



PEOPLE'S DEMOCRATIC REPUBLIC OF ALGERIA
Ministry of Higher Education and Scientific Research
University of Mohamed Khider – BISKRA
Faculty of Exact Sciences, Science of Nature and Life
Computer Science Department

Ordre N° : IA26/M2/2023

Thesis

Submitted in fulfilment of the requirements for the Masters degree in

Computer science

Option : **Artificial Intelligence**

Titre

Deep learning approach for diabetes prediction

By :

ADISSA MOHANNED

Members of the jury :

Name	Grade	President
BOUREKKACHE Samir	MCA	Supervisor
Name	Grade	Member

Session 2023

Dedicace

To my parents Mohamed and Noura,

To my brother Ahmed Tayeb and my sister Soundous,

Thank you for supporting me at all times.

Acknowledgements

First and foremost I would like to thank Allah Almighty for giving me the strength and knowledge to complete this work.

I would like to thank my parents, my brother and my sister for supporting me and for filling my life with joy and positivity.

I would like to thank my supervisor Doctor **BOUREKKACHE Samir**, for guiding me throughout this project.

Finally, to those who encouraged me from my teachers and my friends. Thank you all for supporting me.

May Allah Almighty continue to bless you all.

ADISSA Mohanned

Abstract

Diabetes is a chronic disease that needs to be taken seriously, as it can lead to far more dangerous complications if left untreated. Early detection of this disease can greatly help in curing and saving patients. The progression made by artificial intelligence provides solutions that can be used in order to predict diabetes. In this work, we talked about diabetes, machine and deep learning and the approaches that are used to predict diabetes.

In our project, we have used the LSTM architecture to build a diabetes prediction model, our model achieved accuracy of 98.64% and validation accuracy of 96.99%.

Keywords: Diabetes, Artificial intelligence, Machine learning, Deep learning.

Summary

General introduction.....	1
Chapter I: Diabetes.....	3
I.1. Definition.....	3
I.2. Types of Diabetes.....	4
I.2.1. Type 1 Diabetes	4
I.2.2. Type 2 Diabetes	7
I.2.3. Gestational Diabetes	9
I.3. Complications of diabetes	11
I.4. Prevention of diabetes	11
I.5. Conclusion.....	12
Chapter II: Machine and Deep learning.....	13
II.1. Machine Learning.....	13
II.1.1. Types of Machine Learning	14
II.2. Deep Learning	17
II.2.1. Artificial neural network.....	17
II.3. Conclusion.....	21
Chapter III: Diabetes prediction	22
III.1. Convolutional Neural Networks (CNNs).....	22
III.2. Recurrent Neural Networks (RNNs).....	25
III.2.1. Types of Recurrent neural networks	27
III.3. Long Short-Term Memory (LSTM)	27

III.4.	Deep Belief Networks (DBNs)	28
III.5.	Related Works	29
III.6.	Conclusion.....	35
Chapter IV: Design of the approach		36
IV.1.	Global architecture of the system.....	36
IV.1.1.	Dataset.....	37
IV.1.2.	Preprocessing	37
IV.1.3.	Splitting dataset.....	38
IV.1.4.	LSTM	38
IV.1.5.	Prediction	40
IV.1.6.	Model evaluation.....	40
IV.2.	Conclusion.....	41
Chapter V: Implementation and results		42
V.1.	Implementation frameworks, tools and libraries.....	42
V.2.	Implementation phases	43
V.2.1.	Loading dataset	44
V.2.2.	Preprocessing	45
V.2.3.	LSTM Model.....	49
V.3.	Results	54
V.4.	Comparison with other related works	57
V.5.	Conclusion.....	58
General conclusion.....		59
References		60

List of Figures

Figure I.1 Insulin Pen [8].....	7
Figure II.1 Supervised learning [20].....	14
Figure II.2 Unsupervised learning [22].....	15
Figure II.3 Semi supervised learning [22].....	16
Figure II.4 Reinforcement learning [25].....	17
Figure II.5 Neural network [28].....	18
Figure II.6 Artificial neuron [29].....	19
Figure II.7 Sigmoid function [31].....	19
Figure II.8 ReLU function [33].....	20
Figure II.9 Tanh function [34].....	21
Figure III.1 Convolutional Neural Network Architecture [37].....	22
Figure III.2 Convolved Feature [54].....	23
Figure III.3 Example of Max pooling and Average pooling [54].....	24
Figure III.4 Flattening [40].....	25
Figure III.5 Artificial neural network to Reccurent nerual network [42].....	26
Figure III.6 LSTM network architecture [55].....	28
Figure III.7 Deep Belief networks architecture [56].....	29
Figure III.8 Block diagram of the proposed method.....	30
Figure III.9 Architecture of proposed system for detection of diabetes.....	31
Figure III.10 Proposed method process (D: Diabetic; ND: Not Diabetic).....	32
Figure III.11 Proposed Model for Diabetes Prediction.....	33
Figure III.12 Block diagram of proposed methodology.....	34
Figure IV.1 General Architecture.....	36
Figure IV.2 Dataset Structure.....	37
Figure V.1 Model Libraries.....	43
Figure V.2 Loading dataset.....	44
Figure V.3 Dataset.....	44
Figure V.4 Code for number of samples per class.....	44
Figure V.5 Checking null values.....	45
Figure V.6 Dataset description.....	46

Figure V.7 Data Filtering.....	46
Figure V.8 Selection of features and target.....	47
Figure V.9 Splitting dataset to training and testing.....	47
Figure V.10 Data Scaling.....	48
Figure V.11 Before Scaling	48
Figure V.12 After Scaling.....	48
Figure V.13 Before reshaping.....	48
Figure V.14 Data Reshaping.....	49
Figure V.15 After reshaping	49
Figure V.16 Sequential model	49
Figure V.17 LSTM model architecture.....	49
Figure V.18 Optimizer	50
Figure V.19 Compile	50
Figure V.20 LSTM Model.....	50
Figure V.21 Model Summary	51
Figure V.22 Model Fitting.....	51
Figure V.23 Model evaluation	52
Figure V.24 Model evaluation	53
Figure V.25 Model visualization	53
Figure V.26 Model Accuracy	54
Figure V.27 Model Loss	55
Figure V.28 Model Accuracy	55
Figure V.29 Model testing.....	56
Figure V.30 Confusion matrix	57

List of Tables

Table I.1 Comparison for different diagnose [7]	6
Table III.1 Comparison of related works on diabetes prediction using deep learning techniques.....	34
Table V.1 Number of samples per class	45
Table V.2 Number of samples per class	46
Table V.3 Model results.....	55
Table V.4 Comparison of our model with related works.....	58

General introduction

In recent times, the speed and the accuracy has become more important due to the progression made by many fields, including the medical field. The diversity and abundance of diseases necessitated scientists to find an effective method that helps the specialist in diagnosing human disease. Among these methods is the use of artificial intelligence, which can exceed human potential.

Artificial intelligence has been incorporated into the medical field to help in data analysis and decisions making. We chose this work in order to treat one of the most common diseases, which is diabetes.

Problematic

Diabetes is a chronic disease that affects the human body. This disease is caused by the high levels of blood sugar. It is better to diagnose or predict this disease in the early stages in order to cure the patients as it can lead to serious complications if it is left untreated.

Objective of the work

The objective of this work is to build a deep learning model that can help in predicting diabetes.

The dissertation is structured as follows:

- **Chapter I: Diabetes.** This chapter talks about the diabetes, the different types of this disease, the symptoms and the diagnostics as well as the treatments of diabetes.
- **Chapter II: Machine and Deep learning.** This chapter focus on the subfields of artificial intelligence, machine and deep learning.
- **Chapter III: Diabetes prediction.** In this chapter, we will discuss the different approaches in deep learning that were used to predict diabetes.

- **Chapter IV: Design of the approach.** This chapter presents the design of our model and the steps that we will follow in order to build our model.
- **Chapter V: Implementation and results.** In this final chapter, we will present the implementation code of our model, the results achieved, and finally we compare our model with other related works.

Chapter I: Diabetes

Health diseases are a topic that is taken very seriously among the scientists and doctors around the globe who are trying to work on finding treatments and possible cures to these diseases that pose a threat to human life.

Diabetes is one of these chronic health diseases that affects how your body turns food into energy. According to IDF (International Diabetes Federation) in 2021, around 537 million adults (20-79 years) are living with diabetes, it is expected to rise in 2030 to a number of 643 million, and 783 million by 2045. This disease caused 6.7 million deaths - 1 in every 5 seconds [1]. In Algeria, more than 2 million have diabetes in 2021 [2]. In this chapter, we are going to talk about Diabetes (Types, Symptoms, Diagnostics, and Treatments)

I.1. Definition

Diabetes is a long lasting health condition that happens when your blood sugar (Glucose) is too high. Human body have an organ called pancreas that make Insulin, which is the hormone that helps glucose get into your cells and use it to generate energy [3].

If glucose remain in your blood and does not reach your cells, then this means two things either your body did not make enough insulin or it did not use it well. This can lead to having very severe health problems [3].

I.2. Types of Diabetes

There are three main types of diabetes:

- Type 1
- Type 2
- Gestational Diabetes

I.2.1. Type 1 Diabetes

I.2.1.1. Definition

Type 1 diabetes is an immune system disorder, which means that your own immune system attacks the insulin producing cells in the pancreas. This reaction from your immune system can stop your pancreas from making insulin or making very little insulin [4].

Having your insulin producing cells attacked can lead to high blood sugar inside your bloodstream that causes the symptoms and complications of diabetes.

Type 1 diabetes also known as insulin-dependent or juvenile diabetes, juvenile means that it usually develops in children, teens and young adults and it can happen at any age [5].

About 5 to 10 % people that have diabetes of type 1.

I.2.1.2. Cause

The main cause of type 1 diabetes is the autoimmune reaction this reaction destroys cells called “beta cells” which are the cells that make the insulin hormone inside the pancreas [5].

Autoimmune reaction is a condition in which the body mistakes its own healthy tissues as foreign and attacks them.

There are other possible causes as well: it can be due to genetics or viruses [5].

The symptoms of type 1 diabetes can take months or years to show up.

I.2.1.3. Symptoms

The symptoms of Type 1 diabetes develop very quickly and can appear suddenly at any given time.

Here we list some of these symptoms:

- Feeling more thirsty than usual
- Unexplained weight loss
- Blurry vision
- Severe fatigue and feeling weak

These are the main signs of having **hyperglycemia (blood glucose level too high)** and if diabetes is untreated then it can lead to having a life threatening complication called **diabetes ketoacidosis (DKA)** [6].

Some symptoms of DKA:

- Trouble breathing
- Drowsiness
- Nausea or vomiting
- Stomach pain
- Loss of focus
- A fruity smell of the breath

I.2.1.4. Diagnostic

There are 4 possible ways to diagnose diabetes: Glycated hemoglobin (A1C) test, Fasting blood sugar test, Glucose tolerance test, Random blood sugar test [7].

- **Glycated hemoglobin (A1C) test:** this test measures your average blood sugar level over the past 2 or 3 months. A result of 5.7% is normal, between 5.7 and 6.4% indicates you have prediabetes and 6.5% or higher means you have diabetes.
- **Fasting blood sugar test:** In this test, you need to be fasting overnight. Having a fasting blood sugar level of 99 mg/dL or lower is normal and 100 to 125 mg/dL indicates you have prediabetes and 126 mg/dL or high means you have diabetes.
- **Glucose tolerance test:** This is where you get your blood sugar measured while fasting and then you drink liquid that contains glucose and after 1 to 2 hours or even 3 hours, you test again. At 2 hours, if your blood sugar level is 140 mg/dL or lower then it is normal and you do not have diabetes and if it is 140 to 199 mg/dL than you have prediabetes and if it is 200mg/dL or higher than this indicates that you have diabetes.
- **Random Blood Sugar Test:** You can take this test at any time without the need to fast. A blood sugar level of 200 mg/dL or higher means you have diabetes.

Table I.1 Comparison for different diagnose [7]

Result*	A1C Test	Fasting Blood Sugar Test	Glucose Tolerance Test	Random Blood Sugar Test
Diabetes	6.5% or above	126 mg/dL or above	200 mg/dL or above	200 mg/dL or above
Prediabetes	5.7 – 6.4%	100 – 125 mg/dL	140 – 199 mg/dL	N/A
Normal	Below 5.7%	99 mg/dL or below	140 mg/dL or below	N/A

I.2.1.5. Treatments

There is no known cure for type 1 diabetes right now and because it is known as insulin-dependent then the only way to keep the blood sugar level close to normal is by taking **insulin hormone** using syringe or insulin pump further more you can help your own body by changing your lifestyle to doing more exercises and keeping a healthy weight as well as eating healthy foods [5].



Figure I.1 Insulin Pen [8]

I.2.2. Type 2 Diabetes

I.2.2.1. Definition

Type 2 diabetes occurs when your blood sugar level is too high and you develop **insulin resistance** this means that your cells do not respond normally to insulin [9].

About 90 to 95% of people have type 2 diabetes.

I.2.2.2. Cause

Type 2 diabetes happens when the body tries to adapt by increasing the amount of insulin, which is called **hyperinsulinism** in hopes of getting the cells to respond to the insulin [10].

Eventually your pancreas can not keep up and your beta cells become exhausted and can no longer produce sufficient insulin then your blood sugar level rises and can cause other health problems such as heart disease, vision loss and kidney disease [\[10\]](#).

I.2.2.3. Symptoms

These are some signs and symptoms of type 2 diabetes [\[9\]](#)[\[11\]](#):

- Weight loss
- Headaches
- Dry mouth
- Feeling thirsty
- Fatigue
- Constant hunger

In addition, it is important to note that sometimes people do not experience any symptoms at all so considering other factors such as family history of diabetes or being overweight might help detecting it in the early stages [\[9\]](#).

I.2.2.4. Diagnostic

The diagnostic of type 2 diabetes have the same procedure of type 1 diabetes nothing different between the two types of diabetes.

I.2.2.5. Treatments

Type 2 diabetes can be managed, and in some cases reversed. The treatments for type 2 diabetes help to manage blood sugar levels and reduce the risk of diabetes complication [\[12\]](#).

Having a better diet and doing exercises is extremely important for managing type 2 diabetes and if that is not enough then your doctor will prescribe medication that help your body use insulin more effectively.

Monitoring your blood sugar is necessary because it is the only way to know whether you are meeting your target levels.

I.2.3. Gestational Diabetes

I.2.3.1. Definition

Gestational diabetes is a type of diabetes that develops during pregnancy.

Developing this type of diabetes does not mean that you already had diabetes before you got pregnant. The condition appears because of pregnancy.

Gestational diabetes is usually diagnosed in the 24th to 28th week of pregnancy.

Gestational diabetes most often disappears after the birth of the baby [\[13\]](#).

I.2.3.2. Cause

When the body can not produce enough insulin during pregnancy then this can lead to having very high blood sugar level this increase of glucose level is caused by hormones released by the placenta during pregnancy. The hormone that the placenta release is called human placental lactogen (HPL). The HPL makes the women's body less sensitive to insulin and less able to use it properly [\[14\]](#).

This hormone raises the blood glucose level so that the baby can get the nutrients it needs from the extra glucose in the blood.

At 15 weeks of pregnancy, another hormone called human placental growth hormone increases and causes the blood glucose level to increase as well this hormone can also help regulate the mother's blood glucose level to be sure that the baby gets the right amount of needed nutrients [\[14\]](#).

I.2.3.3. Symptoms

Some of these symptoms are common during pregnancy and are not necessarily a sign of gestational diabetes [15].

- Increased thirst
- Blurred vision
- Tiredness
- Dry mouth
- Nausea

I.2.3.4. Diagnostic

You will test your blood during pregnancy like the other types of diabetes 1 and 2 but in this test it will focus mainly on **glucose challenge test** where you drink a sweet liquid then after 1 hour you are tested and if your blood glucose level is high then you do the **glucose tolerance test** where you fast before the test and test your blood glucose level then after drinking sweet liquid and about 1 to 2 hours passed you are tested again. This test can confirm if you have gestational diabetes [13].

The older a women becomes pregnant the higher the risk of developing gestational diabetes during her pregnancy.

I.2.3.5. Treatments

It is important to treat the gestational diabetes because if left untreated then it can lead to complication for both the mother and the baby.

Controlling your blood sugar is always the goal in all different types of diabetes and this is accomplished by eating wisely, doing exercises, taking medications, monitoring your blood glucose [14].

I.3. Complications of diabetes

Like every other health disease, diabetes have chronic complications as well and they can lead to serious damage if they are unchecked and untreated [16].

- **Eye problems (retinopathy):** people with diabetes can develop an eye disease called diabetic retinopathy, which can affect their eyesight. Taking eye screening test can prevent from having this type of disease and can prevent sight loss.
- **Heart disease:** having a high blood sugar can damage your blood vessels. This can sometimes lead to heart attacks.
- **Kidney problems (nephropathy):** diabetes can cause damage to your kidney over a long period of time and make it harder to clear extra fluid and waste from your body.
- **Nerve damage (neuropathy):** complications of high blood sugar level can cause nerve damage and this affect the functionality between the brain and every part of the body.

I.4. Prevention of diabetes

As of today, there is no known prevention for type 1 diabetes.

However, for type 2 diabetes as we mentioned before you can lower the risk of it by managing your weight and your diet and doing exercises regularly.

In addition, for gestational diabetes or prediabetes these habits can help delay or prevent having type 2 diabetes.

I.5. Conclusion

In this chapter, we presented the chronic disease of diabetes, talked about the different types of the disease and their symptoms as well as the diagnosis and treatment of the disease. We learned that this disease must be taken seriously and treated effectively. To this day diabetes remains a disease that can be treated very effectively but can not be cured.

Chapter II: Machine and Deep learning

Artificial intelligence (AI) is a process of imitating human intelligence. It involves the development of algorithms and models that helps the computers to learn from a large amount of data and be able to interact with humans or other machines. AI has been effectively used in various fields to perform tasks that requires intelligence and processing large amounts of data. In this chapter, we will talk about the subfields of AI both Machine and Deep learning.

II.1. Machine Learning

Machine learning is a subfield of artificial intelligence (AI), and it is defined as the capability of a machine to imitate human intelligence with the help of algorithms that allow the machine to learn and make predictions based of datasets [17].

Tom M. Mitchel defined Machine learning by saying that “Machine learning is the study of computer algorithms that allow computer programs to automatically improve through experience”.

From the definition that was given by **Tom M. Mitchel**, we can conclude that there is a similarity between machine learning and human beings as they all learn by experimentation. If a human is tested in some tasks, he will gain experience and knowledge over time resulting in a higher performance. Same thing for machine learning the machine learning algorithm is trained using training data set it can be labelled or unlabeled data and then give it new input data and measure the accuracy of the prediction if it is acceptable then the machine learning algorithm will be used if it is not then the machine learning algorithm will be trained again with bigger datasets [18].

The main types of machine learning include:

- Supervised learning.
- Unsupervised learning.
- Semi-supervised learning.
- Reinforcement learning.

II.1.1. Types of Machine Learning

II.1.1.1. Supervised Learning

In supervised learning, the model is trained using labelled data, which means that we have the input data as well as the result that should be predicted by the machine most of the time the dataset is split into a training dataset and a testing dataset. The training dataset is used to train the model and the test dataset to evaluate the model after the training. Having labelled data gives the machine knowledge and be able to predict new never seen before samples [19].

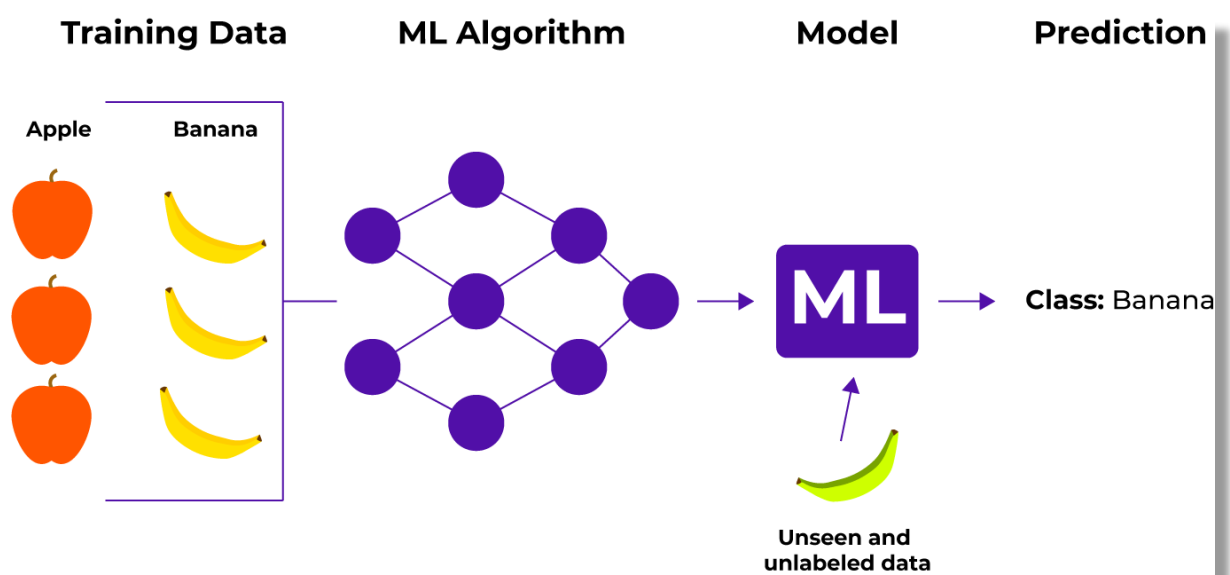


Figure II.1 Supervised learning [20]

The figure II.1 shows how supervised learning is done in this example we have Training data that contains “Two fruits” and they are labelled “Apple, Banana” then we feed this training dataset to the machine learning algorithm after the training we will have a model and with that model we can input new unseen and unlabeled data and have the model predicts the outcome of the input either apple or banana.

II.1.1.2. Unsupervised Learning

Unsupervised learning means that the model is trained using unlabeled and unclassified dataset. The unsupervised learning algorithms discover hidden patterns without the need for human intervention. Discovering hidden patterns make the unsupervised learning the ideal solution for different tasks [21]:

- **Clustering:** grouping unlabeled data based on their similarities or differences.
- **Association:** finding relationships between variables in a given dataset typically used in market data analysis.

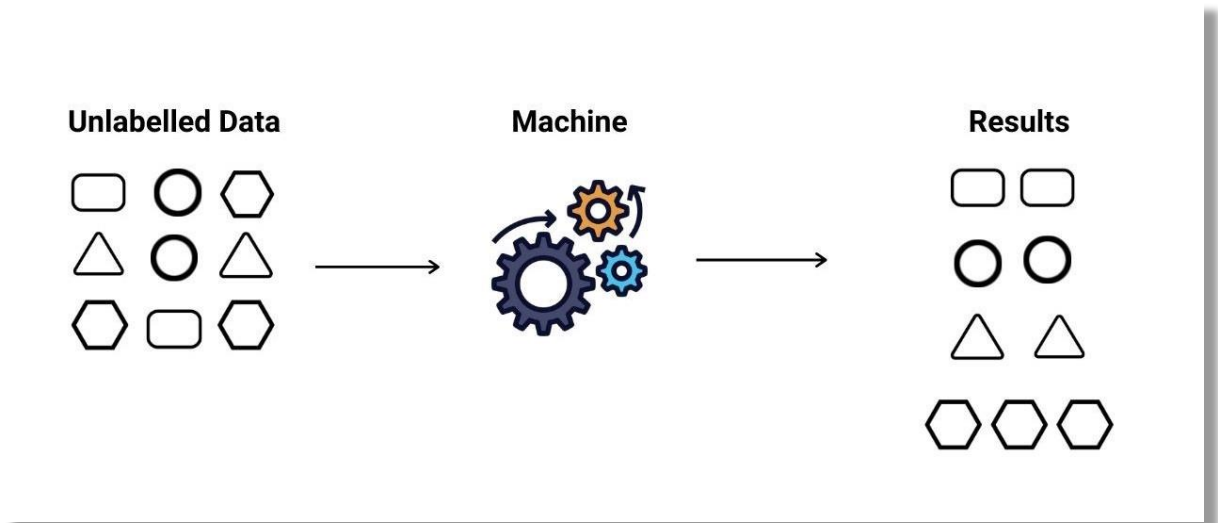


Figure II.2 Unsupervised learning [22]

II.1.1.3. Semi-supervised Learning

This type of learning is a combination between supervised and unsupervised learning. The supervised learning is used to train the algorithm with labelled data and the unsupervised learning is used as a test data with unlabeled data. Improving the performance of the algorithms using labelled data can be challenging and time consuming so the solution was to create the semi supervised learning that calibrate between the performance of supervised learning with the efficiency of unsupervised learning [23].

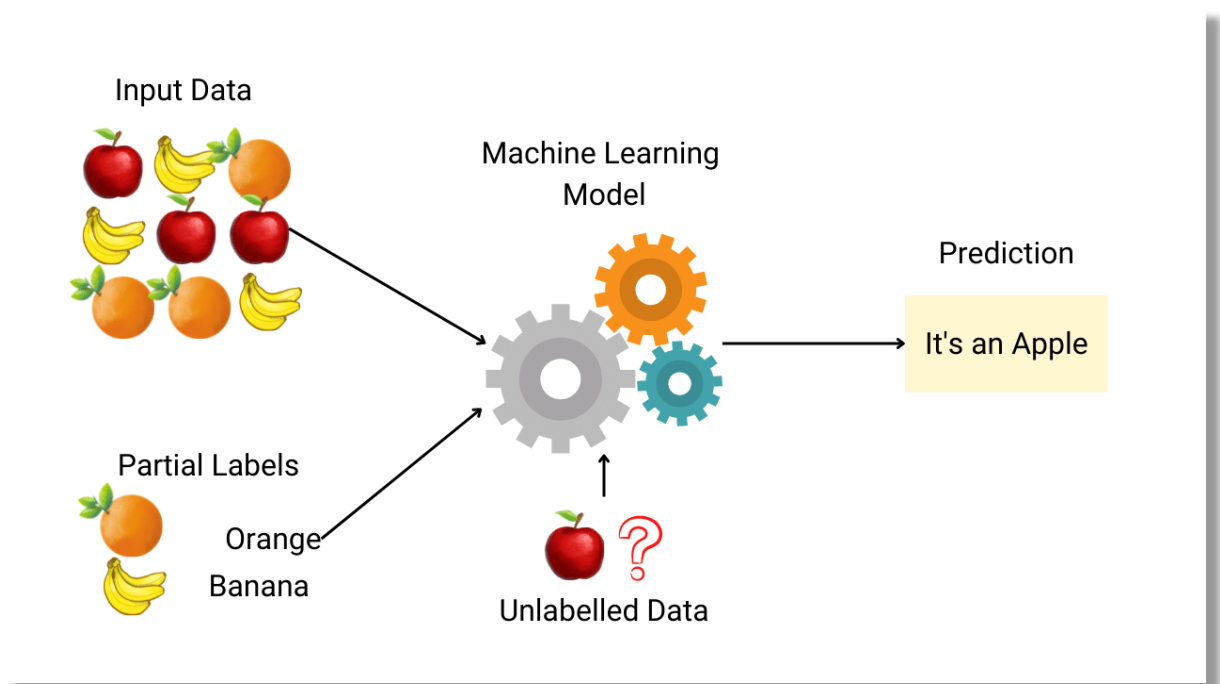


Figure II.3 Semi supervised learning [22]

II.1.1.4. Reinforcement Learning

Reinforcement learning is a machine learning algorithm that seek positive rewards meaning that the agent within the environment can get positive or negative rewards according to the actions taken by him, positive if it is beneficial toward the goal or negative if it is not. Reinforcement learning can be used in robotics, videogames... [24]

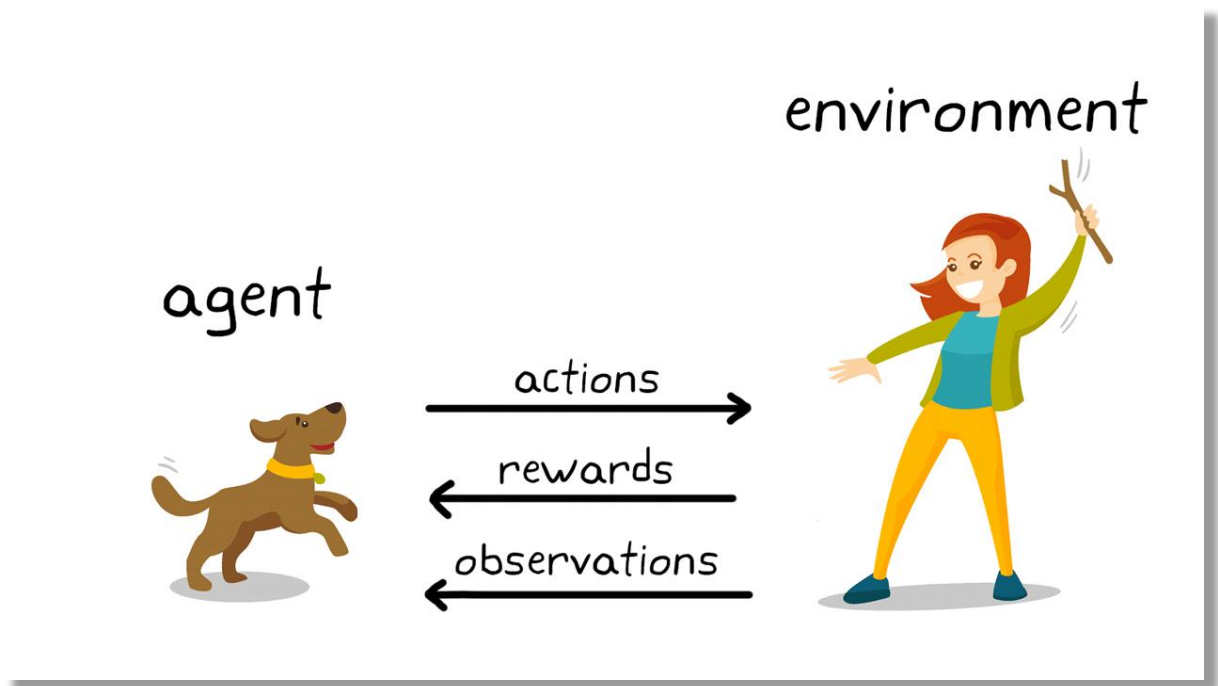


Figure II.4 Reinforcement learning [25]

II.2. Deep Learning

Deep learning is a subset of machine learning, which is a neural network with three or more layers. The training that is done to these neural network help to mimic the human brain. Deep learning is powerful in tasks like classification, natural language processing and speech recognition [26].

Deep learning models are trained using a very large data unlike machine learning where the data is smaller than deep learning [26].

II.2.1. Artificial neural network

Artificial neural networks are made of node layers, they contain input layer and one or more hidden layer and lastly the output layer and they are all connected with each other. Each node has a weight and threshold. If the output of any node is above the specified threshold then that node is activated and the data is passed to the next layer, if it is not above the threshold then the node will not be activated [27].

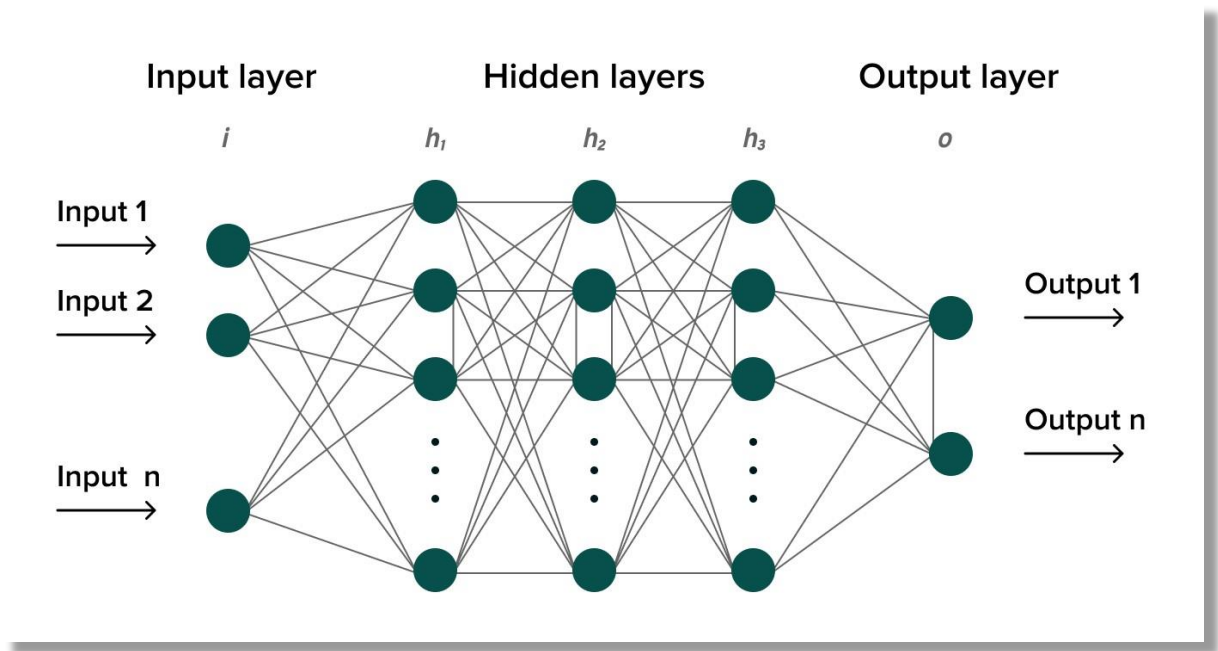


Figure II.5 Neural network [28]

II.2.1.1. Types of layers

From the figure II.5, we can see that the neural network has 3 types of layers

Input layer: The first layer in the neural network that receive data and it is composed of input neurons.

Hidden layer: can be one or more layers, they are located between the input, and the output layer in this hidden layer the problem will be solved.

Output layer: The last layer in the neural network and it is the layer that produce the output of the program.

II.2.1.2. Activation function

An activation function decide whether the neuron should be activated or not. The activation function transform the summed weighted input from the node into an output value that will be fed to the next hidden layer or as output [29].

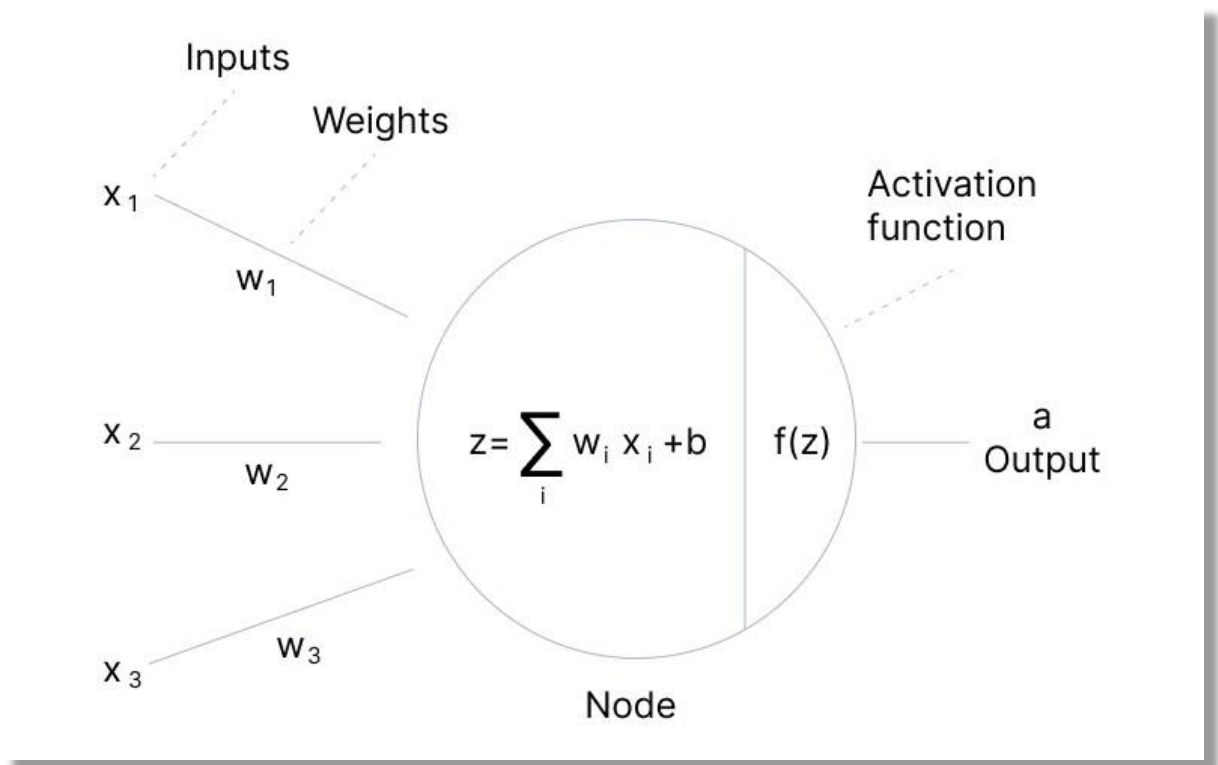


Figure II.6 Artificial neuron [29]

We will mention the most used activation function:

II.2.1.2.1. Sigmoid function:

The value of this function is between 0 and 1 and it takes the shape of letter “S” and it is commonly used for binary classification methods where we only have 2 classes [30].

This function is define by the following equation:

$$S(x) = \frac{1}{1 + e^{-x}}$$

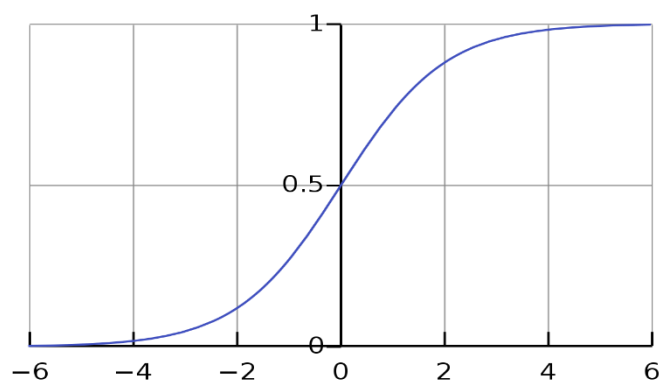


Figure II.7 Sigmoid function [31]

II.2.1.2.2. ReLU function:

ReLU (Rectified Linear Unit) is the most popular and most often used activation function, the output of this function is in the range of 0 to ∞ and that because if this function receive any negative value it replace it with 0 [32].

The equation of this function is:

$$f(x) = \max(0, x)$$

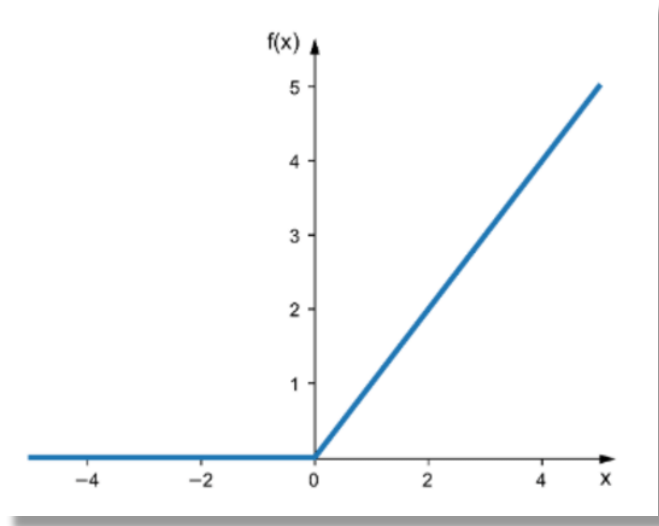


Figure II.8 ReLU function [33]

II.2.1.2.3. Tanh function:

Tanh (Hyperbolic Tangent) function takes any input value and transform it between -1 and 1 [34].

This function is defined as:

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

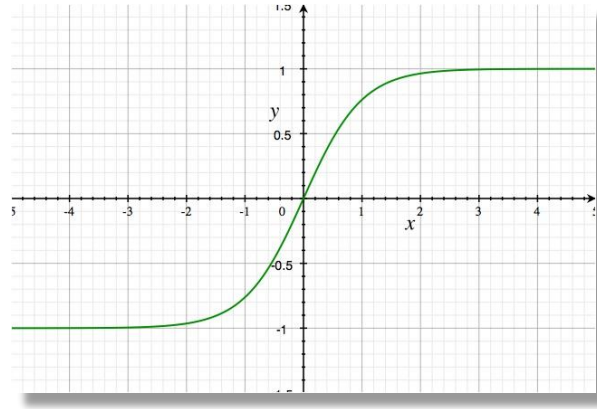


Figure II.9 Tanh function [34]

II.2.1.2.4. Softmax function:

The Softmax function is typically applied to the output layer of a neural network. This activation function transform the output of a neural network into a probability distribution over a set of discrete outcomes.

$$\text{softmax}(z)_i = \sigma(z)_i = \frac{e^{z_i}}{\sum_{k=1}^K e^{z_k}}$$

Given a vector of real valued inputs, $z = (z_1, z_2, \dots, z_K) \in R$ the softmax function computes a vector of k probabilities in the range of [0 and 1] that sums to 1 [35].

II.3. Conclusion

In this chapter, we talked about Machine learning and the different types of it and how each type work and then we defined Deep learning and the artificial neural network as well as some of the most used and popular activation functions. In the next chapter we will mention the popular architecture used in deep learning for predicting diabetes.

Chapter III: Diabetes prediction

There are various approaches in deep learning for diabetes prediction. These approaches use large amounts of data to train the models and then be used to predict the likelihood of a patient having diabetes.

In this chapter, we will talk about the most used approaches in predicting diabetes.

III.1.Convolutional Neural Networks (CNNs):

CNNs are neural network architectures that are powerful in solving complex image recognition and classification tasks. They have the same characteristics of regular neural networks [36].

CNNs were first developed and used around the 1980s at that time the CNN could recognize handwritten digits and it was helpful mostly in the postal sectors such as recognizing postal codes [53].

The CNN architecture is more specific than the other neural networks.

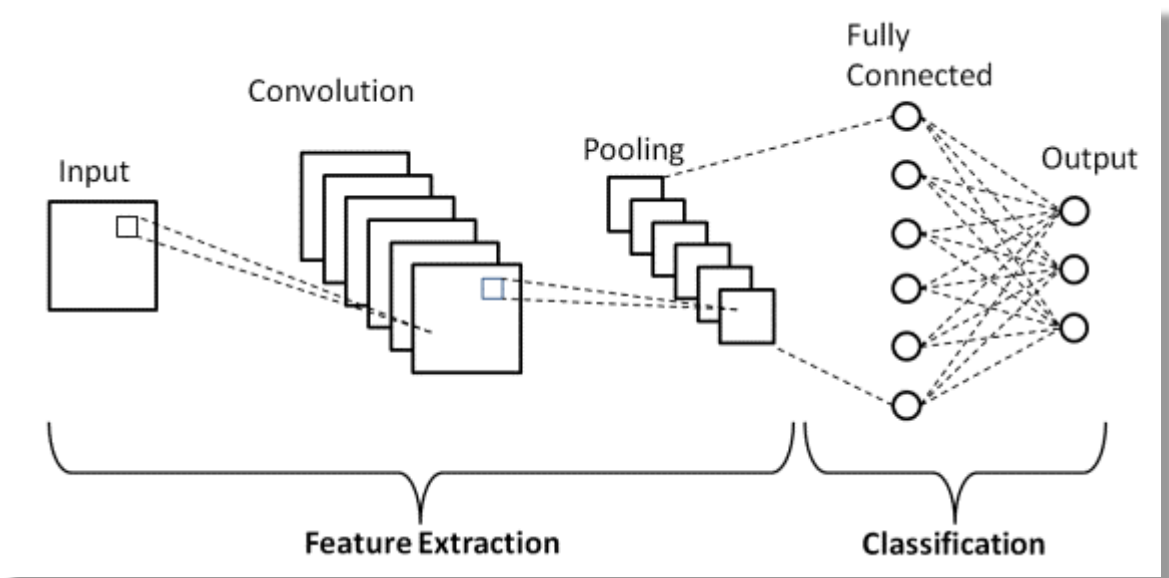


Figure III.1 Convolutional Neural Network Architecture [37]

From Figure III.1 we can see that The CNN is a combination of two basic building blocks:

- **The Convolution Block:** Consists of the Convolution layer and the Pooling layer. This layer forms the essential component of Feature Extraction [38].
- **The Fully Connected Block:** Consists of fully connected simple neural network architecture. The Classification is made based of the input that was given from the first block the convolutional block [38].

There are three types of layers that make up the CNN which are the convolutional layers, pooling layers and fully connected layers.

- **The Convolutional layer:** This layer is the key component of convolutional neural network and is always at least their first layer; the purpose of this layer is to detect the set of features in the images. This is done by using a filter and calculating the convolution product between the input and the filter [39].

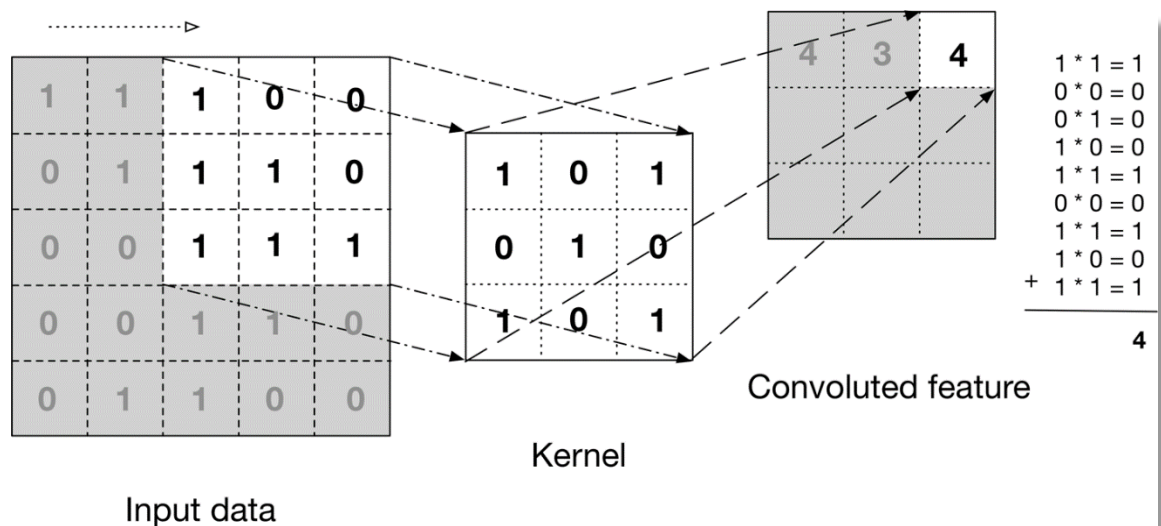


Figure III.2 Convoluted Feature [54]

Figure III.2 shows how the convolution product is calculated. We take Filter/Kernel (3×3 matrix) and apply it to the input data to get the convoluted feature; this output then is passed on to the next layer.

- **The Pooling layer:** In most cases, the convolutional layer is followed by a Pooling layer. This layer is responsible of reducing the size of the output that was passed by the convolution layer. The pooling layer reduce the dimensions of the output, this leads to a decrease of computational power required to process the data [36].

There are two types of pooling: **average pooling** and **max pooling**.

In max pooling, we take the max value among all the values in dimension of for example (2×2 matrix) this means that the values that are taken represent the features that are required to be known and getting rid of the unwanted information [38].

In average pooling, the sum of the values is calculated then divided by the total number existed in the dimension of for example (2×2 matrix) [38].

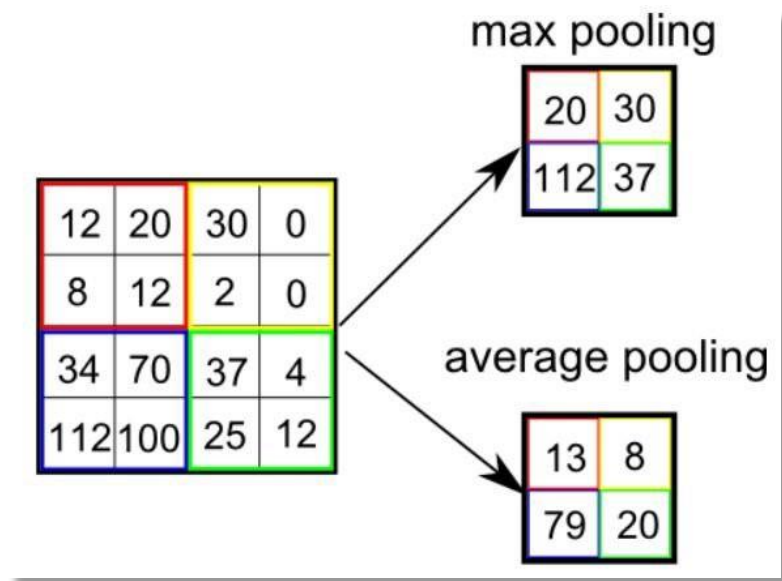


Figure III.3 Example of Max pooling and Average pooling [54]

After the feature extraction is done by the convolution block the result will be a pooled feature map and before passing the output to the fully connected layer, there is a step called **flattening** which is converting the feature map to a single dimensional vector [40].

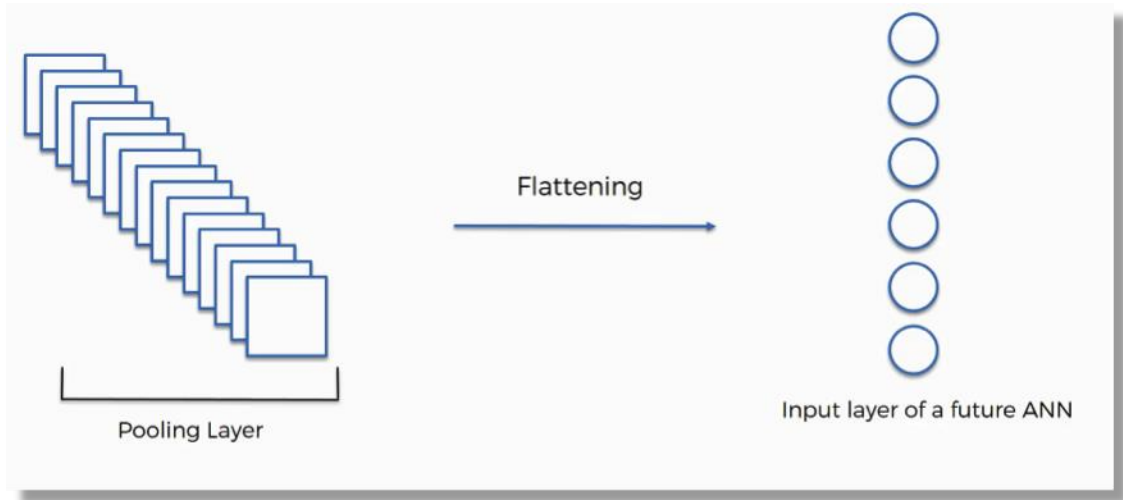


Figure III.4 Flattening [40]

- **The Fully connected layer:** Always the last layer of a neural network convolutional or not. This layer receives an input of single dimensional vector from the convolution block and then performs the classification task of the output according to the categories [38][39].

III.2.Recurrent Neural Networks (RNNs):

Recurrent neural network is a deep learning algorithm that uses sequential data or time series data. RNN works on the principle of saving the output of a particular layer and feeding this back to the input in order to predict the output of the layer [41][42].

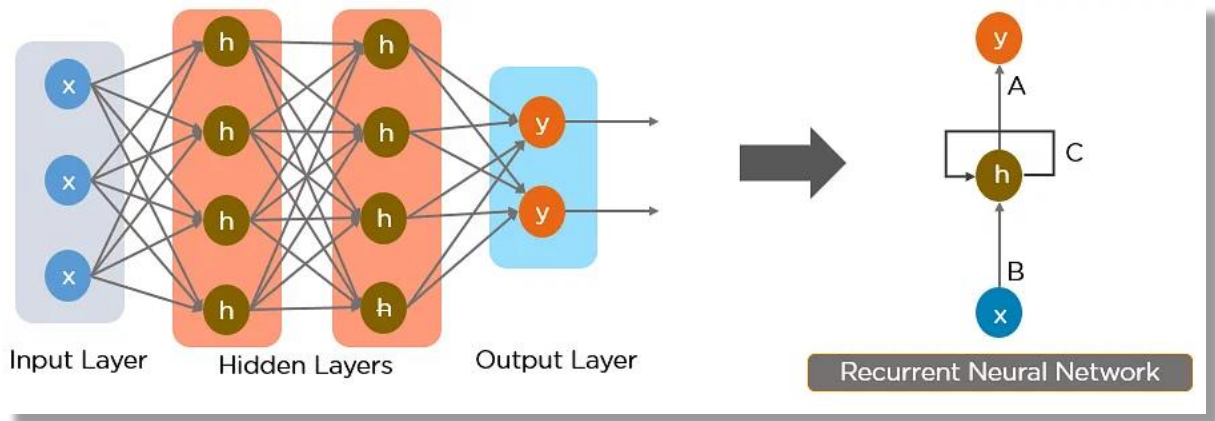


Figure III.5 Artificial neural network to Recurrent neural network [42]

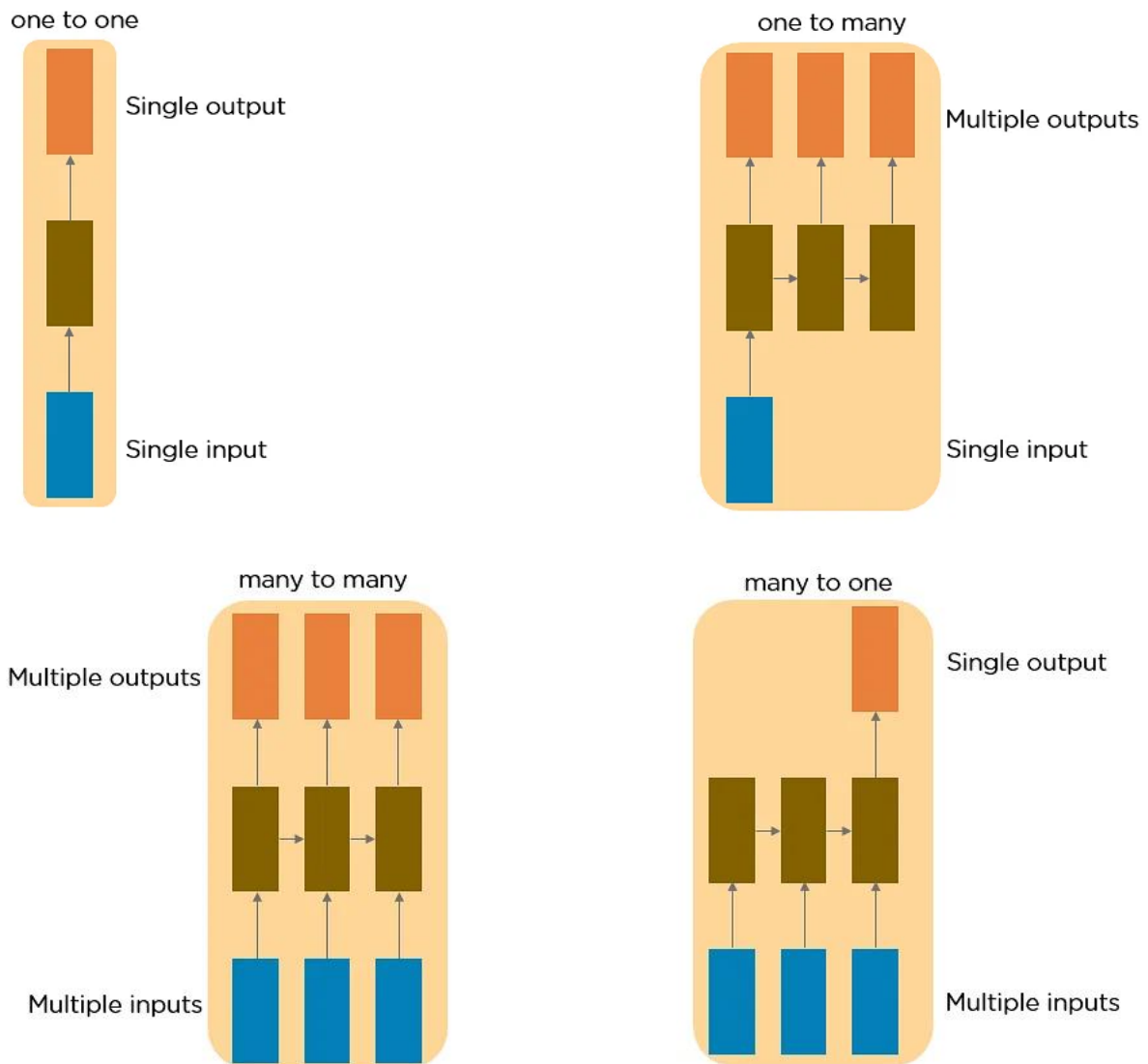
RNNs use backpropagation through time algorithm to determine the gradients. The difference between this and the traditional backpropagation is that backpropagation through time sums errors at each time step as the layers share the parameters across each other. This process lead to RNNs running into two problems, known as exploding gradients and vanishing gradients. The gradient is the slope of the loss function along the error curve [42].

Exploding gradients happens when the gradient is too large; creating an unstable model and the model weights will grow too large as well [42].

Vanishing gradients happens when the gradient is too small and it continues to become smaller, this changing in the gradient will change the weight parameters until they become insignificant, after that, the algorithm is no longer learning [42].

III.2.1. Types of Recurrent neural networks

RNNs have four types [42]



III.3. Long Short-Term Memory (LSTM):

LSTM are an extension of recurrent neural networks mainly introduced to handle situation where the recurrent neural networks fail. LSTM solve the problem of RNNs, which is the vanishing, and exploding gradients [43].

LSTM network architecture consists of three parts.

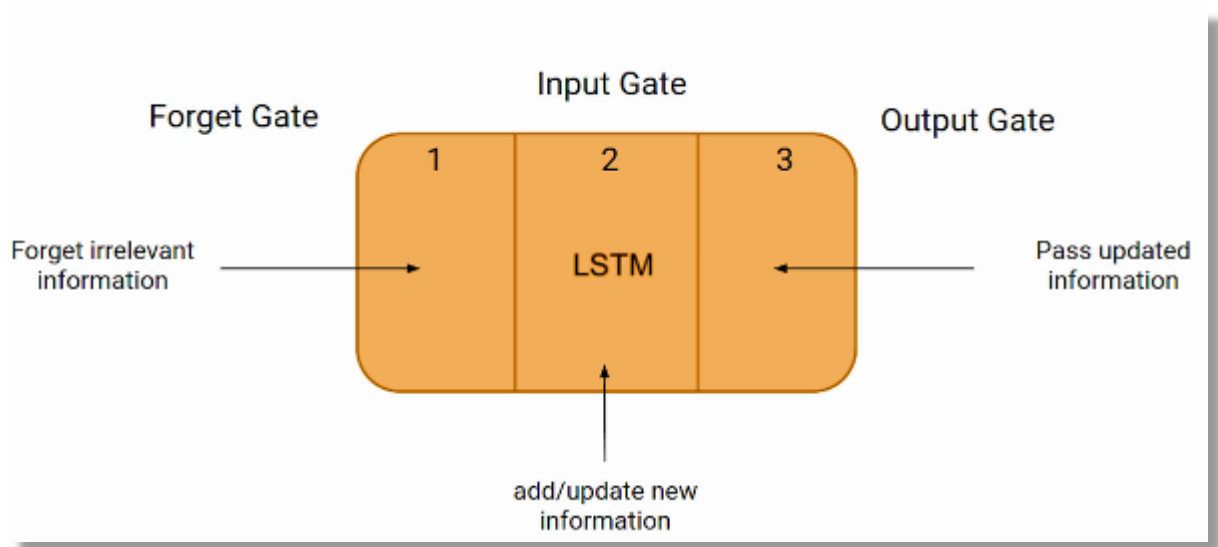


Figure III.6 LSTM network architecture [55]

The three parts are known as Gates.

The first part “Forget Gate” chooses whether the information coming from the previous timestamp is to be remembered or is irrelevant and can be forgotten.

The second part “Input Gate” tries to learn new information from the input.

The third part “Output Gate” the updated information is passed from the current timestamp to the next timestamp.

One cycle of LSTM is considered a single time step [43].

III.4. Deep Belief Networks (DBNs):

Deep belief network is an unsupervised probabilistic algorithm and it is composed of multi-layer of stochastic latent variables. Latent variables are called feature detectors or hidden units and they are binary [44].

The top two layers of DBN are undirected while the lower layers have directed connections from layers above. Each hidden layer of each sub network serve as the visible layer for the next layer [44].

Deep belief networks can be created by stacking RBMs and fine-tuning the resulting deep network via gradient descent and backpropagation [45].

Restricted Boltzman Machines (RBMs) is a type of generative stochastic artificial neural network that can learn a probability distribution from its inputs [45].

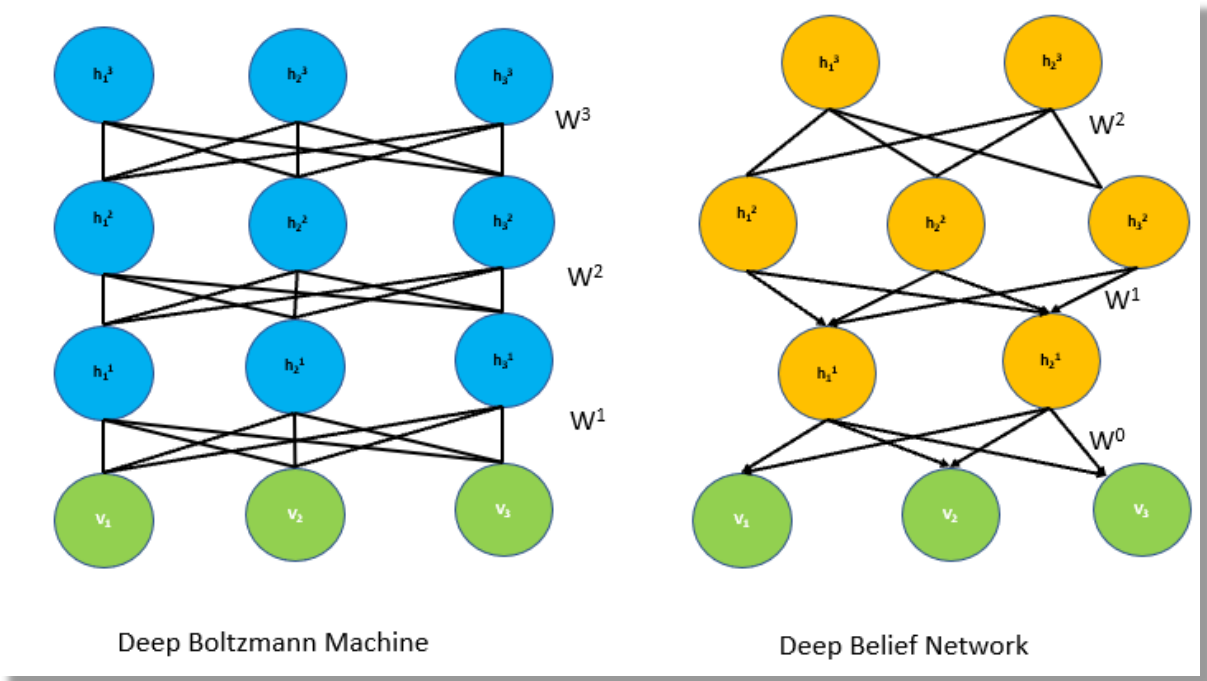


Figure III.7 Deep Belief networks architecture [56]

III.5.Related Works:

After talking about the most used approaches in deep learning, we will now list some related works in the diabetes prediction field using these approaches and we will see how they have been used and the results that was achieved using them.

- **Classification of diabetic and normal fundus images using new deep learning method**

In 2018, Mehdi Torabian ESFAHANI et al [46] used convolutional neural network and they used the architecture of ResNet for classifying eye fundus images into two groups of normal and diabetic images. The dataset was provided by The Kaggle website and it contains 35000 eye fundus images of both healthy and diabetic people.

They did some preprocessing before training and testing to have better quality images by using Gaussian blur and weighted addition as well as applying image normalization to have all images with the size of (512×512). The result that they achieved was 85% accuracy and 86% sensitivity.

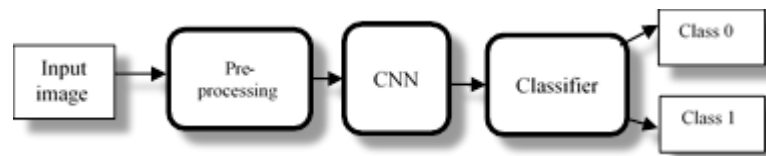


Figure III.8 Block diagram of the proposed method

- **Diabetes detection using deep learning algorithms**

In 2018, Swapna G et al [47] used CNN and CNN-LSTM network and heart rate signals and they achieved a maximum accuracy of 95.1% then they added SVM to the proposed architecture and they achieved an improvement of 0.6% from the last result given them accuracy of 95.7%. They used Pan and Tompkins algorithm to derive the heart rate time series data from ECG signals this real time algorithm can effectively detect QRS complexes in a ECG signal based on its morphological features like slope, amplitude and width. It involves processes like digital bandpass filtering (to reduce false detections due to noise) and thresholding operations (to increase detection sensitivity). The ECG signal is sampled at 500 Hz.

They used 71 dataset and each one contains 1000 sample of electrocardiograms (ECG). The input is passed without any further preprocessing.

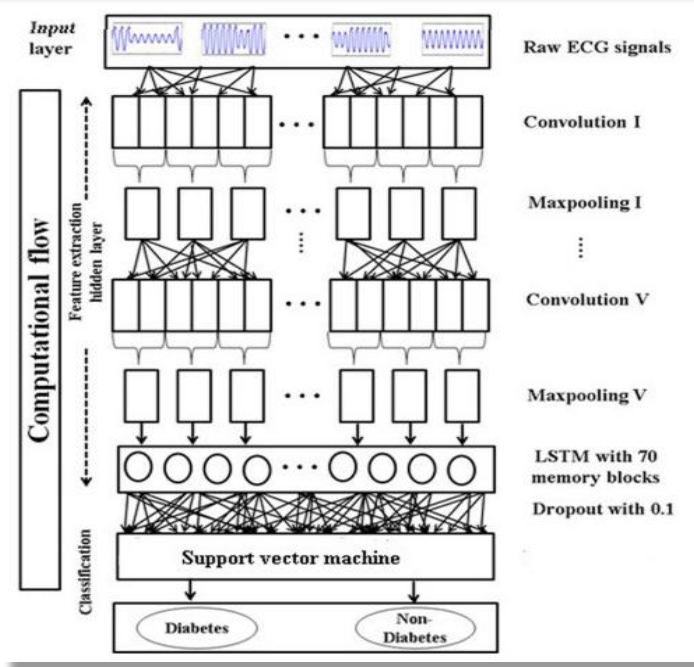


Figure III.9 Architecture of proposed system for detection of diabetes

- **A PREDICTION OF DIABETES POSSIBILITY WITH DEEP LEARNING: LSTM AND ESA HYBRID MODEL**

In 2022, s.m.hasni et al [48] used a combination between long short-term memory networks (LSTM) and convolutional neural networks (CNN) through an explicit semantic analysis (ESA) this model was evaluated using the dataset of Pima Indians Diabetes. They proposed a three steps method for the detection of diabetes. The first step they pass the features obtained from the samples in the dataset through the designed ESA. Then the features obtained from ESA were given as input to LSTM and Finally the samples in the dataset were classified using the softmax classifier. This proposed method reached to 86.4% accuracy.

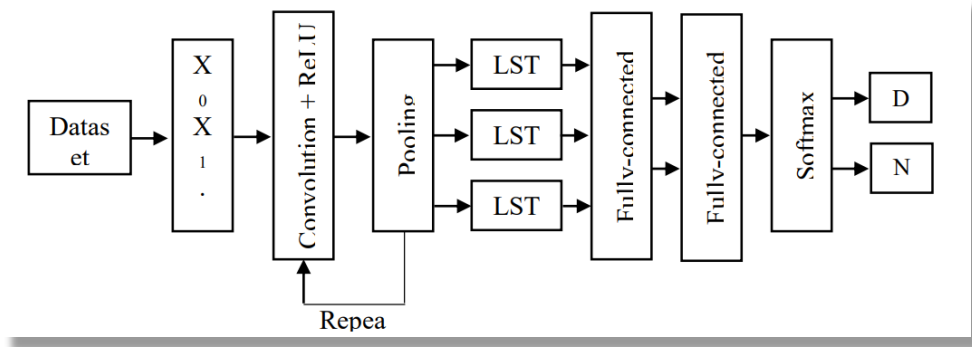


Figure III.10 Proposed method process (D: Diabetic; ND: Not Diabetic)

- **An Effective Approach for Detecting Diabetes using Deep Learning Techniques based on Convolutional LSTM Networks**

In 2021, P.Bharath et al [49] proposed an approach to enhance diabetes prediction using deep learning techniques. They developed a diabetes classification model based on the convolutional long short-term memory (CLSTM) and they used the Pima Indians Diabetes Database (PIDD). They pre-processed the dataset and they selected the essential features. They used two models one with the traditional long short-term memory (TLSTM) and the other with CLSTM. The proposed models achieved an accuracy of 92.5% for TLSTM and 96.8% for CLSTM (test set 10%) and accuracy of 93.7% for TLSTM and 95.6% for CLSTM (test set 20%).

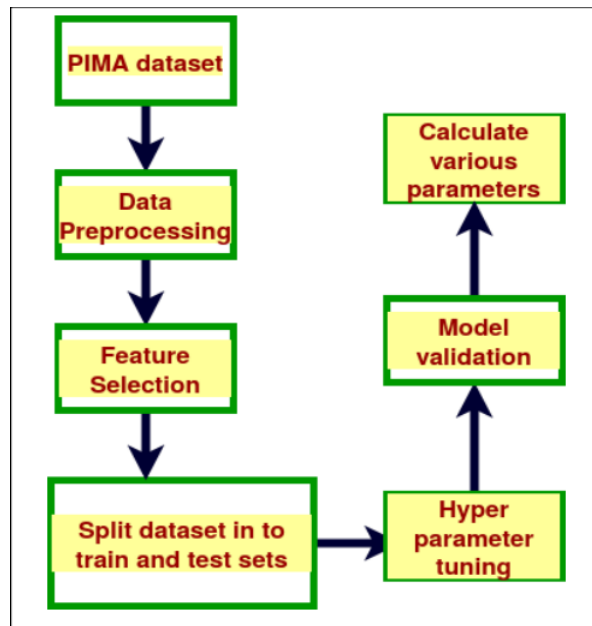


Figure III.11 Proposed Model for Diabetes Prediction

- **CNN Based Diabetic Retinopathy Status Prediction Using Fundus Images**

In 2020, Md. Ahsan et al [50] their computational model consists of a feature extraction phase and a classification phase. In feature extraction phase, they extracted the most appropriate features from digital fundus images by blood vessels and micro aneurysms detection. They used a retinal dataset from kaggle community provided by EyePACS that contains 88,702 sets of high-resolution retina images. The pre-processing converts the images into gray scale and remove the noise from the converted fundus images using adaptive median filter. They used the VGGnet architecture and they were able to achieve an accuracy of 95.41%.

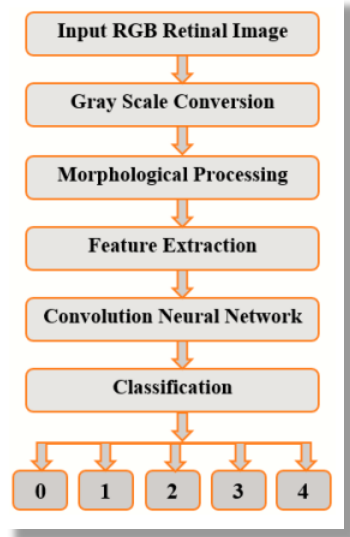


Figure III.12 Block diagram of proposed methodology

Table III.1 Comparison of related works on diabetes prediction using deep learning techniques

Study	Year	Method	Dataset	Preprocessing	Accuracy
Classification of diabetic and normal fundus images using new deep learning method	2018	CNN (ResNet)	Kaggle dataset 35,000 fundus images	Gaussian blur, weighted addition, image normalization(512 ×512)	85%
Diabetes detection using deep learning algorithms	2018	CNN, CNN-LSTM network, SVM	71 ECG datasets each with 1000 samples	Pan and Tompkins algorithm for heart rate time series, digital bandpass filtering, thresholding operations	95.7%
A PREDICTION OF DIABETES POSSIBILITY WITH DEEP LEARNING: LSTM AND ESA HYBRID MODEL	2022	LSTM, CNN, ESA hybrid model	Pima Indians Diabetes Dataset	Explicit Semantic Analysis (ESA)	86.4%

An Effective Approach for Detecting Diabetes using Deep Learning Techniques based on Convolutional LSTM Networks	2021	CLSTM, TLSTM	Pima Indians Diabetes Database	Feature selection	Test set 10% { TLSTM:92.5%, CLSTM:96.8% } Test set 20% { TLSTM:93.7%, CLSTM:95.6% }
CNN Based Diabetic Retinopathy Status Prediction Using Fundus Images	2020	CNN (VGGnet)	EyePACS dataset 88,702 Retina images	Gray scale conversion, adaptive median filter	95.41%

III.6. Conclusion

Diabetes prediction is a complex task that requires the analysis of various risk factors. Both machine and deep learning algorithms have been used to help developing predictive models that can accurately identify individuals at risk of developing diabetes. These models have shown promising results, with high accuracy results that helps in detecting diabetes at early stages. Despite having promising results, further research is needed to develop more models that take into account a wide range of factors that plays a role in developing this chronic disease.

Chapter IV: Design of the approach

The progression made by Artificial Intelligence (AI) in various fields has led to these algorithms to exceed the human potential and it will be more needed in the near future because of the increases that happened to the data and the resources in all fields that requires fast and accurate solutions. As we mentioned before that diabetes will increase and target more people in the near future, this was an opportunity to develop a deep learning model to predict diabetes and help in this field. In this chapter, we present the system design, the dataset that was used, the preprocessing phase, and the proposed LSTM architecture.

IV.1. Global architecture of the system

In order to build a deep learning model that will predict diabetes, the process begins with collecting the dataset then applying the needed preprocessing to improve the quality of the dataset then it will be fed to the deep learning model to do the remaining steps in order to predict new unseen data.

Figure IV.1 represent our system and the steps that it will be following.

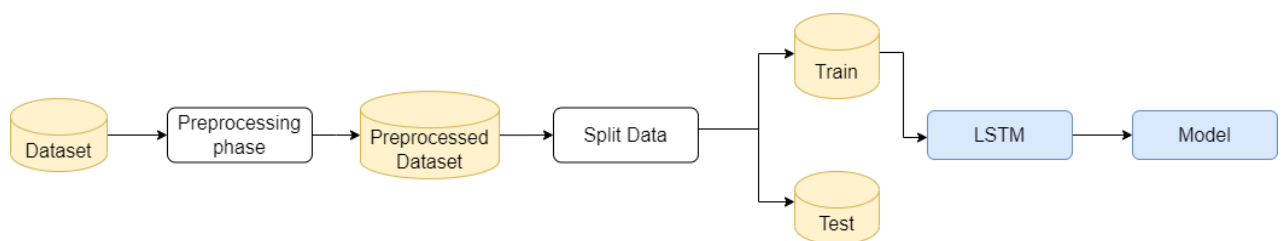


Figure IV.1 General Architecture

IV.1.1. Dataset

IV.1.1.1. Dataset description: The dataset that have been used in this study is available on Kaggle website [51].

-This dataset contains 2000 samples of people with and without diabetes [52].

-This dataset have 8 features (Pregnancies, Glucose, Blood Pressure, Skin Thickness, Insulin, BMI, Diabetes Pedigree Function, Age) and one outcome these features helps in determining if the person have diabetes or not.

IV.1.1.2. Dataset structure: In this dataset there are 2 classes which are:

0: The person does not have diabetes

1: The person does have diabetes

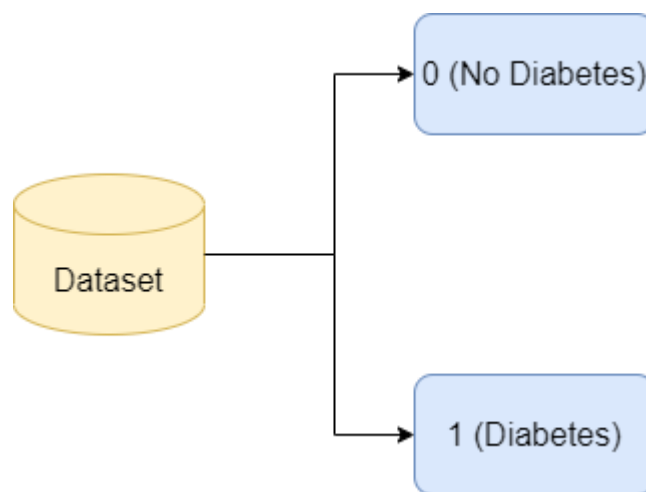


Figure IV.2 Dataset Structure

IV.1.2. Preprocessing

Before providing the dataset to the model it must first go through some preprocessing phase to produce a better quality dataset for the model, we will mention the preprocessing steps.

IV.1.2.1. Data Filtering

In this step we check in the dataset if there is any rows that has zero or null values in specific features (Glucose, Blood Pressure, Skin Thickness, Insulin, BMI) and if they have it we remove them from the dataset.

IV.1.2.2. Data Scaling

The data is scaled to ensure that the input features are beneficial for the training and performance of the model and the scaling normalize the values of the data so it brings them to a similar range and prevent any single feature from dominating the learning process due to its large magnitude.

IV.1.2.3. Data Reshaping

The input features are reshaped into three dimensional shape resulting them to have the shape (samples, time steps, features) so they will be compatible with the LSTM layers in the model.

IV.1.3. Splitting dataset

In this process, the dataset is divided into a training, validation, and testing subsets. Our dataset is divided into: 80% for training and 20% for testing the model and our 80% subset of training is divided as well for 20% validation.

IV.1.4. LSTM

After preprocessing the dataset and splitting it to two subsets training and testing, comes the LSTM where the model will be trained to make diabetes prediction. We mentioned before the LSTM overcomes the RNNs problem, which is the vanishing, and exploding gradients. The architecture of LSTM helps in tasks involving sequential data analysis.

IV.1.4.1. Model parameters

Every Deep learning model have parameters that needs to be specified for it to be trained, we will mention some parameters.

IV.1.4.1.1. Number of Units/Neurons: The number of units or neurons in an LSTM layer determines the capacity of the layer to learn and represent complex patterns in the data.

IV.1.4.1.2. Activation Functions: The activation function defines the output of a neuron given its input, the common activation functions used in LSTM layers include **tanh** and **sigmoid** functions.

IV.1.4.1.3. Learning rate: The learning rate determines the step size at which the model parameters are updated during the training process and it is a constant number usually very small.

IV.1.4.1.4. Optimization Algorithm: The optimization algorithm defines how the model's parameters are updated during the training process.

Popular optimization algorithms used with LSTM models include: Adam, RMSProp, SGD, etc.

IV.1.4.1.5. Epochs: The number of epochs defines how many times the entire dataset is passed forward and backward through the neural network during training.

IV.1.4.1.6. Batch Size: The batch size is the number of training examples used in a single iteration of the training algorithm before updating the model parameters.

IV.1.4.1.7. Loss Function: a method of evaluating how well the models is performing. In the case of predicting diabetes where the outcome is two classes the common loss function used is `binary_crossentropy`.

IV.1.5. Prediction

After the training phase, the model is ready to make predictions of new unseen data and for that the testing subset will be used to test our model.

IV.1.6. Model evaluation

Evaluating deep learning model is measuring the performance of the model and there are many different types of evaluation metrics. The choice of evaluation metrics depends on the nature of the problem. In our model, we will use 4 evaluation metrics: Accuracy, Precision, Recall, F1-Score [66].

To calculate these metrics, the values of true positives (TP), true negatives (TN), false positives (FP) and false negatives (FN) will be used.

$$ACCURACY = \frac{TP + TN}{TP + TN + FP + FN}$$

$$PRECISION = \frac{TP}{TP + FP}$$

$$RECALL = \frac{TP}{TP + FN}$$

$$F1 - SCORE = \frac{2 \times (PRECISION \times RECALL)}{(PRECISION + RECALL)}$$

IV.2. Conclusion

In this chapter, we presented our model architecture and described the dataset and explained step by step the preprocessing phase and talked about LSTM and the essential parameters for a deep learning model. In the following chapter, we will go over the implementation of the model and explaining every line of code and then we will discuss the achieved results of the proposed model.

Chapter V: Implementation and results

In this chapter, we will present the code for each step we proposed for our system, then we will present the result obtained from our model, and finally we will compare our model results with the results of the related works.

V.1. Implementation frameworks, tools and libraries

- **Python**

Python is a high level programming language with dynamic semantics. It is a general-purpose language, which means it can be used to create variety of different programs. It is easy to learn due to its simple syntax. [\[57\]](#)

- **Google Colab**

Colaboratory is a product from Google research. It allows writing and executing python code through the browser. Colab is a hosted jupyter notebook service that requires no setup to use and it provides free of charge computing resources. [\[58\]](#)

- **Kaggle**

Kaggle is an online community platform for data scientists and machine learners where you find and publish datasets or have competitions to solve challenges. [\[51\]](#)

- **Tensorflow**

Tensorflow is open source library for machine learning. It can be used to develop and train machine or deep learning models. [\[59\]](#)

- **Keras**

Keras is a high-level neural network API written in python. It is used to make the implementation of neural networks easy. [\[60\]](#)

- **Matplotlib**

Matplotlib is a plotting library for the Python programming language and its numerical mathematics extension NumPy. [\[61\]](#)

- **Scikit-learn**

Scikit-learn is an open source data analysis library. [\[62\]](#)

- **Pandas**

Pandas is a software library written for the Python programming language for data manipulation and analysis. [\[63\]](#)

- **NumPy**

NumPy is the fundamental package for scientific computing in Python. [\[64\]](#)

- **Seaborn**

Seaborn is a Python data visualization library based on matplotlib. [\[65\]](#)

V.2. Implementation phases

In order to implement our model first we need to import certain libraries that are essential and will help in building the model.

```
import pandas as pd
import tensorflow as tf
import numpy as np
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Dense, LSTM, Dropout
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler
from sklearn.metrics import confusion_matrix
import matplotlib.pyplot as plt
from sklearn import metrics
from sklearn.metrics import classification_report, accuracy_score, precision_score, recall_score, f1_score
import seaborn as sns
```

Figure V.1 Model Libraries

After importing these libraries, we will load the dataset and preprocess the data before feeding it to the LSTM model.

V.2.1. Loading dataset

In order to load the dataset we will use **pandas** as it will help in data analysis and manipulation.

```
dataset = pd.read_csv('/content/diabetes-dataset.csv')
```

Figure V.2 Loading dataset

The dataset that we used is **csv file** and in **pandas** the function **read_csv** takes the path of the dataset and read it and we store the data in a variable called **dataset**.

The imported dataset looks like:

	Pregnancies	Glucose	BloodPressure	SkinThickness	Insulin	BMI	DiabetesPedigreeFunction	Age	Outcome
0	2	138	62	35	0	33.6	0.127	47	1
1	0	84	82	31	125	38.2	0.233	23	0
2	0	145	0	0	0	44.2	0.630	31	1
3	0	135	68	42	250	42.3	0.365	24	1
4	1	139	62	41	480	40.7	0.536	21	0

Figure V.3 Dataset

The dataset contains 2000 samples classed with two outcomes (0 and 1). To see the number of samples per class the function **value_counts** is used.

```
outcome_counts = dataset['Outcome'].value_counts()
print(outcome_counts)
```

Figure V.4 Code for number of samples per class

The result of this line of code is:

Table V.1 Number of samples per class

Outcome	Number of samples
0	1316
1	684

V.2.2. Preprocessing

V.2.2.1. Data Filtering

Before doing any filtering to the data, we need to check first if the dataset contains any zero or null values in the features.

```
dataset.isnull().sum()
Pregnancies      0
Glucose           0
BloodPressure    0
SkinThickness    0
Insulin           0
BMI              0
DiabetesPedigreeFunction  0
Age              0
Outcome          0
dtype: int64
```

Figure V.5 Checking null values

From the figure V.5, we can see that the dataset does not have any null values. However if we use a dataset description we can have a better data analysis to the dataset.

```
dataset.describe()
```

	Pregnancies	Glucose	BloodPressure	SkinThickness	Insulin	BMI	DiabetesPedigreeFunction	Age	Outcome
count	2000.000000	2000.000000	2000.000000	2000.000000	2000.000000	2000.000000	2000.000000	2000.000000	2000.000000
mean	3.703500	121.182500	69.145500	20.935000	80.254000	32.193000	0.470930	33.090500	0.342000
std	3.306063	32.068636	19.188315	16.103243	111.180534	8.149901	0.323553	11.786423	0.474498
min	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.078000	21.000000	0.000000
25%	1.000000	99.000000	63.500000	0.000000	0.000000	27.375000	0.244000	24.000000	0.000000
50%	3.000000	117.000000	72.000000	23.000000	40.000000	32.300000	0.376000	29.000000	0.000000
75%	6.000000	141.000000	80.000000	32.000000	130.000000	36.800000	0.624000	40.000000	1.000000
max	17.000000	199.000000	122.000000	110.000000	744.000000	80.600000	2.420000	81.000000	1.000000

Figure V.6 Dataset description

From the figure V.6, we can see that some of the features have a **min value of 0** which in real life can not happen beside the feature of **pregnancies**.

Therefore, the features that are going to be filtered from the dataset are **Glucose, Blood Pressure, Skin Thickness, Insulin, and BMI**.

```
diabetes_dataset = dataset[(dataset.Glucose != 0) & (dataset.BloodPressure != 0) & (dataset.SkinThickness != 0) & (dataset.Insulin != 0) & (dataset.BMI != 0)]
```

Figure V.7 Data Filtering

After executing this line of code, we have a better dataset that does not contain any zero or null values and we store it in a new variable called **diabetes_dataset**.

This new filtered dataset contains 1035 samples.

Table V.2 Number of samples per class

Outcome	Number of samples
0	698
1	337

V.2.2.2. Selection of input features and output target

In this dataset we have 8 features: Pregnancies, Glucose, Blood Pressure, Skin Thickness, Insulin, BMI, Diabetes Pedigree Function, Age, and one target: Outcome.

```
X = diabetes_dataset.drop('Outcome', axis='columns').values
y = diabetes_dataset['Outcome'].values
```

Figure V.8 Selection of features and target

We assign the first variable by dropping the outcome column, as it is a target not a feature and assign the second variable by choosing from the dataset only the outcome column.

V.2.2.3. Splitting dataset

After importing **train_test_split** from **sklearn.model_selection** we use it to split the dataset to training and testing subsets.

```
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)
```

Figure V.9 Splitting dataset to training and testing

We used **random_state** to control the shuffling of data. If we do not set it then every time we get a different test and train sets and this will affect the performance of the model.

V.2.2.4. Data Scaling

As we said in the last chapter that when the data is scaled within a similar range it helps the model performance and prevent any features from dominating.

```
scaler = StandardScaler()  
X_train = scaler.fit_transform(X_train)  
X_test = scaler.transform(X_test)
```

Figure V.10 Data Scaling

We can take an example before and after scaling.

```
X_test[25]  
array([ 1.    , 125.   , 70.    , 24.    , 110.   , 24.3   , 0.221,  
       25.   ])
```

Figure V.11 Before Scaling

After applying the scaling code presented in figure V.10, we have a new scaled data.

```
X_test[25]  
array([-0.68989501,  0.07972054, -0.10047743, -0.50086787, -0.38402094,  
       -1.28764024, -0.90280787, -0.57204759])
```

Figure V.12 After Scaling

From figure V.11 and V.12 we can see that after scaling the data, the values of the features are scaled down to a similar range.

V.2.2.5. Data Reshaping

```
X_train.shape  
(828, 8)
```

Figure V.13 Before reshaping

The shape of the data is in 2D shape and it represent (number of samples, number of features).

The reshaping of the data is necessary because the LSTM expects a 3D input shape with the dimensions (number of samples, number of time steps, number of features).

```
X_train = X_train.reshape((X_train.shape[0], 1, X_train.shape[1]))
X_test = X_test.reshape((X_test.shape[0], 1, X_test.shape[1]))
```

Figure V.14 Data Reshaping

After applying the reshaping, the new shape of the data will be:

```
X_train.shape
(828, 1, 8)
```

Figure V.15 After reshaping

V.2.3. LSTM Model

After loading the dataset, preprocessing it and splitting it to training and testing sets we are ready to start building a deep learning model.

The model that is going to be used is the sequential model.

```
model = Sequential()
```

Figure V.16 Sequential model

Our model architecture

```
model.add(LSTM(128, input_shape=(1, X_train.shape[2]), activation='tanh', return_sequences=True))
model.add(Dropout(0.2))
model.add(LSTM(64, activation='tanh'))
model.add(Dense(1, activation='sigmoid'))
```

Figure V.17 LSTM model architecture

The model is a 4 layers model: 2 LSTM layers, 1 dropout layer, and 1 dense layer.

```
optimizer = tf.keras.optimizers.Adam(learning_rate=0.03)
```

Figure V.18 Optimizer

We have used the Adam optimizer with a learning rate = 0.03.

To compile our model we used this code

```
model.compile(loss='binary_crossentropy', optimizer=optimizer, metrics=['accuracy'])
```

Figure V.19 Compile

The loss function is a `binary_crossentropy` because we have a binary classification problem where the task is to predict whether an individual has diabetes or not.

Our model code will look like this:

```
# Create the model
model = Sequential()
model.add(LSTM(128, input_shape=(1, X_train.shape[2]), activation='tanh', return_sequences=True))
model.add(Dropout(0.2))
model.add(LSTM(64, activation='tanh'))
model.add(Dense(1, activation='sigmoid'))

# Compile the model
optimizer = tf.keras.optimizers.Adam(learning_rate=0.03)
model.compile(loss='binary_crossentropy', optimizer=optimizer, metrics=['accuracy'])

history = model.fit(X_train, y_train, epochs=50, batch_size=32, verbose=2, validation_split = 0.2)
```

Figure V.20 LSTM Model

To have a summary about the model we use the summary function.

```
model.summary()
```

```
Model: "sequential"
```

Layer (type)	Output Shape	Param #
lstm (LSTM)	(None, 1, 128)	70144
dropout (Dropout)	(None, 1, 128)	0
lstm_1 (LSTM)	(None, 64)	49408
dense (Dense)	(None, 1)	65

```
=====  
Total params: 119,617  
Trainable params: 119,617  
Non-trainable params: 0
```

Figure V.21 Model Summary

Model.fit is a function that is used to train the neural network model it fits the model to the training data and performs the training process.

```
history = model.fit(X_train, y_train, epochs=50, batch_size=32, verbose=2, validation_split = 0.2)
```

Figure V.22 Model Fitting

The evaluation of the model is performed using **model.evaluate** function. This function evaluates the trained model on the specified test data to measure its performance.

Additionally we are using other evaluation metrics such as confusion matrix, accuracy, precision, recall, F1-Score, and the classification report to further analyze the model's performance on the test data.

```
# Evaluate the model
print('=====TRAIN RESULTS=====')
Train_score, Train_acc = model.evaluate(X_train, y_train, verbose=0)

# Print the results
print('Train score:', Train_score)
print('Train accuracy:', Train_acc)

print('=====TEST RESULTS=====')

Test_score, Test_acc = model.evaluate(X_test, y_test, verbose=0)

# Calculate confusion matrix
y_pred = model.predict(X_test)
y_pred = (y_pred > 0.5)
cm = confusion_matrix(y_test, y_pred)

#####
# Evaluate the model
accuracy = accuracy_score(y_test, y_pred)
precision = precision_score(y_test, y_pred)
recall = recall_score(y_test, y_pred)
f1 = f1_score(y_test, y_pred)
classification_report_str = classification_report(y_test, y_pred)
```

Figure V.23 Model evaluation

```

# Print the results
print('Accuracy:', accuracy)
print('Precision:', precision)
print('Recall:', recall)
print('F1-Score:', f1)
print('Classification Report:')
print(classification_report_str)

print('Test score:', Test_score)
print('Test accuracy:', Test_acc)
print('Confusion matrix:')
print(cm)

cm_display = metrics.ConfusionMatrixDisplay(confusion_matrix = cm, display_labels = ['0 No Diabetes', '1 Diabetes'])
cm_display.plot()
plt.show()

```

Figure V.24 Model evaluation

To visualize the performance of the model we use this following code:

```

# Visualize the results
plt.figure(figsize=(12, 4))

# Plot accuracy
plt.subplot(1, 2, 1)
plt.plot(history.history['accuracy'], label='Training Accuracy')
plt.plot(history.history['val_accuracy'], label='Validation Accuracy')
plt.title('Accuracy')
plt.xlabel('Epoch')
plt.ylabel('Accuracy')
plt.legend()

# Plot loss
plt.subplot(1, 2, 2)
plt.plot(history.history['loss'], label='Training Loss')
plt.plot(history.history['val_loss'], label='Validation Loss')
plt.title('Loss')
plt.xlabel('Epoch')
plt.ylabel('Loss')
plt.legend()

plt.tight_layout()
plt.show()

```

Figure V.25 Model visualization

V.3. Results

After talking about the implementation of our model now we will present the achieved results after evaluating the model.

- A plot of training accuracy and validation accuracy

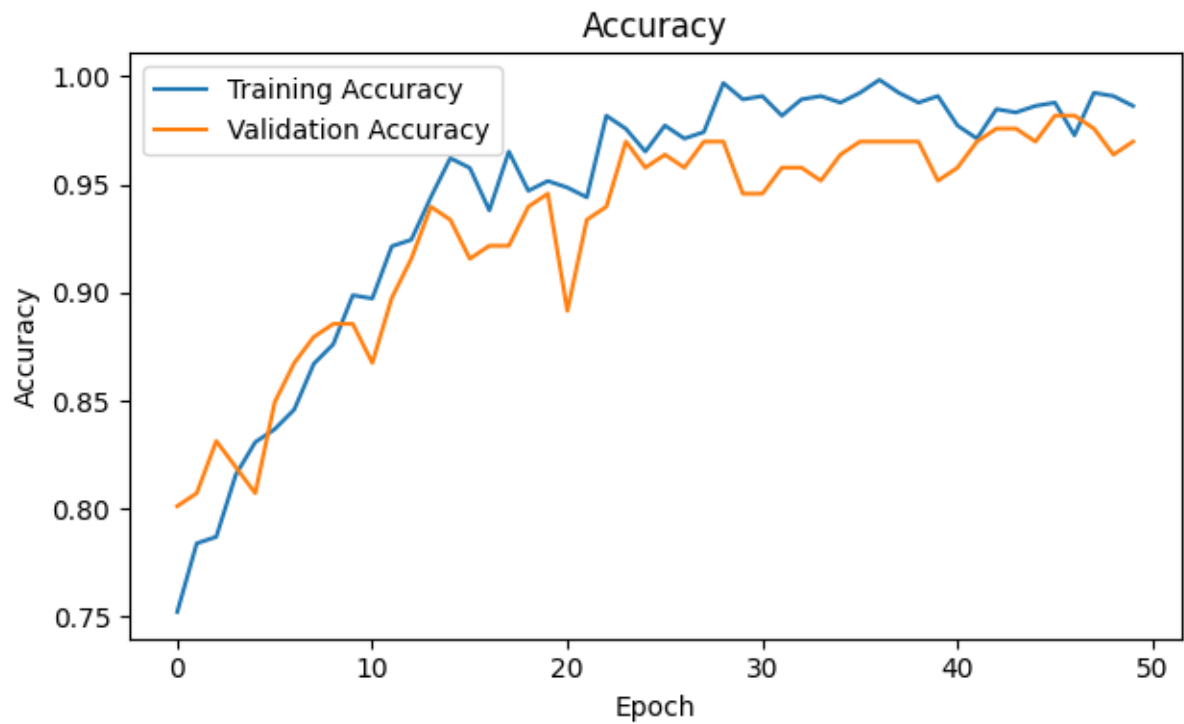


Figure V.26 Model Accuracy

- A plot of training loss and validation loss

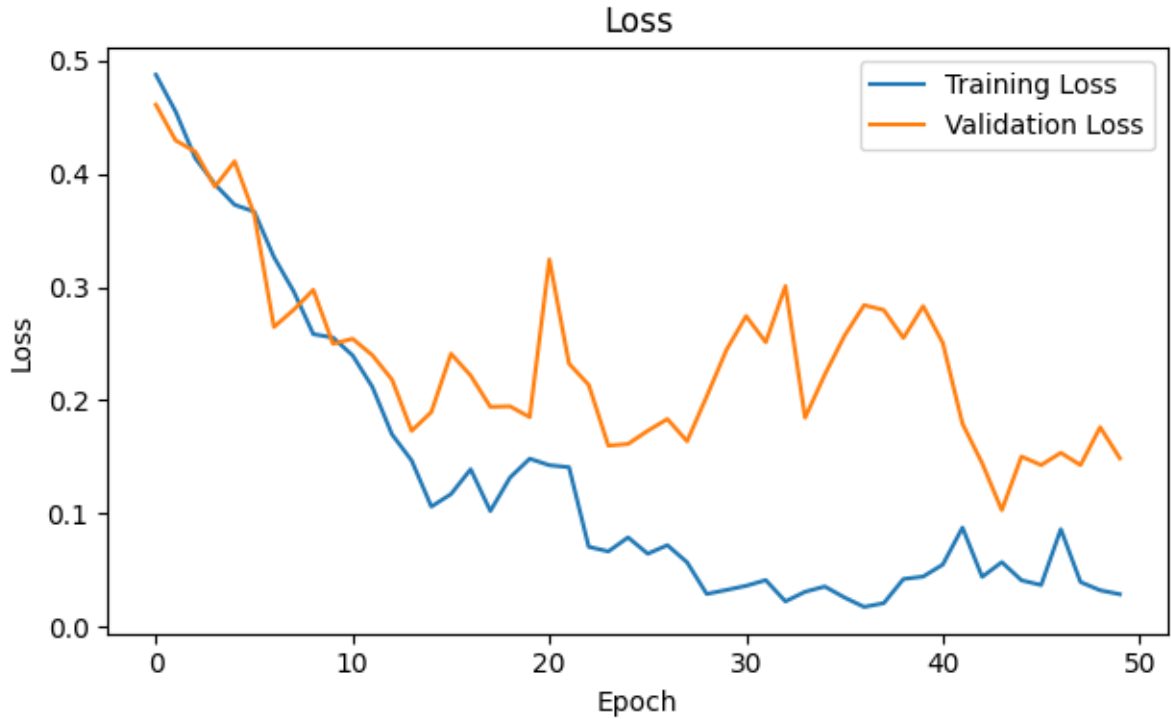


Figure V.27 Model Loss

The plots of the accuracy and the loss show a good fit because:

- The training accuracy and validation accuracy are converging to the same value.
- The training loss and validation loss are decreasing to the same value.

Our model performance are in the table V.3.

```
Epoch 50/50
21/21 - 0s - loss: 0.0288 - accuracy: 0.9864 - val_loss: 0.1488 - val_accuracy: 0.9699
```

Figure V.28 Model Accuracy

Table V.3 Model results

	Accuracy	Validation Accuracy	Loss	Validation Loss
LSTM Model	98.64%	96.99%	2.88%	14.88%

The results achieved after testing our model with the test subset

- **Metric results**

```
=====TEST RESULTS=====
7/7 [=====] - 0s 5ms/step
Accuracy: 0.9903381642512077
Precision: 1.0
Recall: 0.967741935483871
F1-Score: 0.9836065573770492
Classification Report:
      precision    recall  f1-score   support

     0       0.99       1.00       0.99       145
     1       1.00       0.97       0.98        62

 accuracy                0.99       207
 macro avg              0.99       0.98       0.99       207
 weighted avg          0.99       0.99       0.99       207

Test score: 0.16309846937656403
Test accuracy: 0.990338146686554
Confusion matrix:
[[145  0]
 [ 2  60]]
```

Figure V.29 Model testing

From figure V.29, we can tell that our model is performing quite well as it achieved:

- Accuracy of 0.9903 this indicates that the model correctly predicts the class label for 99.03% of the samples in the test set.
- Precision of 1.0 for class 1 and 0.99 for class 0 this indicates that the model has a high ability to correctly identify true positive samples for both classes.
- Recall of 0.9677 for class 1 and 1.0 for class 0 indicating a high ability to capture most of the positive samples.
- F1-Score of 0.9836 indicates that the model has both good precision and recall.

- **Confusion matrix**

The confusion matrix is a table that is used to evaluate the performance of a classification model.

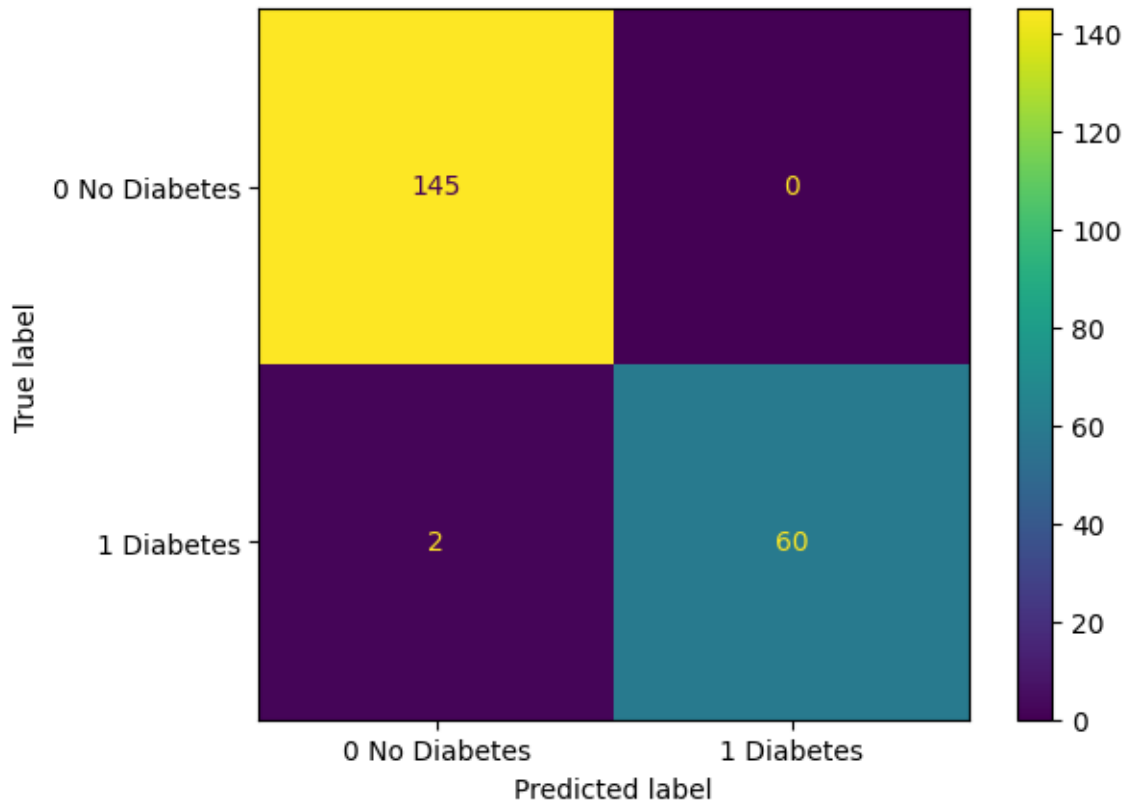


Figure V.30 Confusion matrix

- The model could predict all 145 samples correctly as class 0 No Diabetes.
- The model could predict 60 out of 62 samples as class 1 Diabetes.

V.4. Comparison with other related works

We will compare our model results with other related works results, we will be using model accuracy to compare the results.

Table V.4 Comparison of our model with related works

	Accuracy
Our Model	98.64%
Classification of diabetic and normal fundus images using new deep learning method [46]	85%
Diabetes detection using deep learning algorithms [47]	95.7%
A PREDICTION OF DIABETES POSSIBILITY WITH DEEP LEARNING: LSTM AND ESA HYBRID MODEL [48]	86.4%
An Effective Approach for Detecting Diabetes using Deep Learning Techniques based on Convolutional LSTM Networks [49]	Test set 10% { TLSTM:92.5%, CLSTM:96.8% } Test set 20% { TLSTM:93.7%, CLSTM:95.6% }
CNN Based Diabetic Retinopathy Status Prediction Using Fundus Images [50]	95.41%

V.5. Conclusion

In this chapter, we have presented how the model is implemented, we talked about the results that were achieved by the model, and lastly we compared our model results to other related works results. The achieved results represent a model with a good performance that can help in diabetes prediction.

General conclusion

This work represents a case study and aims to develop a diabetes prediction model using a deep learning approach. To achieve research objectives we proposed a deep learning model with LSTM architecture and it turned out that our proposed model achieved better results than the related works that we mentioned.

Despite that the model achieved excellent results, it opens ways for future works such as model improvements.

References

- [1] Diabetes facts & figures. The International Diabetes Federation (IDF).
<https://idf.org/aboutdiabetes/what-is-diabetes/facts-figures.html>
- [2] Algeria diabetes report 2000 — 2045. International Diabetes Federation.
<https://diabetesatlas.org/data/en/country/3/dz.html>
- [3] What is Diabetes? National Institute of Diabetes and Digestive and Kidney Diseases. <https://www.niddk.nih.gov/health-information/diabetes/overview/what-is-diabetes>
- [4] What is Diabetes? Centers for Disease Control and Prevention.
<https://www.cdc.gov/diabetes/basics/diabetes.html>
- [5] What Is Type 1 Diabetes? Centers for Disease Control and Prevention.
<https://www.cdc.gov/diabetes/basics/what-is-type-1-diabetes.html>
- [6] Type 1 Diabetes. National Institute of Diabetes and Digestive and Kidney Diseases. <https://www.niddk.nih.gov/health-information/diabetes/overview/what-is-diabetes/type-1-diabetes>
- [7] Diabetes Tests. Centers for Disease Control and Prevention.
<https://www.cdc.gov/diabetes/basics/getting-tested.html>
- [8] Insulin Pen Image. Buy Canadian Insulin.
<https://www.buycanadianinsulin.com/wp-content/uploads/2022/06/61aa5569b48c4-e1656532324384.jpg>
- [9] What Is Type 2 Diabetes? (2022, October 13). Endocrineweb.
<https://www.endocrineweb.com/conditions/type-2-diabetes>
- [10] Type 2 Diabetes. Centers for Disease Control and Prevention.
<https://www.cdc.gov/diabetes/basics/type2.html>
- [11] Understanding Type 2 Diabetes. (2021, October 19). Healthline.
<https://www.healthline.com/health/type-2-diabetes#symptoms>
- [12] Type 2 diabetes - Diagnosis and treatment - Mayo Clinic.
<https://www.mayoclinic.org/diseases-conditions/type-2-diabetes/diagnosis-treatment/drc-20351199>
- [13] Professional, C. C. M. Gestational Diabetes. Cleveland Clinic.
<https://my.clevelandclinic.org/health/diseases/9012-gestational-diabetes>

- [14] What Is Gestational Diabetes? (2022, April 4). Endocrineweb.
<https://www.endocrineweb.com/conditions/gestational-diabetes>
- [15] Gestational diabetes. nhs.uk. <https://www.nhs.uk/conditions/gestational-diabetes/>
- [16] Complications of diabetes. Diabetes UK.
<https://www.diabetes.org.uk/guide-to-diabetes/complications>
- [17] El Naqa, I., & Murphy, M. J. (2015). What is machine learning? (pp. 3-11). Springer International Publishing.
- [18] Iriondo, R., & Shukla, P. (2022). What is Machine Learning (ML)? Towards AI. <https://towardsai.net/p/machine-learning/what-is-machine-learning-ml-b58162f97ec7>
- [19] Nasteski, V. (2017). An overview of the supervised machine learning methods. Horizons. b, 4, 51-62.
- [20] Supervised Learning Image. Neurospace.
<https://neurospace.io/blog/2020/08/what-is-supervised-learning/images/what-is-supervised-learning.png>
- [21] What is Unsupervised Learning? IBM.
<https://www.ibm.com/topics/unsupervised-learning>
- [22] Supervised Unsupervised and Semi-supervised learning Images. Enjoy Algorithms. <https://www.enjoyalgorithms.com/blogs/supervised-unsupervised-and-semisupervised-learning>
- [23] Introduction to Semi-Supervised Learning. Javatpoint.
<https://www.javatpoint.com/semi-supervised-learning>
- [24] Reinforcement Learning Tutorial. Javatpoint.
<https://www.javatpoint.com/reinforcement-learning>
- [25] What Is Reinforcement Learning? Math Works.
<https://www.mathworks.com/discovery/reinforcement-learning.html>
- [26] Kelleher, J. D. (2019). Deep learning. MIT press.
- [27] What is a neural network? IBM. <https://www.ibm.com/topics/neural-networks>
- [28] Neural network image. Serokell.
[https://serokell.io/files/no/no39r3xi.2_\(3\)_1.png](https://serokell.io/files/no/no39r3xi.2_(3)_1.png)

- [29] Sharma, S., Sharma, S., & Athaiya, A. (2017). Activation functions in neural networks. *Towards Data Sci*, 6(12), 310-316.
- [30] Sigmoid Activation Function. *The Science of Machine Learning*. <https://www.ml-science.com/sigmoid-activation-function?rq=sigmoid>
- [31] Sigmoid Activation Function Curve. <https://en.wikipedia.org/wiki/File:Logistic-curve.svg>
- [32] Activation Functions in Neural Networks - Towards Data Science. *Medium*. <https://towardsdatascience.com/activation-functions-neural-networks-1cbd9f8d91d6>
- [33] ReLu Activation Function Graph. *ResearchGate*. https://www.researchgate.net/figure/ReLU-function-graph_fig2_346250677
- [34] Tanh Activation Function. *The Science of Machine Learning*. <https://www.ml-science.com/tanh-activation-function?rq=Hyperbolic>
- [35] Softmax Function. *The Science of Machine Learning*. <https://www.ml-science.com/softmax-function?rq=softmax>
- [36] Ajit, A., Acharya, K., & Samanta, A. (2020, February). A review of convolutional neural networks. In *2020 international conference on emerging trends in information technology and engineering (ic-ETITE)* (pp. 1-5). IEEE.
- [37] CNN Architecture. *Medium*. <https://medium.com/techiepedia/binary-image-classifier-cnn-using-tensorflow-a3f5d6746697>
- [38] A Basic Introduction to Convolutional Neural Network. *Medium*. <https://medium.com/@himadrisankarchatterjee/a-basic-introduction-to-convolutional-neural-network-8e39019b27c4>
- [39] Understand the architecture of CNN - Towards Data Science. *Medium*. <https://towardsdatascience.com/understand-the-architecture-of-cnn-90a25e244c7>
- [40] Convolutional Neural Networks (CNN): Step 3 - Flattening. *SuperDataScience*. <https://www.superdatascience.com/blogs/convolutional-neural-networks-cnn-step-3-flattening>
- [41] What are Recurrent Neural Networks? *IBM*. <https://www.ibm.com/topics/recurrent-neural-networks>

- [42] Recurrent Neural Network. Simplilearn.
https://www.simplilearn.com/tutorials/deep-learning-tutorial/rnn#what_is_a_recurrent_neural_network_rnn
- [43] Van Houdt, G., Mosquera, C., & Nápoles, G. (2020). A review on the long short-term memory model. *Artificial Intelligence Review*, 53, 5929-5955.
- [44] Hinton, G. E. (2009). Deep belief networks. *Scholarpedia*, 4(5), 5947.
- [45] Fischer, A., & Igel, C. (2012). An introduction to restricted Boltzmann machines. In *Progress in Pattern Recognition, Image Analysis, Computer Vision, and Applications: 17th Iberoamerican Congress, CIARP 2012, Buenos Aires, Argentina, September 3-6, 2012. Proceedings 17* (pp. 14-36). Springer Berlin Heidelberg.
- [46] Esfahani, M. T., Ghaderi, M., & Kafiyeh, R. (2018). Classification of diabetic and normal fundus images using new deep learning method. *Leonardo Electron. J. Pract. Technol*, 17(32), 233-248.
- [47] Swapna, G., Vinayakumar, R., & Soman, K. P. (2018). Diabetes detection using deep learning algorithms. *ICT express*, 4(4), 243-246.
- [48] s.m.hasni, M., Gihedan, A., Albargathe, S., & M ABDULWANES AWAD, O. (2022). A PREDICTION OF DIABETES POSSIBILITY WITH DEEP LEARNING: LSTM AND ESA HYBRID MODEL. *ALQIRTAS JURNAL*, 20, 24-38.
- [49] Chowdary, P. B. K., & Kumar, R. U. (2021). An Effective Approach for Detecting Diabetes using Deep Learning Techniques based on Convolutional LSTM Networks. *International Journal of Advanced Computer Science and Applications*, 12(4).
- [50] Raj, M. A. H., Al Mamun, M., & Faruk, M. F. (2020, June). CNN based diabetic retinopathy status prediction using fundus images. In *2020 IEEE Region 10 Symposium (TENSYP)* (pp. 190-193). IEEE.
- [51] Datasets. Kaggle. <https://www.kaggle.com/datasets>
- [52] Diabetes Data Set. Kaggle.
<https://www.kaggle.com/datasets/vikasukani/diabetes-data-set>
- [53] Introduction to Convolutional Neural Networks Architecture. ProjectPro.
<https://www.projectpro.io/article/introduction-to-convolutional-neural-networks-algorithm-architecture/560>

- [54] Introduction to Convolutional Neural Networks (CNN). Analytics Vidhya. <https://www.analyticsvidhya.com/blog/2021/05/convolutional-neural-networks-cnn/>
- [55] Learn About Long Short-Term Memory (LSTM) Algorithms. Analytics Vidhya. <https://www.analyticsvidhya.com/blog/2021/03/introduction-to-long-short-term-memory-lstm/>
- [56] An Overview of Deep Belief Network (DBN) in Deep Learning. Analytics Vidhya. <https://www.analyticsvidhya.com/blog/2022/03/an-overview-of-deep-belief-network-dbn-in-deep-learning/>
- [57] Python. Python.org. <https://www.python.org/>
- [58] Google Colaboratory. <https://colab.research.google.com/>
- [59] TensorFlow. <https://www.tensorflow.org/>
- [60] Keras. <https://keras.io/>
- [61] Matplotlib — Visualization with Python. <https://matplotlib.org/>
- [62] scikit-learn. <https://scikit-learn.org/stable/>
- [63] pandas - Python Data Analysis Library. <https://pandas.pydata.org/>
- [64] NumPy. <https://numpy.org/>
- [65] seaborn: statistical data visualization. <https://seaborn.pydata.org/>
- [66] Accuracy, Recall, Precision, F1-Score. Medium. <https://towardsdatascience.com/accuracy-recall-precision-f-score-specificity-which-to-optimize-on-867d3f11124>