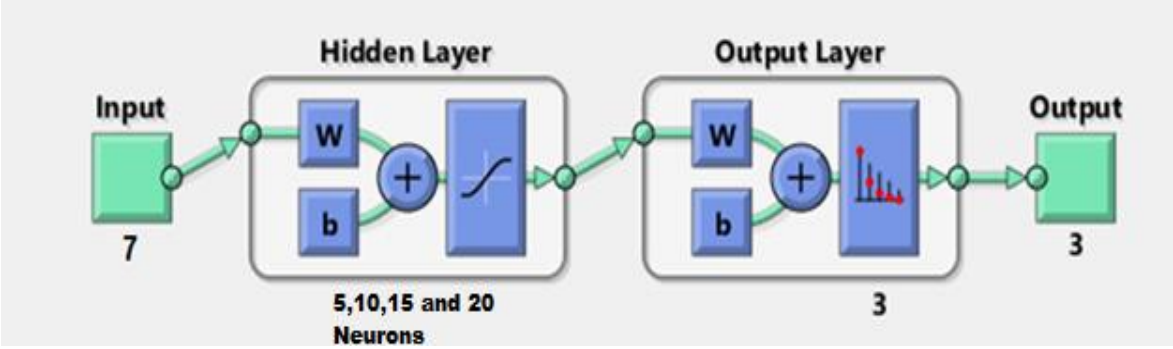



**Figure 4.21: General Neural Network structure for classification Local datasets**

We used scaled conjugate gradient backpropagation for training the data on each dataset in local datasets. After experimenting with training the model, the results that we obtained with changing the number of neurons in the hidden layer were according to each experiment on each dataset from local datasets on table (4.5). The table shows the training accuracy and test accuracy for each city dataset from the local dataset, when changing the number of neurons in the hidden layer within the neural network. We have five datasets, each data set has a specific record number. Each dataset had four different experiments, each experiment had a different number of neurons, and the training accuracy and testing accuracy were recorded for each experiment.

**Table 4.5: MLPNNs Classification Results on Local Datasets**

| City | Number of neurons | Train Accuracy | Test Accuracy |
|---|---|---|---|
| All Cities (3775) Records | 5 | 86.7% | 86.0% |
| | 10 | 88.4% | 88.2% |
| | 15 | 89.3% | 89.0% |
| | **20** | **89.7%** | **89.2%** |
| Nablus (818) Records | **5** | **86.7%** | **82.9%** |
| | 10 | 85.8% | 79.6% |
| | 15 | 85.1% | 79.4% |
| | 20 | 84.3% | 78.2% |
| **Qalqiliah (1105) Records** | 5 | 89.8% | 85.5% |
| | **10** | **90.9%** | **89.2%** |
| | 15 | 90.9% | 87.9% |
| | 20 | 90.8% | 87.3% |
| Salfeit (822) Records | **5** | **88.5%** | **81.3%** |
| | 10 | 87.9% | 81.2% |
| | 15 | 87.6% | 80.2% |
| | 20 | 86.4% | 79.1% |
| Ramallah (1030) Records | 5 | 90.4% | 92.3% |
| | **10** | **91.3%** | **86.5%** |
| | 15 | 91.1% | 85.9% |
| | 20 | 90.1% | 85.1% |

The best accuracy we got from local datasets classification is the dataset for the city of Qalqiliah (1105) records and it was 90.9% using 10 neurons in the hidden layer. We also notice from the table that the classification based on MLPNN, which the larger the data size, the greater the number of neurons in the hidden layer gives better accuracy. For example, the highest classification accuracy was obtained in the dataset (all cities) when we used 20 neurons and it was the size of the data (3775) records, while in the dataset of the city of Nablus (818) records had the best classification accuracy when we used 5 neurons, and this conclusion can be observed on the rest of the datasets of cities.

**Results from First stage:**

After the end of the experiments from the first stage, it was the classification (All Cities) dataset using Multi-Layer Perceptron Neural Networks (MLPNNs) Model. The results had obtained from these figures below figure (4.22) shows the confusion matrices and ROC curves. The overall accuracy was 89.7%, and accuracy percentages for training, validation, and testing were 89.7%,90.3%, and 89.2% respectively. These figures represent the accuracy was obtained from local datasets (All Cities) when using the MLPNNs model for classification when a number of neurons in the hidden layer = 20. In the figure (4.22) also shows the receiver operating characteristic (ROC) curves for the (All Cities) dataset. The three lines with red, green, and blue colors for all figures represent the ROC curve for training, validations, and testing datasets.

The sensitivity is 91.5%, Specificity is 98.7%, and the precision is 97.1%, these results for class 1 in All ROC. The sensitivity is 92.3%, Specificity is 94.1%, and the precision is 91.2%, these results for class 2 in All ROC. The sensitivity is 83.8%, Specificity is 91.9%, and the precision is 79.8%, these results for class 3 in All ROC.

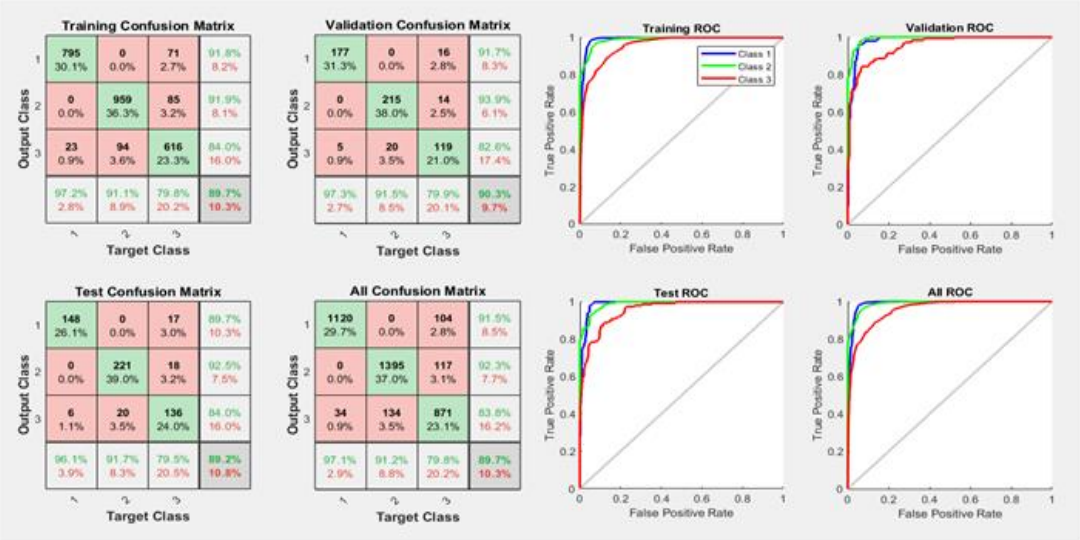



**Figure 4.22: Confusion Matrices and ROC curves for classifying (All Cities) dataset using MLPNNs when N=20.**

**Results From Second Stage:**

After the end of the experiments from the second stage, classification each city dataset (Nablus, Qalqiliah, Salfeit, and Ramaalah datasets)), using Multi-Layer Perceptron Neural Networks (MLPNNs) Model. The results (Over all accuracy) in each city dataset were as follows, according the table (4.6)

**Table 4.6: Overall accuracy for MLPNNs experiments on Local Datasets**

| Dataset | The number of neurons that gives the best result | Overall Accuracy |
|---|---|---|
| All Cities | 20 | 89.7% |
| Nablus | 5 | 86.1% |
| **Qalqiliah** | **10** | **90.5%** |
| Salfeit | 5 | 87.5% |
| Ramallah | 10 | 89.9% |

We conclude that the data from the city of Qalqiliah (Qalqiliah dataset) obtained the highest accuracy in classification by MLPNNs 90.5%. When using N=10.

Figure (4.23) shows and summarize the highest overall accuracy value (achieved when changing the value of N) of the MLPNNs classification model for each dataset from the local datasets.

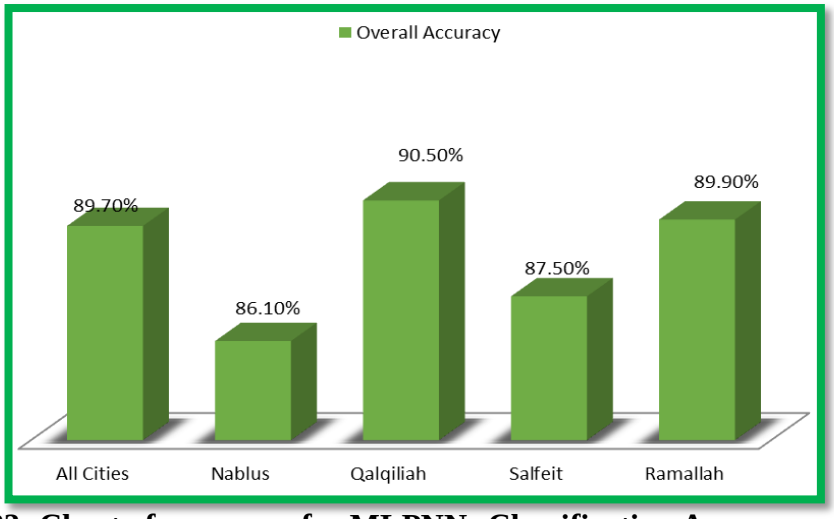



**Figure 4.23: Chart of summary for MLPNNs Classification Accuracy on five Dataset from local dataset**

At the end of the all experiments on all Local datasets, table (4.7) shows the Overall accuracy for all classification models were had been used on five local datasets.

**Table 4.7: Table summarizing of the overall accuracy of all classification models for the Local datasets**

| City dataset | Classification Model Type | Overall Accuracy |
|---|---|---|
| All Cities Dataset | Decision Tree | 90.60% |
| | Naive Bayes | 89.70% |
| | SVM | 90.40% |
| | KNN | 86.70% |
| | **Ensemble** | **91.10%** |
| | MLPNNs , N=20 | 89.10% |
| Nablus Dataset | **Decision Tree** | **88.50%** |
| | Naïve Bayes | 85.50% |
| | SVM | 87.00% |
| | KNN | 80.00% |
| | Ensemble | 85.80% |
| | MLPNNS, N=5 | 86.10% |
| Qalqiliah Dataset | **Decision Tree** | **91.90%** |
| | Naive Bayes | 91.80% |
| | SVM | 90.20% |
| | KNN | 85.50% |
| | Ensemble | 90.70% |
| | MLPNNs, N=10 | 90.50% |
| Salfeit Dataset | **Decision Tree** | **89.90%** |
| | Naïve Bayes | 88.10% |
| | SVM | 87.20% |
| | KNN | 83.60% |
| | Ensemble | 89.30% |
| | MLPNNs, N=5 | 87.50% |
| **Ramallah Dataset** | **Decision Tree** | **92.40%** |
| | Naïve Bayes | 92.00% |
| | SVM | 91.60% |
| | KNN | 87.80% |
| | Ensemble | 92.30% |
| | MLPNNs, N=10 | 89.90% |

Figure (4.24) shows the Overall accuracy of all classification models used in the classification of local datasets.

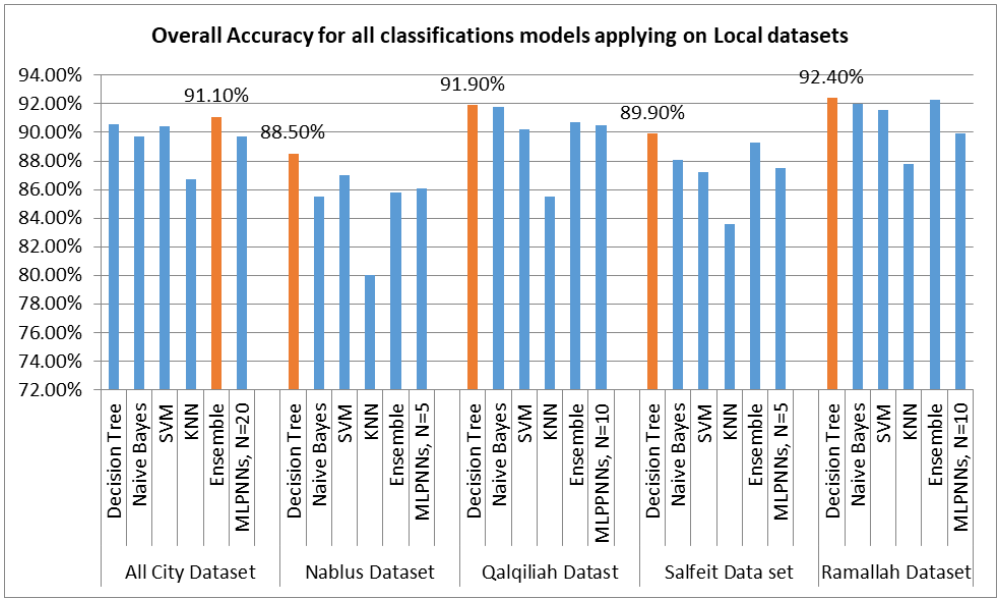



**Figure 4.24: Chart of summary for all Classification models Overall Accuracy applying on five Local Datasets**

We conclude that the Ensemble model gave the highest accuracy in classification the (All cities dataset) 91.10%. Decision tree model record the highest accuracy for classification of the (Nablus dataset), (Qalqiliah dataset), (Salfeit dataset), and (Ramallah dataset) 88.50%, 91.90%, 89.90%, and 92.40% respectively.

The highest overall accuracy value (92.40%) was obtained from the (Ramallah dataset) by applying the decision tree (DT) model.

### 4.3.3 Comparison between Results for Global Data and Local Data

To make a comparison between the results of global data and local datasets (All cities datasets), we must first find the common variables between the two datasets; due to the presence of some differences in the variables. In this experiment, some variables will be

removed from the global dataset to be more involved with the local dataset (All cities dataset).

The new global data set will be according to the first column in table 4.8. We will call it (Custom Global dataset), after excluding the variables that do not exist in the local dataset like (On_thyroxine, Query_on_thyroxine, Sick, Thyroid_surgery, Query_hypothyroid, Query_hyperthyroid, Goitre, Tumor, Hypopituitary, Psych, TT4, and T4U); the reason is that there are no equivalent variables in the local dataset so that the comparison is more reasonable.

The local dataset will be the same as before with the variable (city) was omitted, according to the second column in table (4.8). In short, global and local datasets that are compared in terms of variables will have equivalent variables where the comparison is most useful and meaningful.

**Table 4.8: Custom global dataset variables and local dataset variables (All cities dataset) to make a comparison**

| # | Feature Name in Global dataset | Feature Name in Local dataset | Description | Range of Value | Type of value |
|---|---|---|---|---|---|
| 1. | Age | Age | Age of the patient | (0.01–0.97) in Global<br>(1 – 82) in Local | Continuous variable |
| 2. | Sex | Sex | Sex of the patient | 0 (Male)<br>1 (Female) | Binary variable |
| 3. | Pregnant | Pregnant | positive pregnancy result | 0 (FALSE)<br>1 (TRUE) | Binary variable |
| 4. | On_antithyroid_medication | Treatment | Any Type of treatment | 0 (FALSE)<br>1 (TRUE) | Binary variable |
| 5. | I131_treatment | | | | |
| 6. | Lithium | | | | |
| 7. | TSH | TSH | Thyroid-stimulating hormone | (0.0–0.53) in Global<br>(0.01-52.63) in Local | Continuous variable |
| 8. | T3 | FT3 | Free /Triiodothyronine hormone | (0.0005–0.18) in Global<br>(1.10-14.28) in | Continuous variable |

| | | | | | |
|---|---|---|---|---|---|
| | | | | Local | |
| 9. | FTI | FT4 | Free Thyroxine Index/ Free Thyroxine | (0.002–0.642) in Global (0.15-4.90) in Local | Continuous variable |

❖ **Classification Results:**

In these experiments we apply five classification algorithms they are as follows: decision tree (DT), Naïve Bayes (NB), support vector machine (SVM), K-nearest neighbor (KNN), and Ensemble. The First experiments were done on the custom global dataset. [9 variables * 7200 records]. In the experiments, we have used 5-fold cross-validation. Table (4.9) shows and summarizes the results for the custom global dataset.

**Table 4.9: Classification results for different models on custom global dataset (9 variables)**

| Model Type | Accuracy | Sensitivity | Specificity | precision |
|---|---|---|---|---|
| Decision Tree | 98.30% | 96.56% | 97.72% | 96.00% |
| Naive Bayes | 94.30% | 59.55% | 76.54% | 81.33% |
| SVM | 96.50% | 81.39% | 90.32% | 95.86% |
| KNN | 95.70% | 71.92% | 84.73% | 82.80% |
| **Ensemble** | **98.40%** | **96.34%** | **97.80%** | **94.72%** |

It was concluded that Ensemble is the best model type for classification this dataset in case (9 variable); due to higher accuracy 98.4%, and it is almost the same to Decision Tree accuracy result 98.3%; here 5-fold cross-validation was used, which (it) means that 20% testing data and 80% for training. The second experiment was done on the local dataset (All cities dataset) after the removal of the variable (city) from the variables in the dataset, where the classification was applied to all records from all cities (whole data

(3775) records). [7 variables * 3775 records]. In the experiments, we have used 5-fold cross-validation. Table (4.10) shows and summarizes the results for the local dataset.

**Table 4.10: Classification results for different models on Local dataset (All cities dataset)**

| Model Type | Accuracy | Sensitivity | Specificity | Precision |
|---|---|---|---|---|
| Decision Tree | 90.60% | 89.58% | 95.14% | 90.75% |
| Naive Bayes | 89.70% | 89.44% | 94.86% | 88.00% |
| SVM | 90.40% | 90.00% | 95.18% | 89.40% |
| KNN | 86.70% | 86.60% | 96.50% | 88.22% |
| **Ensemble** | **91.10%** | **90.20%** | **95.39%** | **89.70%** |

It was concluded that Ensemble is the best model type for classification this dataset in case all cities (whole local dataset record) due to higher accuracy 91.1%, and it is close to Decision Tree accuracy result 90.6%; here 5-fold cross-validation was used, which (it) means that 20% testing data and 80% for training. We conclude also that the custom global dataset recorded a higher accuracy than the local dataset when the process of comparison was made between them in all classification models. And the Ensemble model is the best model for the classification of the two datasets. Figure (4.25) shows the comparison between the custom global dataset and the local dataset (all cities) in terms of overall accuracy for each classification model.

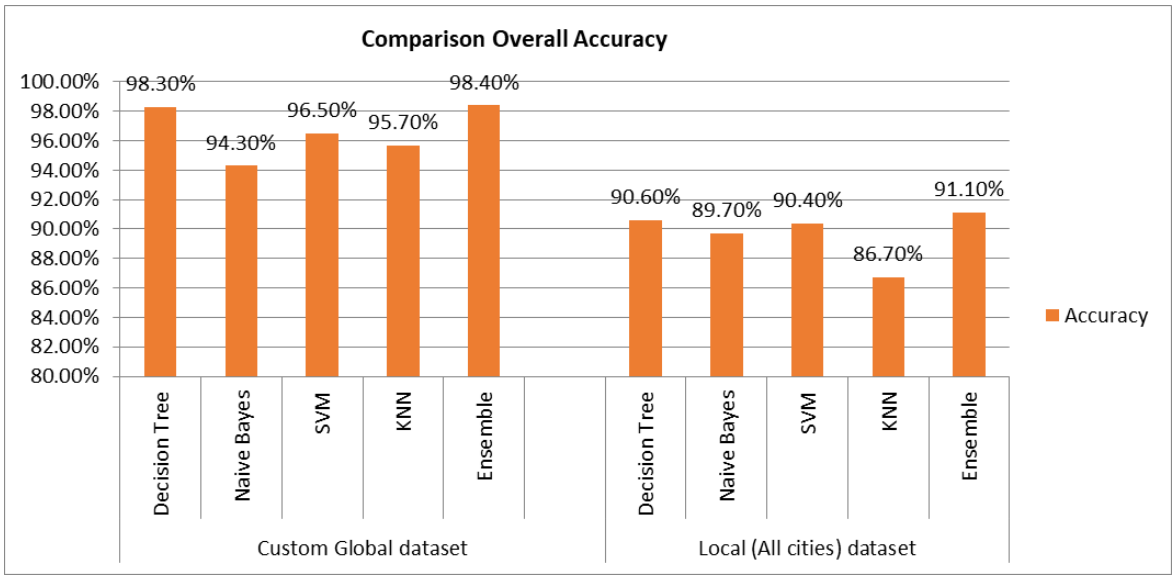



**Figure 4.25: Chart of summary for all models Classification Accuracy on Local & Custom Global datasets**

❖ **Neural Network Results**:

We build a neural network model with two-layer feed-forward, with sigmoid and softmax output neurons. The experiments were done on a custom global dataset & on the Local dataset (All Cities). Many tests were done to get the best results. For each test of 5, 10, 15, and 20 neurons were used in the hidden layer. The records divide as the following for the five above datasets: 70% for training, 15% for testing and 15% for validation. Scaled conjugate gradient backpropagation was used for training the data. After training with changing the number of neurons, the results had been obtained according to the table (4.11).

**Table 4.11: MLPNNs Classification Results on Custom Global Dataset & (All cities) dataset**

| | Custom Global Dataset | | Local Dataset (All Cities) | |
|---|---|---|---|---|
| **Number of neurons in the hidden layer** | **Train Accuracy** | **Test Accuracy** | **Train Accuracy** | **Test Accuracy** |
| 5 | 94.20% | 93.80% | 86.7% | 86.0% |
| 10 | 93.30% | 93.30% | 88.4% | 88.2% |
| 15 | 94.10% | 93.80% | 89.3% | 89.0% |
| **20** | **94.40%** | **94.30%** | **89.7%** | **89.2%** |

Table (4.11) represents the MLPNNs Models results for two datasets to be comparison. The model achieved the best performance with N=20 in both datasets, but in the custom global dataset where the performance parameters of the 20 Neurons in the hidden layer represented as 94.40%, 94.30%, for training accuracy, test accuracy respectively. While in the local dataset (all cities) where the performance parameters of the 20 Neurons in the hidden layer represented as 89.70%, 89.20%, for training accuracy, test accuracy respectively. We conclude that the MLPNNs model achieved a higher accuracy result in the ***custom global dataset***. Figure (4.26) shows the test accuracy percentage for each dataset with changing the number of neurons.

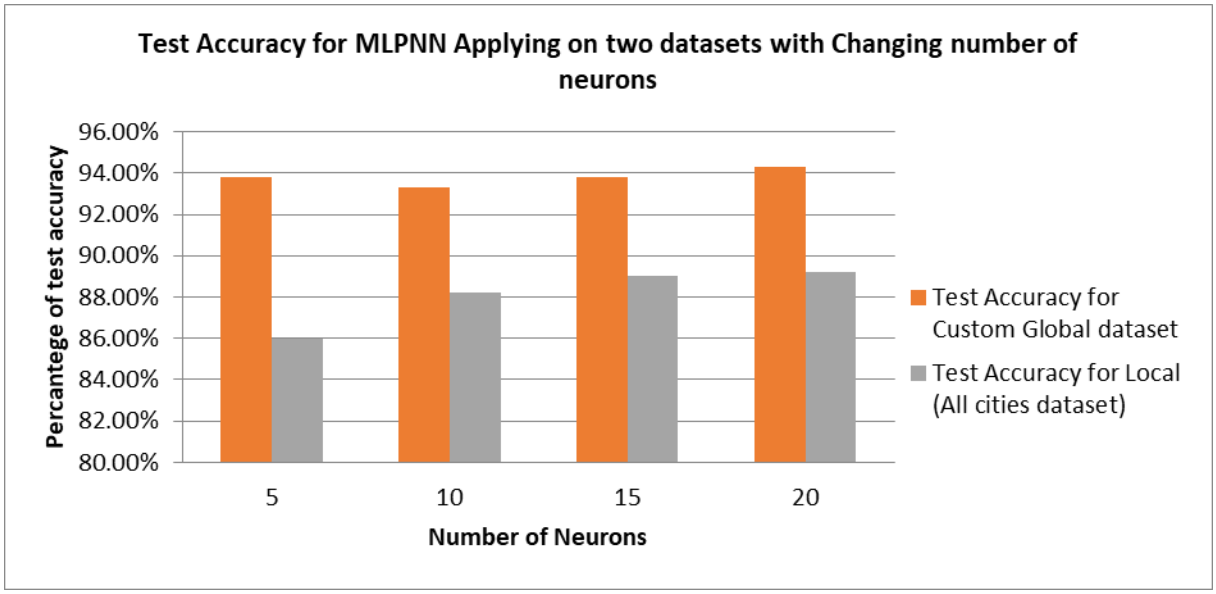



**Figure 4.26: Chart of summary for MLPNNs Classification test accuracy on Local dataset & Custom Global datasets**

## 4.4 Hybrid Model Practical Experiments

In this section, we use hybrid models that combine multi-layer perceptron with optimization algorithms (Genetic Algorithms) to improve the accuracy of the thyroid disease classification system. The GAs-MLPNNs algorithms were used to create class classification on the Custom Global dataset and Local (All cities dataset). The goal of

the experiments performed using the GAs-MLPNNs model was to find the best classification accuracy for the two datasets. To improve the performance of the model we were through many trials and errors, we have used the following parameters and fixed them in the Genetic Algorithms (GAs) model, according to table (4.12).

**Table 4.12: Fixed Genetic Algorithm Parameter**

| | |
|---|---|
| Number of generation | 200 |
| Number of population | 50 |
| Crossover | Uniform 3 point random |
| Mutation | Uniform Constraint dependent |
| Probability of crossover | 1 |
| Probability of Mutation | 0.01 |

Then many tests were done to get the best results through changing the number of neurons in the MLPNNs model. For each test, we used 5, 10, 15, and 20 neurons in the hidden layer. The results were as follows on the two datasets according to table (4.13).

**Table 4.13: Hybrid Model (GAs-MLPNNs) Classification Results on Custom Global Dataset & (All cities) dataset**

| | Custom Global Dataset | | Local Dataset (All Cities) | |
|---|---|---|---|---|
| Number of neurons in the hidden layer | Train Accuracy | Test Accuracy | Train Accuracy | Test Accuracy |
| 5 | 93.00% | 91.00% | 96.00% | 94.00% |
| 10 | 96.30% | **95.00%** | 97.50% | **96.00%** |
| 15 | 95.50% | 94.00% | 96.70% | 95.00% |
| 20 | 92.80% | 90.00% | 94.50% | 92.00% |

Table (4.13) represents the Hybrid Model (GAs-MLPNNs) results for two datasets to be comparison. The model achieved the best performance with N=10 in both datasets, but in the Local dataset (all cities) where the performance parameters of the 10 Neurons in the hidden layer represented as 97.50%, 96.0%, for training accuracy, test accuracy respectively. While in the Custom global dataset where the performance parameters of

the 10 Neurons in the hidden layer represented as 96.30%, 95.00%, for training accuracy, test accuracy respectively.

We conclude that the MLPNNs model achieved a higher accuracy result in the Local dataset (all cities). Figure (4.27) shows the test accuracy percentage for two datasets with changing the number of neurons.

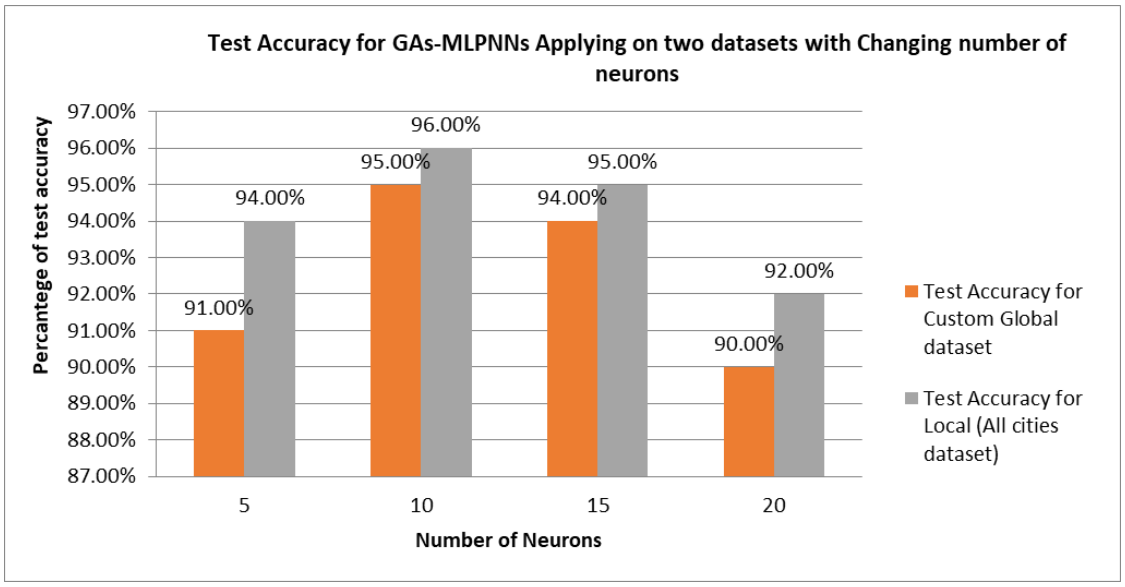



**Figure 4.27: Chart of summary for Hybrid Model (GAs-MLPNNs) Classification test accuracy on two datasets**

## 4.5 Deep Learning Practical Experiments

In this section, we use deep learning to classify ultrasound medical images of the thyroid gland as malignant or benign. I've built a Visual Geometry Group 16 (VGG-16) model, which is a model built based on the architecture of Convolutional Neural Network (CNN) to classify. I used Jupyter Notebook to write Python code using Keras library. We divided the image dataset as the following: 80% for training, 124 images, 20% for validation, 31 images. After training the model, the results came as follows:

Figure (4.28) shows the learning curve for changing in the accuracy of training and the accuracy of the validation with the increase in the number of the epoch (time).

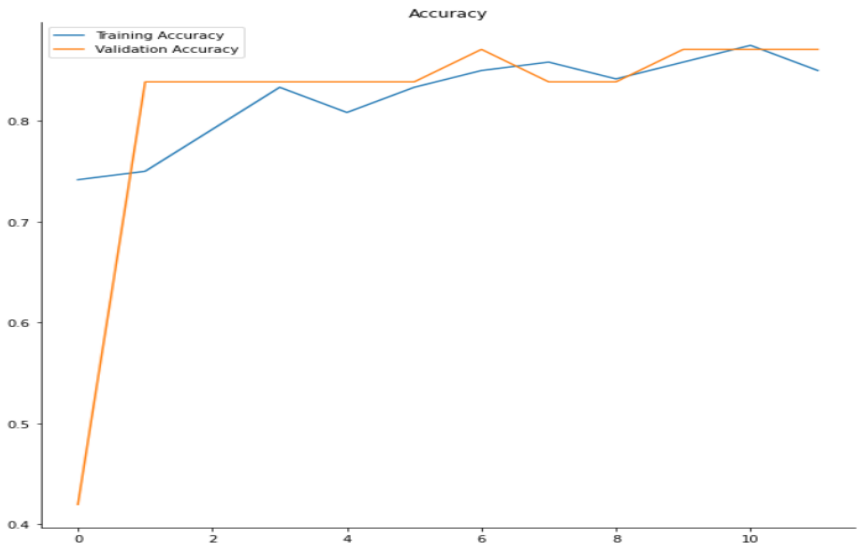



**Figure 4.28: Performance learning curves for training & validation accuracy**

The VGG-16 model performance is growing over epoch, which means the model is improving with experience (its learning). We also see it grows at the beginning, but over time it reaches a plateau, meaning it's not able to learn anymore. Figure (4.29) shows the confusion matrix of the test data for the model, where (B) is an abbreviation for a benign tumor, while (M) is an abbreviation for malignant tumor:

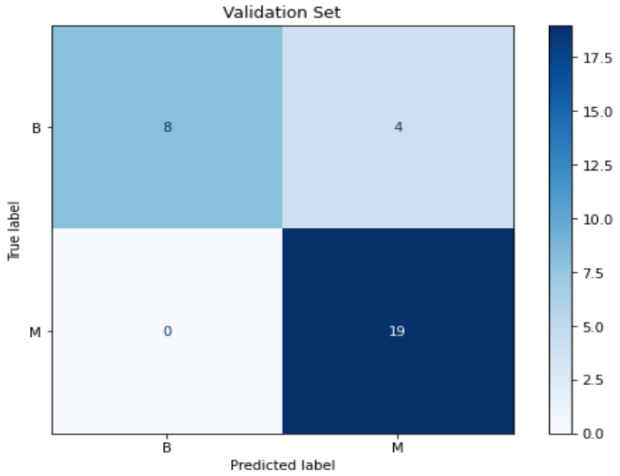



**Figure 4.29: Confusion matrix for vgg-16 model**

From the confusion matrix for the VGG-16 model, we obtained sensitivity, specificity, and precision percentages equal 66.7%, 100%, and 100% respectively by some calculations from these results based on these parameters (TP, TN, FP, and TN).

The Figure (4.30) shows the images that the model failed to classify them. In fact, they are ultra-sound images of a thyroid gland diagnosed by a benign tumor, and the model classified them as images of a malignant tumor.

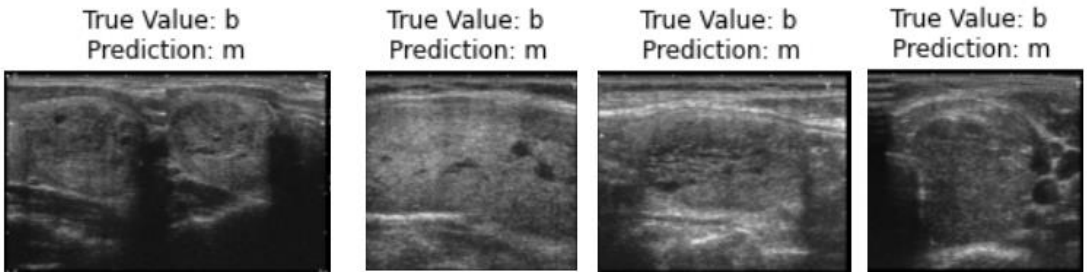



**Figure 4.30: Images that the VGG-16 model failed to classify**

The table (4.14) shows a summary of the results of the VGG-16 model

**Table 4.14: Summary of result for vgg-16 model**

| | precision | recall | f1-score | support |
|---|---|---|---|---|
| b | 1.00 | 0.67 | 0.80 | 12 |
| m | 0.83 | 1.00 | 0.90 | 19 |
| accuracy | | | 0.87 | 31 |

In conclusion, the table (4.14) shows the final results of the model, where we note that the overall accuracy of the model is 87%, where (b) is an abbreviation for a class benign tumor, while (m) is an abbreviation for class malignant tumor.

## 4.6 Challenges and Limitation

We encountered several challenges and obstacles during the collection of data used in the models. The most prominent problem was the lack of arrangement of medical laboratory data in health centers and local hospitals, and a large part of it was written on paper and not in computerized spreadsheets. The lack of standardization of some variables in medical analysis in laboratories, as each laboratory has variables that differ from the other; so we excluded a lot of data for this reason. Also, there are some records that contain missing data, and this problem we deal with it by using the mathematical methods we mentioned in Chapter 3.

One of the obstacles we encountered was the lack of sufficient image quantity for the ultra-sound Thyroiditis images dataset; and this somewhat reduced the accuracy of the (VGG-16) model. Also, this model needs a long training time on the graphic data and a processor with powerful specifications, and this was not available, as all the training operations were done on a Personal Computer (PC) with simple specifications that we mentioned in the introduction of this chapter. Better results could be obtained if a computer with higher specifications is used. Also, in order to generalize this research all over Palestine, data from many other cities larger population is needed to be collected.

# Chapter 5

# Conclusion and Future Work

## 5.1 Conclusion

Health is considered the human capital, without which he would not be able to carry out daily chores and various activities and it is one of the components of his happiness in this life. Therefore, a person must initiate periodic reassurance of his health in all stages of his life; because the discovery and diagnosis of any disease in its early stages contributes to the process of accelerating recovery and returning to the clear enjoyment of health. The term “thyroid disease” is often used to refer to common thyroid diseases, the most famous of which are hypothyroidism and hyperthyroidism [12] [16], as well as various infections. So, it was important to diagnose and detect thyroid diseases; because thyroid disease does not affect the thyroid gland in particular but rather exceeds its impact, it leads to imbalances in vital bodily functions [19] [20] [79]. Consequently, this work aimed to develop an intelligent medical diagnosis and decision support model that helps detect the disease at an early stage before it worsens.

The medical expert system will help in the task of rapid diagnosis so that treatment can be started immediately, and the effects of delays in detection and diagnosis are avoided. This is the role of expert systems in the medical field, which is to sustenance decision support systems [80]; because they contribute to making decisions based on human experience for a specialist in a particular field: Because the diagnosis is often confused and inaccurate.

In this work, we have covered several Machine Learning (ML) models and benefited from their advantages in classification and diagnosis in medical practice, as they contain valuable information. Medical data can be used to train different machine learning models and build expert systems from them; these enrich doctors' diagnoses and give them a new perspective.

In this thesis, all ML models (DT, NB, SVM, KNN, Ensemble, and MLPNNs) were applied on the global dataset from (UCI) to classify thyroid diseases; the model that achieved the highest classification accuracy was the Ensemble model by 99.6%. Then, we have applied all ML models (DT, NB, SVM, KNN, Ensemble, and MLPNNs) on the local dataset from Palestine to classify thyroid diseases; the model that achieved the highest classification accuracy was the Ensemble model by 91.10%. Also, we divided the local dataset into four subsets, which are: Nablus dataset, Qalqiliah dataset, Salfit dataset, and Ramallah dataset, the DT model obtained the highest classification accuracy among all the models that we had been applied to each sub-dataset, by 88.50%, 91.90%, 89.90%, and 92.40% respectively. Then we made a comparison between the classification accuracy of the local dataset and the global dataset after its modification (Custom global dataset), and the custom global dataset obtained a higher classification accuracy in all models. Hybrid Model (GAs-MLPNNs) was used to improve the accuracy of the classification of thyroid diseases, the model was applied on a Custom global dataset and Local (All cities) dataset, the results were as follows: 95.00% for the custom global dataset, and 96.00% for local (All cities) dataset. Finally, a VGG-16 model based on CNN was used to classify a group of ultrasound images of the thyroid gland into a malignant tumor or a benign gland tumor, and the accuracy result of the classification was 87.00%.

## 5.2 Future Work & Recommendations

In future work, we will try to expand the variables of the local dataset in intensive cooperation with endocrinologists to add clinical symptoms to the variables that give a more accurate diagnosis. More Hybrid systems can also be used like integrating Particle Swarm Optimization (PSO) algorithm with Genetic algorithm (GA) namely GA-PSO can be made so that it shows better results and accuracy than using single techniques.

Also, on the level of deep learning, it is possible in future work to form a local dataset from thyroid images and use other models for classification such as (VGG-19) for example and make comparisons in terms of accuracy in more than one model.

This work can be developed and create a user friendly interface application, that helps medical specialists to diagnose the thyroid diseases as well as diagnose medical images. Encouraging work to establish a share medical database for all medical laboratories in Palestine; in order to create a medical data repository for Palestine that includes all governorates; in order to collect larger data about many diseases and use them for scientific research in the field of machine learning.

## Bibliography

[1] Ioniţă, I., & Ioniţă, L. (2016). Prediction of thyroid disease using data mining techniques. BRAIN. Broad Research in Artificial Intelligence and Neuroscience, 7(3), 115-124

[2] Umar Sidiq, D., Aaqib, S. M., & Khan, R. A. (2019). Diagnosis of various thyroid ailments using data mining classification techniques. Int J Sci Res Coput Sci Inf Technol, 5, 131-6.

[3] Duggal, P., & Shukla, S. (2020, January). Prediction of thyroid disorders using advanced machine learning techniques. In 2020 10th International Conference on Cloud Computing, Data Science & Engineering (Confluence) (pp. 670-675). IEEE.

[4] Tyagi, A., Mehra, R., & Saxena, A. (2018, December). Interactive thyroid disease prediction system using machine learning technique. In 2018 Fifth International Conference on Parallel, Distributed and Grid Computing (PDGC) (pp. 689-693). IEEE.

[5] Chandel, K., Kunwar, V., Sabitha, S., Choudhury, T., & Mukherjee, S. (2016). A comparative study on thyroid disease detection using K-nearest neighbor and Naive Bayes classification techniques. CSI transactions on ICT, 4(2-4), 313-319.

[6] Prerana, P. S., & Taneja, K. (2015). Predictive data mining for diagnosis of thyroid disease using neural network. International Journal of Research in Management, Science & Technology, 3(2), 75-80.

[7] Rehman, S. U., Cope, D. W., Senseney, A. D., & Brzezinski, W. (2005). Thyroid disorders in elderly patients. Southern medical journal, 98(5), 543-550.

[8] Yadav, D. C., & Pal, S. (2019). To generate an ensemble model for women thyroid prediction using data mining techniques. Asian Pacific journal of cancer prevention: APJCP, 20(4), 1275.

[9] Jamkhandikar, D., & Priya, N. (2020). Thyroid Disease Prediction Using Feature Selection And Machine Learning Classifiers. The International Journal of analytical and experimental modal analysis, ISSN, (0886-9367).

[10] Raisinghani, S., Shamdasani, R., Motwani, M., Bahreja, A., & Raghavan Nair Lalitha, P. (2019, April). Thyroid prediction using machine learning techniques. In International Conference on Advances in Computing and Data Sciences (pp. 140-150). Springer, Singapore.

[11] Challa, M. (2019, September). Optimized Machine Learning Models for Diagnosis and Prediction of Hypothyroidism and Hyperthyroidism. In International Conference On Computational Vision and Bio Inspired Computing (pp. 286-293). Springer, Cham.

[12] Maniakas, A., Davies, L., & Zafereo, M. E. (2018). Thyroid disease around the world. Otolaryngologic Clinics of North America, 51(3), 631-642.

[13] Hou, Y., Chen, C., Zhang, L., Zhou, W., Lu, Q., Jia, X., ... & Zhan, W. (2021). Using deep neural network to diagnose thyroid nodules on ultrasound in patients with hashimoto's thyroiditis. Frontiers in Oncology, 11, 371.

[14] Shankarlal, B., Sathya, P. D., & Sakthivel, V. P. (2020). Computer-Aided Detection and Diagnosis of Thyroid Nodules Using Machine and Deep Learning Classification Algorithms. IETE Journal of Research, 1-12.

[15] Liu, Y., Xu, Y., Meng, X., Wang, X., & Bai, T. (2020). A study on the auxiliary diagnosis of thyroid disease images based on multiple dimensional deep learning algorithms. Current Medical Imaging, 16(3), 199-205.

[16] Gottwald-Hostalek, U., & Schulte, B. (2021). Low awareness and under-diagnosis of hypothyroidism. Current Medical Research and Opinion, (just-accepted), 1-10.

[17] Rafieian-Kopaei, M. (2018). Thyroid diseases: Pathophysiology and new hopes in treatment with medicinal plants and natural antioxidants. International Journal of Green Pharmacy (IJGP), 12(03).

[18] Gustin, K., Barman, M., Skröder, H., Jacobsson, B., Sandin, A., Sandberg, A. S., ... & Kippler, M. (2021). Thyroid hormones in relation to toxic metal exposure in pregnancy, and potential interactions with iodine and selenium. Environment International, 157, 106869.

[19] Nageswari, C. S., Kumar, M. V., Raveena, C., Sharma, J. S., & Devi, M. Y. (2021). An Identification and Classification of Thyroid Diseases Using Deep Learning Methodology. REVISTA GEINTEC-GESTAO INOVACAO E TECNOLOGIAS, 11(2), 2004-2015.

[20] Singh, S., Duggal, J., Molnar, J., Maldonado, F., Barsano, C. P., & Arora, R. (2008). Impact of subclinical thyroid disorders on coronary heart disease, cardiovascular and all-cause mortality: a meta-analysis. International journal of cardiology, 125(1), 41-48.

[21] Marmett, B., Carvalho, R. B., Fortes, M. S., & Cazella, S. C. (2018). Artificial Intelligence technologies to manage obesity. VITTALLE-Revista de Ciências da Saúde, 30(2), 73-79.

[22] Amisha, Malik, P., Pathania, M., & Rathaur, V. (2019, July). Overview of artificial intelligence in medicine. Retrieved January 02, 2021, from

[23] Sisodia, D., & Sisodia, D. S. (2018). Prediction of diabetes using classification algorithms. Procedia computer science, 132, 1578-1585.

[24] Khumaidi, A., Yuniarno, E. M., & Purnomo, M. H. (2017, August). Welding defect classification based on convolution neural network (CNN) and Gaussian kernel. In 2017 International Seminar on Intelligent Technology and Its Applications (ISITIA) (pp. 261-265). IEEE.

[25] Berndt, D. J., Fisher, J. W., Hevner, A. R., & Studnicki, J. (2001). Healthcare data warehousing and quality assurance. Computer, 34(12), 56-65.

[26] Desai, M. P. (1997). Disorders of thyroid gland in India. The Indian Journal of Pediatrics, 64(1), 11-20.

[27] Rajalakshmi, A. N., & Begam, F. (2021). Thyroid Hormones in the Human Body: A review. Journal of Drug Delivery and Therapeutics, 11(5), 178-182.

[28] Murphy, P. M., & Aha, D. W. (1994). UCI Repository of machine learning databases Irvine. CA: University of California, Department of Information and Computer Science.

[29] Malinowski, J. R., Denny, J. C., Bielinski, S. J., Basford, M. A., Bradford, Y., Peissig, P. L., ... & Crawford, D. C. (2014). Genetic variants associated with serum thyroid stimulating hormone (TSH) levels in European Americans and African Americans from the eMERGE Network. PloS one, 9(12), e111301.

[30] Reh, A., Grifo, J., & Danoff, A. (2010). What is a normal thyroid-stimulating hormone (TSH) level? Effects of stricter TSH thresholds on pregnancy outcomes after in vitro fertilization. Fertility and sterility, 94(7), 2920-2922.

[31] Cho, B. Y., Lee, H. K., Chung, J. H., Lee, S. I., Kim, W. B., & Koh, C. S. (2019). Evaluation of the Usefulness of Free T4 Measured by 2-Step Iimmunoextraction in the Patients with Thyroid and Non-Thyroid Diseases. Journal of Korean Endocrine Society, 10(2), 98-104

[32] Plowden, T. C., Schisterman, E. F., Sjaarda, L. A., Perkins, N. J., Silver, R., Radin, R., ... & Mumford, S. L. (2017). Thyroid-stimulating hormone, anti–thyroid antibodies, and pregnancy outcomes. American journal of obstetrics and gynecology, 217(6), 697-e1.

[33] Ataoğlu, H. E., Ahbab, S., Serez, M. K., Yamak, M., Kayaş, D., Canbaz, E. T., ... & Yenigün, M. (2018). Prognostic significance of high free T4 and low free T3 levels in non-thyroidal illness syndrome. European journal of internal medicine, 57, 91-95.

[34] Kim, E. Y., Kim, S. H., Rhee, S. J., Huh, I., Ha, K., Kim, J., ... & Ahn, Y. M. (2015). Relationship between thyroid-stimulating hormone levels and risk of depression among the general population with normal free T4 levels. Psychoneuroendocrinology, 58, 114-119.

[35] McNeil, A. R., & Stanford, P. E. (2015). Reporting thyroid function tests in pregnancy. The Clinical Biochemist Reviews, 36(4), 109.].

[36] Bekkering, G. E., Agoritsas, T., Lytvyn, L., Heen, A. F., Feller, M., Moutzouri, E., ... & Vermandere, M. (2019). Thyroid hormones treatment for subclinical hypothyroidism: a clinical practice guideline. Bmj, 365.

[37] Hegedüs, L. (2001). Thyroid ultrasound. Endocrinology and metabolism clinics of North America, 30(2), 339-360.

[38] Pedraza, L., Vargas, C., Narváez, F., Durán, O., Muñoz, E., & Romero, E. (2015, January). An open access thyroid ultrasound image database. In 10th International Symposium on Medical Information Processing and Analysis (Vol. 9287, p. 92870W). International Society for Optics and Photonics.

[39]http://cimalab.intec.co/applications/thyroid/?fbclid=IwAR3RXenVPMj_JE1LVqSne8CxnkC16F3hHriSfAH74BjGMIWXiaCh-JRE818

[40] Uddin, S., Khan, A., Hossain, M. E., & Moni, M. A. (2019). Comparing different supervised machine learning algorithms for disease prediction. BMC medical informatics and decision making, 19(1), 1-16.

[41] Osisanwo, F. Y., Akinsola, J. E. T., Awodele, O., Hinmikaiye, J. O., Olakanmi, O., & Akinjobi, J. (2017). Supervised machine learning algorithms: classification and comparison. International Journal of Computer Trends and Technology (IJCTT), 48(3), 128-138.

[42] Susmaga, R. (2004). Confusion matrix visualization. In Intelligent Information Processing and Web Mining (pp. 107-116). Springer, Berlin, Heidelberg.

[43] Tharwat, A. (2020). Classification assessment methods. Applied Computing and Informatics.

[44] Cawley, G. C., & Talbot, N. L. (2010). On over-fitting in model selection and subsequent selection bias in performance evaluation. The Journal of Machine Learning Research, 11, 2079-2107.

[45] Iwana, B. K., & Uchida, S. (2021). An empirical survey of data augmentation for time series classification with neural networks. Plos one, 16(7), e0254841.

[46] Chen, Y. W., & Lin, C. J. (2006). Combining SVMs with various feature selection strategies. In Feature extraction (pp. 315-324). Springer, Berlin, Heidelberg.

[47] Thomas, M. T. C. A. J., & Joy, A. T. (2006). Elements of information theory.

[48] Vergara, J. R., & Estévez, P. A. (2014). A review of feature selection methods based on mutual information. Neural computing and applications, 24(1), 175-186.

[49] Kaiser, J. (2014). Dealing with missing values in data. Journal of systems integration, 5(1), 42-51.

[50] Pratro, S., & Panda, B. S. (2020, March). A Novel Concept and Review on Retrieval of Missing Data. In 2020 International Conference on Computer Science, Engineering and Applications (ICCSEA) (pp. 1-4). IEEE.

[51] Niculescu-Mizil, A., & Caruana, R. (2005, August). Predicting good probabilities with supervised learning. In Proceedings of the 22nd international conference on Machine learning (pp. 625-632).

[52] Charbuty, B., & Abdulazeez, A. (2021). Classification based on decision tree algorithm for machine learning. Journal of Applied Science and Technology Trends, 2(01), 20-28.

[53] Ertam, F., & Aydın, G. (2017, October). Data classification with deep learning using Tensorflow. In 2017 international conference on computer science and engineering (UBMK) (pp. 755-758). IEEE.

[54] Yi, W., Lu, M., & Liu, Z. (2011). Multi-valued attribute and multi-labeled data decision tree algorithm. International Journal of Machine Learning and Cybernetics, 2(2), 67-74.

[55] Kesavaraj, G., & Sukumaran, S. (2013, July). A study on classification techniques in data mining. In 2013 fourth international conference on computing, communications and networking technologies (ICCCNT) (pp. 1-7). IEEE.

[56] Kotsiantis, S. B. (2013). Decision trees: a recent overview. Artificial Intelligence Review, 39(4), 261-283.

[57] Shannon, C. E. (2001). A mathematical theory of communication. ACM SIGMOBILE mobile computing and communications review, 5(1), 3-55.

[58] Song, Y. Y., & Ying, L. U. (2015). Decision tree methods: applications for classification and prediction. Shanghai archives of psychiatry, 27(2), 130.

[59] Jadon, P., Bhatia, D., & Mishra, D. K. (2019, December). A BigData approach for sentiment analysis of twitter data using Naive Bayes and SVM Algorithm. In 2019 Sixteenth International Conference on Wireless and Optical Communication Networks (WOCN) (pp. 1-6). IEEE.

[60] Huang, S., Cai, N., Pacheco, P. P., Narrandes, S., Wang, Y., & Xu, W. (2018). Applications of support vector machine (SVM) learning in cancer genomics. Cancer genomics & proteomics, 15(1), 41-51.

[61] Widodo, A., & Yang, B. S. (2007). Support vector machine in machine condition monitoring and fault diagnosis. Mechanical systems and signal processing, 21(6), 2560-2574.

[62] Liu, H., & Zhang, S. (2012). Noisy data elimination using mutual k-nearest neighbor for classification mining. Journal of Systems and Software, 85(5), 1067-1074.

[63] Bazmara, M., Movahed, S. V., & Ramadhani, S. (2013). KNN Algorithm for Consulting Behavioral Disorders in Children. Journal of Basic and Applied Scientific Research, 3, 12.

[64] Bar, A., Rokach, L., Shani, G., Shapira, B., & Schclar, A. (2013, May). Improving simple collaborative filtering models using ensemble methods. In International Workshop on Multiple Classifier Systems (pp. 1-12). Springer, Berlin, Heidelberg.

[65] Street, W. N., & Kim, Y. (2001, August). A streaming ensemble algorithm (SEA) for large-scale classification. In Proceedings of the seventh ACM SIGKDD international conference on Knowledge discovery and data mining (pp. 377-382).

[66] Reddy, G. T., Bhattacharya, S., Ramakrishnan, S. S., Chowdhary, C. L., Hakak, S., Kaluri, R., & Reddy, M. P. K. (2020, February). An ensemble based machine learning

model for diabetic retinopathy classification. In 2020 international conference on emerging trends in information technology and engineering (ic-ETITE) (pp. 1-6). IEEE.

[67] Zareapoor, M., & Shamsolmoali, P. (2015). Application of credit card fraud detection: Based on bagging ensemble classifier. Procedia computer science, 48(2015), 679-685.

[68] Bühlmann, P. (2012). Bagging, boosting and ensemble methods. In Handbook of computational statistics (pp. 985-1022). Springer, Berlin, Heidelberg.

[69] Özçift, A. (2011). Random forests ensemble classifier trained with data resampling strategy to improve cardiac arrhythmia diagnosis. Computers in biology and medicine, 41(5), 265-271.

[70] Abiodun, O. I., Jantan, A., Omolara, A. E., Dada, K. V., Umar, A. M., Linus, O. U., ... & Kiru, M. U. (2019). Comprehensive review of artificial neural network applications to pattern recognition. IEEE Access, 7, 158820-158846.

[71] Gardner, M. W., & Dorling, S. R. (1998). Artificial neural networks (the multilayer perceptron)—a review of applications in the atmospheric sciences. Atmospheric environment, 32(14-15), 2627-2636.

[72] Nahiduzzaman, M., Nayeem, M. J., Ahmed, M. T., & Zaman, M. S. U. (2019, December). Prediction of heart disease using multi-layer perceptron neural network and support vector machine. In 2019 4th International conference on electrical information and communication technology (EICT) (pp. 1-6). IEEE.

[73] Kumar, M., Husain, M., Upreti, N., & Gupta, D. (2010). Genetic algorithm: Review and application. Available at SSRN 3529843.

[74] Yu, D., Wang, H., Chen, P., & Wei, Z. (2014, October). Mixed pooling for convolutional neural networks. In International conference on rough sets and knowledge technology (pp. 364-375). Springer, Cham.

[75] Traore, B. B., Kamsu-Foguem, B., & Tangara, F. (2018). Deep convolution neural network for image recognition. Ecological Informatics, 48, 257-268.

[76] Simonyan, K., & Zisserman, A. (2014). Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv:1409.1556.

[77] Raghavendra, U., Fujita, H., Bhandary, S. V., Gudigar, A., Tan, J. H., & Acharya, U. R. (2018). Deep convolution neural network for accurate diagnosis of glaucoma using digital fundus images. Information Sciences, 441, 41-49.

[78] Yu, W., Yang, K., Bai, Y., Xiao, T., Yao, H., & Rui, Y. (2016, June). Visualizing and comparing AlexNet and VGG using deconvolutional layers. In Proceedings of the 33 rd International Conference on Machine Learning.

[79] Klemperer, J. D. (2002). Thyroid hormone and cardiac surgery. Thyroid, 12(6), 517-521.

[80] Bashir, S., Qamar, U., Khan, F. H., & Naseem, L. (2016). HMV: A medical decision support framework using multi-layer classifiers for disease prediction. Journal of Computational Science, 13, 10-25.

[81] Baxt, W. G. (1995). Application of artificial neural networks to clinical medicine. The lancet, 346(8983), 1135-1138.

[82] Suykens, J. A., & Vandewalle, J. (1999). Least squares support vector machine classifiers. Neural processing letters, 9(3), 293-300.

[83] Adeniyi, D. A., Wei, Z., & Yongquan, Y. (2016). Automated web usage data mining and recommendation system using K-Nearest Neighbor (KNN) classification method. Applied Computing and Informatics, 12(1), 90-108.

[84] Mariotti, S., Franceschi, C., Cossarizza, A., & Pinchera, A. (1995). The aging thyroid. Endocrine reviews, 16(6), 686-715.

# الملخص

تعتبر الغدة الدرقية من أهم أجزاء الجسم، حيث أن أي خلل فيها يؤدي إلى العديد من المشاكل الصحية. تعتبر فحوصات الغدة الدرقية من الفحوصات الضرورية الدورية التي يجب إجراؤها للاطمئنان على الصحة، حيث أن أي مشكلة فيها تتسبب في مضاعفات صحية إذا لم يتم علاجها في الوقت المناسب. لذلك، هناك حاجة ماسة للكشف المبكر عن هذه الأمراض وكذلك الدقة في التشخيص ، وهنا دور واضح يمكن لنهج الذكاء الاصطناعي (AI) التدخل وأن يدعم القرارات الطبية للكشف عن أمراض الغدة الدرقية قبل أن تتفاقم حالة المرض وتؤثر على الجسم بمشاكل أخرى. يستخدم هذا البحث النموذج الهجين للتعلم الآلي (ML) للشبكات العصبية (NNs) والخوارزميات الجينية [GAs–MLPNNs] (GAs)، ونموذج التعلم العميق (DL) لتصنيف أمراض الغدة الدرقية والتنبؤ بها اعتمادًا على الميزات والصور الطبية.

يعتمد هذا البحث على عدة مصادر للبيانات، فقد استخدمنا مجموعة بيانات عالمية، كما تم أيضا جمع مجموعات بيانات محلية من أربع مدن في فلسطين، ومجموعة بيانات عالمية لصور موجات فوق صوتية للغدة الدرقية. تم عمل معالجة مسبقة للبيانات، وكذلك استخراج الميزات لمجموعات البيانات. تتمثل الخطوة الأخيرة في تطبيق أساليب مختلفة للذكاء الاصطناعي للتعرف على الأنماط في مجموعات البيانات ومن ضمن النماذج التي تم استخدامها. نموذج شجرة القرار (DT)، نموذج بايز الساذج (NB)، نموذج آلة المتجهات الداعمة (SVM)، نموذج الجار الأقرب(KNN), نموذج طرق Ensemble ،الشبكات العصبية متعددة الطبقات (MLPNNs)، نموذج هجين يعتمد على الخوارزمية الجينية مع نموذج الشبكات العصبية متعددة الطبقات (GAs–MLPNNs) ، والتعلم العميق حيث تم استخدام نموذج (VGG–16) مبني على أساس الشبكة العصبية التلافيفية (CNN) من أجل التعرف على النمط الموجود في مجموعات البيانات الصورية وتحسين نتائج دقة تصنيف التهابات الغدة الدرقية بدقة عالية.

في المرحلة الأولى، تـــم تطبيـــق DT و NB و SVM و KNN و Ensemble و MLPNNs على مجموعة البيانات العالمية لتصنيف أمراض الغدة الدرقية، وكانت نتائج الدقة كمـــا يلـــي: 99.5٪ ، 93.3٪ ، 98.2٪ ، 95.1٪ ، 99.6٪ و 95.6٪ على التوالي. في المرحلة الثانية، تم تطبيق DT و NB و SVM و KNN و Ensemble و MLPNNs على مجموعة البيانـــات المحلية بمرحلتين :المرحلة الأولى كانت البيانات الكاملة التي تم جمعها مـــن المـــدن الأربـــع (مجموعة بيانات جميع المدن)، والمرحلة الثانية: بيانات كل مدينة على حدى. حيـــث حصـــلت مجموعات بيانات جميع المدن على أعلى دقة عند تطبيق نموذج Ensemble بنسبة 91.10٪. وكان أفضل نموذج من حيث دقة النتائج لكل مدينة على حدى هو نموذج شجرة القـــرار (DT) الذي نتج عن تطبيقه الحصول على الدقة ؛ 92.40٪ ، 88.50٪ ، 91.90٪ ، 89.90٪ فـــي مجموعات بيانات كل من مدينة رام الله ونابلس وقلقيلية وسلفيت على التوالي.

في المرحلة الثالثة ، تم إنشاء مجموعة بيانات عالمية مخصص عن طريـــق اســـتبعاد بعـــض المتغيرات الغير مشتركة مع متغيرات مجموعة البيانات المحليـــة لمقارنـــة نتـــائج التصـــنيف بمجموعات البيانات المحلية (مجموعة بيانات كل المدن). في المرحلة الرابعـــة، تـــم اســـتخدام النموذج الهجين (GAs-MLPNNs) لتحسين دقة تصنيف أمراض الغدة الدرقية ، وتم تطبيـــق النموذج على مجموعة البيانات العالمية المخصصة ومجموعة البيانات المحلية (مجموعـــة كـــل المدن)، وكانت النتائج على النحو التالي: 95٪ لمجموعة البيانات العالمية المخصصـــة، و96٪ لمجموعة البيانات المحلية (مجموعة بيانات كل المدن). في المرحلة الخامســـة والأخيـــرة، تـــم استخدام نموذج VGG-16 المعتمد على الشبكة العصبية التلافيفية CNN لتصنيف مجموعة من صور الموجات فوق الصوتية للغدة الدرقية إلى صور ورم خبيث أو صور ورم غـــدة حميـــد، وكانت نتيجة دقة التصنيف 87.00 % .