FIRAT UNIVERSITY
GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES
TÜRKİYE

THE PERFORMANCE COMPARISON OF ENSEMBLE MACHINE LEARNING CLASSIFIERS ON MEDICAL DATASETS

Mohammed LAWAN

Master's Thesis

DEPARTMENT OF SOFTWARE ENGINEERING

NOVEMBER 2022

FIRAT UNIVERSITY
GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES
TÜRKİYE

Department of Software Engineering

Master's Thesis

THE PERFORMANCE COMPARISON OF ENSEMBLE MACHINE LEARNING CLASSIFIERS ON MEDICAL DATASETS

Author
Mohammed LAWAN

Supervisor
Prof. Dr. Engin AVCI

NOVEMBER 2022
ELAZIG

FIRAT UNIVERSITY
GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES
TÜRKİYE

Department of Software Engineering

Master's Thesis

Title:	The Performance Comparison of Ensemble Machine Learning Classifiers On Medical Datasets
Author:	Mohammed LAWAN
Submission Date:	23 September 2022
Defense Date:	07 November 2022

THESIS APPROVAL

This thesis, which was prepared according to the thesis writing rules of the Graduate School of Natural and Applied Sciences, Fırat University, was evaluated by the committee members who have signed the following signatures and was unanimously approved after the defense exam made open to the academic audience.

Signature

Supervisor:	Prof. Dr. Engin AVCI Firat University, Faculty of Technology	Approved
Chair:	Assist. Prof. Dr. Bihter DAS Firat University, Faculty of Technology	Approved
Member:	Assoc. Prof. Dr. Eser SERT Malatya Turgut Ozal University, Faculty of Engineering and N. Science	Approved

This thesis was approved by the Administrative Board of the Graduate School on

..... / ..... / 20 .....

Signature

Prof. Dr. Burhan ERGEN
Director of the Graduate School

DECLARATION

I hereby declare that I wrote this Master's Thesis titled “The Performance Comparison of Ensemble Machine Learning Classifiers On Medical Datasets” in consistent with the thesis writing guide of the Graduate School of Natural and Applied Sciences, Firat University. I also declare that all information in it is correct, that I acted according to scientific ethics in producing and presenting the findings, cited all the references I used, express all institutions or organizations or persons who supported the thesis financially. I have never used the data and information I provide here in order to get a degree in any way.

07 November 2022

Mohammed LAWAN

PREFACE

There was a huge gap in the medical field in an adaptation of computing to facilitate fast and accurate diagnostics of results for time-consuming chronic diseases. This research is geared toward the application of the Performance Comparison Of Ensemble Machine Learning Classifiers On Medical Datasets for medical practitioners.

This research will go a long way in addressing the limitation of computing in the medical field, we used a series of medical datasets to set up this simulation to enable us to practice this experiment, among the medical dataset used, are Brain tumors, Lung cancer, and Diabetes. These diseases are critically harmful to human beings and are so dangerous and expensive to maintain. We have to contribute to our quarter by analyzing these datasets to give hope and reduce the workload to patients and doctors and the entire humanity. The result is fascinating and courageous and will be integrated into the medical system.

Prof. Engin AVCI has been an invaluable mentor throughout my studies. I will also like to appreciate my sponsor, the National Information Technology Development Agency (NITDA).

My appreciation goes to my, uncles, aunts, brothers, sisters, scholars, my nephew Fatima Kaka SHEHU, my wife, and my daughter Aisha LAWAN. My twin sister Halima BK who without mentioning everything is incomplete and a friend becomes a sister of Dr. Asmau ZAKARI. Finally, I dedicated this thesis to my late parents and my late brother Baba LAWAN who promise to come to my graduation.

Mohammed LAWAN
ELAZIG, 2022

CONTENTS

ABSTRACT

The Performance Comparison of Ensemble Machine Learning Classifiers On Medical Datasets

Mohammed LAWAN

Master's Thesis

FIRAT UNIVERSITY
Graduate School of Natural and Applied Sciences

Department of Software Engineering

November 2022, Page: xi + 42

The machine learning process may be broken down into two distinct phases: learning, in which the classification algorithmic program is trained, and classification, in which the algorithmic program labels new data. Classification, also known as supervised learning, is a technique for processing data that divides the information into predetermined groups and categories.

We are examining the J48 Decision Tree, the Artificial Neutral Network (ANN), the Extreme Learning Machine (ELM), and other other notable ensemble machine learning classifiers. Single-hidden-layer feedforward neural networks spawned the new field of ELM, which focuses on regression and classification challenges. In this thesis, activation functions will be used to apply the three basic classifiers—ELM, J48, and ANN—to medical datasets. These databases include information on diseases such as diabetes and hepatitis. The ELM classifier is fed both healthy and harmful features. In addition, the logarithmic and tangent sigmoid activation functions, as well as others, are included into ELM in order to optimize classification performance. Based on information theory, the J48 approach does classification using decision trees. It's a version of Ross Quinlan's older ID3 approach, sometimes referred to as J48 in Weka and named after Java. SVM is sometimes referred to as a statistical classifier due to the fact that C4.5 creates decision trees that are utilized for classification. The C4.5 algorithmic rule (J48) is often used to categorize data in a vast array of fields, including the diagnosis of coronary heart disease by evaluating clinical data, classification of data for e-governance, and many others. In the realm of computing, artificial neural networks (ANNs) were developed in an attempt to emulate the neuronal network that makes up a person's brain. ANNs enable computers to absorb information and make judgments in a way that closely resembles that of humans. The artificial neural network is the product of programmers' desire for computers to act similarly to an interconnected network of brain cells. We will employ classifiers such as ELM, Sigmoid, and Relu for a medical dataset of patients with a patient population of 900 to 1000, leveraging a free online dataset of patients with mild to severe illness difficulties.

Keywords: Extreme learning machine; Artificial intelligence; logarithmic sigmoid; computer vision; tangent sigmoid; single-hidden layer feedforward neural networks, ANN, J48, SVM

ÖZET

Uç Öğrenme Makinesi Sınıflandırıcılarının Tıbbi Veri Kümeleri Üzerindeki Performans Karşılaştırması

Mohammed LAWAN

Yüksek Lisans Tezi

FIRAT ÜNİVERSİTESİ
Fen Bilimleri Enstitüsü

Yazılım Mühendisliği Anabilim Dalı

Kasım 2022, Sayfa: xi + 42

Makine öğrenimi süreci iki ana aşamaya ayrılır: sınıflandırma algoritmik programının öğretildiği öğrenme aşaması ve algoritmik programın yeni verileri etiketlediği sınıflandırma aşaması. Denetimli öğrenme olarak da bilinen sınıflandırma olarak bilinen veri işleme tekniği, bilgileri önceden belirlenmiş gruplara ve kategorilere ayırır.

J48 Karar Ağacı, Yapay Nötr Ağ (YSA), Aşırı Öğrenme Makinesi (ELM) ve diğer önemli topluluk makine öğrenimi sınıflandırıcıları, incelemekte olduklarımızdır. ELM, tek gizli katmanlı ileri beslemeli sinir ağlarından gelişen regresyon ve sınıflandırma sorunları için yeni bir alandır. Bu tezde, üç ana sınıflandırıcıyı (ELM, J48 ve ANN) tıbbi veri kümelerine uygulamak için aktivasyon fonksiyonları kullanılacaktır. Diyabet, hepatit ve diğer tıbbi durumlar bu veritabanlarına dahil edilmiştir. ELM sınıflandırıcısı, sağlıklı ve sağlıksız özelliklerle sağlanır. ELM ayrıca logaritmik sigmoid ve tanjant sigmoid dahil olmak üzere belirli etkinleştirme işlevlerinden en yüksek sınıflandırma performansını elde etmek için yapılmıştır. Bilgi teorisine dayalı karar ağaçlarının kullanımı ile J48 algoritması sınıflandırma yapar. Bu, Ross Quinlan'ın Weka'da J48 olarak adlandırılan ve Java'dan sonra adlandırılan eski ID3 algoritmasının geliştirilmiş halidir. C4.5'in sınıflandırma için kullanılan karar ağaçları üretmesi nedeniyle, SVM'ye genellikle istatistiksel sınıflandırıcı denir. C4.5 algoritmik kuralı (J48) genellikle klinik verileri yorumlayarak koroner kalp hastalığını tanımlama, e-yönetişim için bilgileri kategorilere ayırma ve çok daha fazlası gibi çeşitli alanlardaki bilgileri kategorize etmek için kullanılır. Bilgisayar dünyasında, bir kişinin beynini oluşturan nöronlar ağını kopyalamaya çalışmak için yapay bir sinir ağı geliştirildi, böylece bilgisayarlar bilgiyi kavramayı ve insan benzeri bir şekilde seçim yapmayı seçebildi. Programcılar, bilgisayarların birbirine bağlı beyin hücreleri ağı gibi çalışmasını, dolayısıyla hayali sinir ağı olmasını isterler. Hafif ila şiddetli hastalık güçlükleri olan hastalardan oluşan ücretsiz bir çevrimiçi veri kümesinin kullanılmasıyla, 900 ila 1000 arasında değişen bir tıbbi veri kümesi için ELM, Sigmoid, Relu ve diğer sınıflandırıcıları kullanacağız.

Anahtar Kelimeler: Aşırı öğrenme makinesi; Yapay zeka; logaritmik sigmoid; Bilgisayar görüşü; teğet sigmoid; tek gizli katman ileri beslemeli sinir ağları, ANN, J48, SVM

LIST OF FIGURES

Page

LIST OF TABLES

SYMBOLS AND ABBREVIATIONS

Abbreviations

ANN : Artificial Neural Network
AI : Artificial Intelligence
ML : Machine Learning
DL : Deep Learning
ELM : Extreme Learning Machine
CNN : Convolutional Neural Network

1. INTRODUCTION

Extreme learning machines (ELM) are an innovative approach to solving regression and classification problems using a single-hidden-layer feed-forward neural network. Here, we employ activation functions to apply ELM to medical datasets. [1]. These medical databases cover heart illness, leukemia, sickle cell anemia, as well as kidney and bone marrow transplants. The healthy and unhealthy traits are presented as inputs to the ELM classifier [2]. The ELM was designed to achieve the highest possible classification performance [3] for specific activation functions, including the logarithmic sigmoid, sinusoid, and tangent sigmoid. Our bodies are composed of countless billions of trillions of cells. Throughout the course of our lives, we will expand and divide as needed. Cancer happens when regular biological processes fail, causing our cells to produce new body cells while old or aberrant cells do not die as they should. Normal biological processes are difficult for the human body to do. [4]. According to studies and research, cancer is a collection of diseases that impact one-third of the world's population. Nonetheless, hematologic malignancies (blood cancers) and brain tumors [5] are the two most prevalent types of cancer. Despite the fact that they may manifest and respond to treatment differently, certain diseases share many characteristics. Machine learning is being utilized in medicine to automate disease identification [2]. Finding a machine learning method that provides the quickest and most accurate findings and applying it to this medical data [3] could be the largest obstacle on the route to accomplishing this goal. The ELM classifier approach [1] has been proposed as a solution to this problem because it has been proven in academic literature to deliver speedy and accurate results. Data classification is crucial to data mining, making it one of the most critical tasks involved. The input is a large dataset consisting of training records with various attributes. Categorical characteristics have fields other than numbers, whereas numerical attributes have numerical ranges. The class label is an additional feature. This classifier's purpose is to create a panel model for predicting class labels from unlabeled data. Typically, the success of a classification system is measured by the proportion of correctly classified rows. This is within the realm of acceptability. It should almost definitely be determined. A study of multiple activation functions, including logarithmic Sigmoid, Sinusoid, and tangent Sigmoid, will be conducted to improve the ELM classifier's performance in terms of rapid identification accuracy on a larger set of medical data from the UCI database. Ultimately, the activation function with the highest quality-of-life outcomes will be chosen. An ELM Sigmoid has been developed so that all installed CNN networks can reach their full potential. To summarize our contributions, we have provided.

ELM with single-hidden-layer feed-forward neural networks is an innovative technique for regression and classification. We apply ELM with activation functions to medical datasets. These

resources cover heart disease, leukemia, sickle cell anemia, and organ transplants. ELM classifier receives a list of positive and negative qualities [2]. ELM was designed to work with logarithmic sigmoid, sinusoid, and tangent sigmoid activation functions to improve classification accuracy. Humans have billions of cells. We'll keep evolving and forming new groups as long as we're alive. Our cells continue to produce new body cells, but the old or abnormal ones don't die off. It hinders our bodies function. [4]. According to studies, cancer is a set of linked, equally lethal diseases. Hematologic (blood) and brain cancers are the most common. Despite having similar characteristics, different diseases develop and respond to treatment differently. Machine learning in healthcare [2] aims to improve disease diagnosis speed and accuracy. Finding a machine learning technique that offers the fastest and most accurate results [3] is the biggest challenge in achieving this goal. Prior research has shown that the ELM classifier method [1] provides fast and accurate results. Data mining requires classification. Input includes training records with different attributes. "Category" includes non-numerical attributes, while "number" has numerical attributes. Class naming is another distinction. This classifier aims to develop a panel model that can accurately predict class labels for unlabeled data. Classifiers' performance is measured by the percentage of categorized rows. A suitable category fits something. Likewise, to improve the ELM classifier's performance on a larger collection of UCI medical data, we plan to examine its reaction to other activation functions, such as the logarithmic Sigmoid, Sinusoid, and tangential Sigmoid. It chooses the highest-performing activation function. Final phase: ELM Sigmoid merges with CNN networks.

This is what we added to the table:

i. ELM is the unique technique utilized in this study for SLFN. Categorization within the field of automated diagnostic systems is effective.
ii. Previous research on autonomous diagnostic systems has demonstrated that the slow learning rate of feed-forward neural networks is a significant drawback. This situation of feed-forward neural networks is due to two fundamental factors: (1) a huge variety of algorithms are used to train neural networks, and (2) these learning algorithms modify all network parameters frequently [5–9]. To overcome the limitations of feed-forward neural networks in the field of autonomous diagnostics, this study provides an ELM algorithm in SLFN [5–9] that randomly picks remote nodes and computes the SLFN Output Weight. This approach provides strong generalization performance and rapid learning rates in concept.
iii. The Sigmoid method is a notable machine-learning technique for handling categorical and continuous data [19–20].

iv. In the field of artificial intelligence, an ANN seeks to duplicate the network of neurons that comprise normal memory so that computers can absorb concepts and make decisions in a life-like manner. In an artificial neural network, computers are programmed to act similarly to a network of brain cells. [18-20].

v. Machine Learning Techniques (MLT) predict early medical datasets to save lives. Numerous medical datasets are accessible in various data stores utilized by practical applications. Researchers have utilized diseases in medical diagnostics and various data mining methods. This system employs multiple algorithms, including noteworthy prediction algorithms, to create an ensemble hybrid model by combining diverse strategies and procedures into one to improve performance and select the most accurate classifier.

1.1. Problem Statement

The medical field is broad and analytical; by specializing, our project will be limited to certain diseases and patients. Our bodies have always consisted of millions of cell nuclei that develop and multiply as needed throughout our lives. Typically, when aberrant or old tissue dies, illnesses such as cancer emerge. When something goes wrong with this process, our cells continue to produce new cells, while old or defective cells do not die as they should. As they proliferate, cancer cells may displace healthy ones. It complicates the functioning of our bodies. Numerous individuals can effectively treat cancer. Following cancer treatment, more people than ever before are living healthy, productive lives.

1.2. Purpose

The primary purpose is to determine time-efficient and faster algorithms for diagnosing patients at an early stage of treatment, Maintenance, checkups, and emergencies. With the advancement of technology and computing resources availability and usage, it is necessary to utilize a smaller amount's weight for better generalization [21]. Life is so vital that doctors are so sensitive to data and information of their patients, which has changed the medical industries in terms of fast detection of disease, the accuracy of the result, and reliability of the diagnostics.

Machine Learning has become the source of accurate data prediction, which saves time, energy, resources, and lives. It uses data as its input and processes the data as a piece of information as an output format. There is a need to check a comparative analysis between many classifiers and algorithms to check which among them has a quality and accuracy level of acceptance for decision making. In this thesis, we try to evaluate the ELM and another classifier to determine the best diagnostics using three different medical datasets that is Lung cancer, diabetes, and Brain tumor.

1.3. Hypothesis

As stated in the problem statement, the proposed research is to find the fastest and most high-value accuracy of ELM and other classifiers.

1.4. Overview of the research

The following is an overview of the thesis.

- Chapter one. This chapter explores the significance of the research thesis in the field of medical science in terms of giving the most optimal diagnostics of disease in the shortest period possible and accurate prediction of results. Medical practitioners are frequently overburdened with work and, at times, negligent in carrying out their responsibilities, resulting in the loss of lives or suffering of patients. Consequently, the objective of this research is to develop a precise algorithm for detecting and diagnosing patients utilizing AI technology.
- Chapter two: we will cover the dataset's characteristics and traits, as well as the thesis' methodology and methodology. This section contains background information regarding artificial intelligence, neural networks, ELM, and other classifications.
- Chapter three: offers the study's findings, i.e., the thesis's stage of development and implementation. We examine in-depth information investigations, the model performance of various deployed datasets,
- Chapter four: describes the study's analysis.
- Chapter Five: Conclusion of the resulting hypothesis.

2. RELATED STUDIES

Using machine learning algorithms in medicine allows for rapid and automated disease diagnosis. While pursuing this objective, the most significant obstacle is the necessity to create and implement a machine-learning solution that provides the quickest and most accurate outcomes for these medical datasets. The Extreme Learning Machine (ELM) classifier approach, Sigmoid, ReLU, and others have been recommended, since they have been seen to produce quick and accurate results in previous research. The following papers and articles are among the numerous that have contributed to the success of comparing ELM and different classifiers.

Numerous academics have labored to develop and improve the ELM algorithm. It is possible to train data blocks with fixed or variable sizes using an incremental value and a scientific online sequential ELM approach. As a result, it offers a method for training the ELM algorithm using an excessive number of data samples. [18].

Zhang et al. developed the residual compensation ELM for a regression issue by employing a large baseline layer for a specific idea between input and output, followed by the extra layers for excess replacement of distinct layers inside a technique. It addresses multi-class and binary classification issues requiring forecasting models with varying accuracy [36].

Xiao et al. presented a class-specific cost restriction for extreme learning machines and their kernel-based extensions in their 2017 publication. Enhancing modeling abilities and resilience to Gaussian and non-Gaussian noise, among others, they introduced a robust ELM.

To address regression issues, Martnez-Martnez et al. [19] suggested employing a regularized ELM, which provides the best ELM for classification and regression problems as a normalized ELM utilizing the least angle regression (LARS) method. This technique has some disadvantages when working with a collection of interconnected variables.

Various researchers have investigated the distributed ELM algorithm, presented the parallel ELM technique (PELM) based on MapReduce, and developed a parallel OS-similar based on the MapReduce paradigm to train incremental samples on a parallel similar ELM approach involving numerous matrix operations. Numerous research has concentrated on the efficient multiplication of similar matrices. MPI and SUMMA are the most often employed distributed matrix multiplication algorithms among parallel matrix multiplication systems. The primary downsides of this strategy are the data stored in the cluster's shared memory and the construction of sophisticated fault tolerance methods; to solve this issue, other academics offered a way for integrating R. This remains a problem of error tolerance. Hadoop and MapReduce are the foundations of HAMA, which spreads the computation of matrix operations over several nodes. This strategy is inefficient due to the overhead and disk operations incurred by MapReduce processes, making the use of HAMA ineffective.

A. Khellal, H. Ma, and Q. Fei developed an Ensemble of Extreme Learning Machines for Regression. In numerous instances of real-world regression, it compares our results to those of rival methods. First, we compare our results to those of AVG-ELM, an ensemble that calculates the final forecast by averaging all individual responses. In addition, a comparative to our method to well-known algorithms such as Support Vector Regression (SVR), Bidirectional Extreme Learning Machine (B-ELM), and Enhanced Random Search Based Incremental Extreme Learning Machine (EI-ELM) [18, 19]. Generalization performance was examined using the test RMSE (Root Mean Square Error), while speed was compared using training time (in seconds) and test time (in seconds).

L. Gan, L. Extreme Learning Machine had two iterations. Distributor ELM reduces training errors in networks with a fixed number of hidden nodes. The hidden node bias and input weight distributions are encoded with undetermined parameters and optimized with gradient-based optimizers. OD-ELM outperforms other contemporary approaches in the same (slightly better) generation performance condition while requiring less time and fewer hidden nodes. 2) Space Embedded ELM incorporates output space structural data into the hidden layer to improve regression performance. The Human Eva-I dataset tests show that SEELM can capture output space structural data to improve ELM's 3D human pose reconstruction performance. ELM and its derivatives improve classification and regression accuracy. Fresh data will boost performance. Bahtiar, Wibawa, Malik Chronic kidney disease was predicted using traditional ELM and Kernel-Based ELM. Classic ELM, linear ELM, polynomial ELM, RBF-ELM, and wavelet ELM were evaluated in six situations. RBF-ELM predicts chronic renal disease best, both with all features and a subset. Specificity and sensitivity were both 100% [40].

S. Baraha and P. K. Biswal, evaluated ELM with the sigmoid activation function and demonstrated that the testing accuracy of ELM with these activation functions was enhanced. The suggested design uses the estimated functions. To decrease hardware complexity right shifters and comparators are utilized instead of divide-by-zero circuits. The cosine/Fourier basis activation function requires the least amount of hardware resources, whereas the hard limit activation function is the quickest.

S. Ismaeel, A. Miri, and D. Chourishi utilized genuine data from the Cleveland Clinic Foundation; the ELM intensity is used to mimic cardiac disease [37]. With five outputs, this model beats prior BPNN models with four classes or ELM models with a single job (0–4). Using the ELM method, we may assess a new patient's condition using all of our prior knowledge. The ELM solves the problem of learning time, making it more suitable for usage with massive data. It may also be enhanced to address the issue of missing attributes by estimating the impact of their absence on the overall decision.

A. K. Alshamiri, A. Singh, and B. R. Surampudi analyzed the performance of ELM and No-Prop algorithms on a benchmark classification dataset using a benchmark dataset. The ELM and No-Prop techniques for training feed-forward neural networks with randomized hidden layers are discussed separately. In both procedures, the weights of the hidden neurons are initialized at random. While ELM computes output weights analytically, the No-Prop method utilizes the LMS algorithm to repeatedly improve output weights. In this work, we explored and evaluated the performance of the ELM and No-Prop classification algorithms on a variety of benchmark classification data sets. Simulations demonstrate that the No-Prop method is stable and performs well in generalization but has a delayed convergence time [39]. Experiments demonstrate that even though classic ELM has a fast learning rate, it has poor generalization performance when not regularized. The results also indicate that ELM with a regularization parameter outperforms No-Prop on all data sets and performs well in generalization.

The interruption detection principles proposed by P. Gattineni and G. R. S. Dharan have been utilized for decades. In addition, unique ML procedures must be utilized; nonetheless, certain techniques are more appropriate when dissecting enormous data, considering interruption, and uncovering hypothesized systems and statistical outlines. Numerous ML algorithms, including SVM, RF, and ELM, have been investigated and evaluated in an effort to solve the aforementioned issue. ELM surpasses other methods in precision, accuracy, and evaluation of complete knowledge assessments.

3. DEEP LEARNING

Deep learning is a machine learning technique that employs neural networks with three or more layers. These neural networks are designed to mimic human brain activity; they can "learn" from vast quantities of data but with different degrees of effectiveness. Although a single-layer neural network may still provide approximations, more hidden layers may aid in optimizing and adjusting for precision.

Deep learning is utilized by both software and hardware for artificial intelligence (AI). Increase automation by automating both analytical and physical operations. Deep learning technology lies at the heart of common products and services, such as digital assistants and innovations.

Deep learning neural networks, sometimes called artificial neural networks, imitate the human brain using a variety of data inputs, weights, and biases. These components collaborate to recognize, categorize, and characterize data objects with efficiency.

Deep neural networks are composed of several layers of linked nodes, each of which enhances or refines prediction or categorization. The transmission of computations through a network is known as forwarding propagation. There are discernible input and output layers in a deep neural network. After processing the data at the input layer, the deep learning model performs the final prediction or classification at the output layer. The digitization of medical information and photographs has greatly improved the healthcare business due to the application of deep learning algorithms. The program may allow medical imaging specialists and radiographer to examine and evaluate more pictures in less time.

3.1. Artificial Neural Network (ANN)

A network of artificial neurons forms the nodes of an artificial neural network, which is best represented as a weighted directed graph. As shown in Figure 3, the connection between neuron output and input may be seen as directed edges with weights. The input signal from an external source is supplied as a sample to the artificial neural network, which also provides the image as a vector. Then, each n input theoretically receives the symbol x(n) in these inputs. Figure 3.1 below shows the flow of artificial neural network.

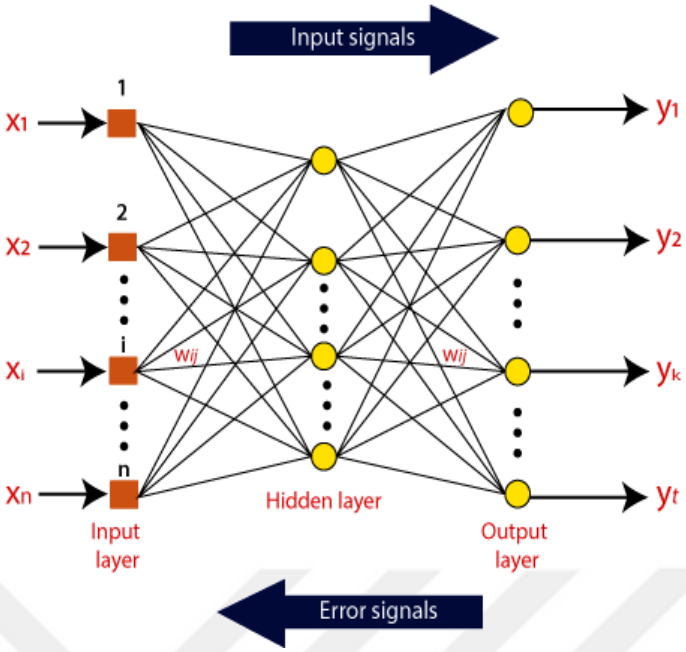

Figure 3.1 The flow of Artificial Neural Network [31]

Afterward, every one of the entries is increased via its related weights (those weights are the facts employed by the artificial neural networks to remedy a specified issue). In a synthetic neural network, the weights often make up the electrical current that connects neurons. The computing unit compiles all of the weighted inputs. If the weighted total is equivalent to 0, bias is applied to cause the output to be non-0 or anything else to scale as much as the system's response. Bias has an equal entry, and the weight equals 1. Here, the full weighted inputs may be within the variety of zero to helpful infinity. Here, a certain maximum rate is benchmarked to keep the response within the preferred fee's bounds, and the activation characteristic Figure 3.1 is used to surpass the total weighted inputs. The switch capabilities employed to get the desired output are referred to by the activation characteristic. There is a unique activation feature, although typically, both linear and non-linear units of power are used. The Binary, Linear, and Tangent sigmoidal activation capabilities are a few of the fundamental units of activation capabilities. Figure 3.2 below depicts the architectural design of ANN.

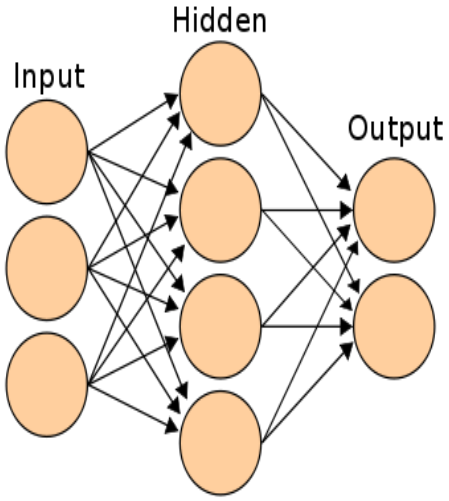

Figure 3.2 ANN Architecture [32]

ANN is applicable in predictive modeling. This is because artificial neural networks (ANN) often attempt to over-fit the connection. When what occurred in the past is being recreated almost exactly, ANN is often used. Suppose we want to use an ANN to target consumers based on their prior responses. Since the link between the response and other predictors may have been over-fitted, it will likely choose the incorrect approach. For whatever reason, it performs very well in speech and picture recognition situations.

3.1.1. Types of Artificial Neural Network (ANN)

A variety of different types of artificial neural networks (ANN) mimic the neuronal and social activities of the human mind, and a synthetic neural community also performs functions. Most artificial neural networks are quite good at their assigned tasks, such as segmentation or classification, and have certain characteristics in common with their more highly complex counterparts.

i. **Feedback ANN:**

This type of network consists of a core neural network with an input layer, an output layer, and at least one layer of neurons. On the basis of the aggregate activity of the participating neurons and the result obtained by comparing the output to the input, the size of the network can be estimated. This network's key benefit is that it specifies how to assess and identify input patterns. Networks using a single hidden layer feed-forward.

ii. **Feed-Forward ANN:**

This network type is a core neural network with an input layer, an output layer, and at least one layer of neurons. While evaluating its output by examining its input, the scale of the network may be recognized based on the group behavior of the implicated neurons, and the

result can be determined. This network describes how to assess and detect input patterns, which is its primary benefit.

3.2. Single Hidden Layer Feed-Forward Networks (Extreme Learning Machine)

A single-layer feed-forward neural network (SLFN) has the following architecture as depicted in Figure 3.3. When selecting the appropriate activation function, the SLFNs may build borders with any forms and approximate any function with an arbitrarily tiny error [14].

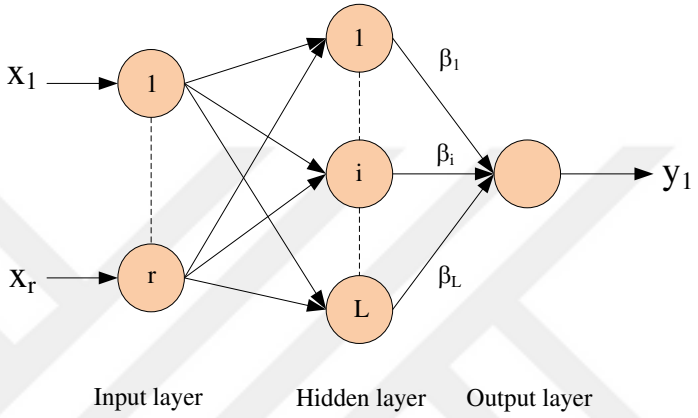

Figure 3.3. Structure of SLFN [33]

The ith output matching to the jth input pattern of an SLFN with n-inputs and C outputs is provided as follows: $o_{ji} = h_j a_i$ (1)

Where $a_i = [a_{i1}, a_{i2}, \ldots, a_{il}]^T$ is the weight vector connecting from the hidden layers to the its output, hj is the hidden layer output vector determined by Eq.(2).

$$h_j = \left[f(w_1 x_j + b_1), f(w_2 x_j + b_2), \ldots, f(w_K x_j + b_K)\right]^T \quad (2)$$

In Eq.(2), wk is the weight vector connecting the inputs to the hidden layer and is determined as Eq.(3). bm is also biased, and f(.) is the activation function.

$$w_k = [w_{K1}, w_{K2}, \ldots, w_{Kn}]^T \quad (3)$$

The main goal of the training process is to determine network parameters that minimize error function defined by:

$$E = \sum_{j=1}^{n} (o_j - t_j)^2 \quad (4)$$

This is equivalent to the solution of a linear system as follows:

$$T = H \times A \quad (5)$$

where H is the output matrix for the hidden layer. One solution to this issue is ELM, where the hidden layer output matrix H is constructed using a random selection of input weights and biases (w, b), and the output weights are then determined by:

$$\hat{A} = H^* \times T \tag{6}$$

Where H^* is the pseudo inverse or Moore-Penrose generalized inverse of H.

Training feed-forward neural networks take longer than necessary [5–9]. This is the most significant limitation of feed-forward neural networks. This conduct is investigated in [5–9]. There are two contributing factors: The neural network is trained using slow gradient-based approaches. This is due to the frequent adjustment of network parameters by the learning algorithms. The extreme learning machine (ELM) learning algorithm is presented to overcome these concerns in [1–5]. In this learning process, the input weights of single-layer feed-forward neural networks (SLFN) are chosen at random, whilst the output weights are derived analytically. The following is a list of essential ELM characteristics:

1. The ELM is really rapid.
2. The ELMs have a more streamlined act.
3. Unlike conventional gradient-based learning algorithms, which suffer from minor problems such as over-fitting, local minima, and incorrect learning rates, the ELM often reaches the answers directly.
4. The ELM technique may train SLFNs with a range of activation functions that are comparable. ELM picks and adjusts the weights between input and hidden neurons at random, in accordance with a continuous probability density function with a uniform distribution function ranging from -1 to 1. Then, the weights between the SLFN's hidden neurons and output neurons are analyzed

This is a quick explanation of the ELM approach for SLFNs:

Given a training set $N = \{(x_i, t_i) | x_i \in R^n, t_i \in R^m, i = 1, 2, ..., N$, activation function f(.) and the number of hidden nodes L:

- ✓ Randomly assign input weight wi and bias bi, (i=1, 2, …, N).
- ✓ Find the hidden layer output matrix H.
- ✓ Find the output weight using $\beta = H^* T$.

Where T is the target matrix, H^* is the Moore–Penrose generalized inverse of the matrix H.

3.3. Activation Function

This activation function illustrates the information included in the inputs, the result of an input, or a set of inputs. To achieve the intended outcome, we choose whether to stimulate or

deactivate neurons. In addition, a non-linear adjustment is made to the input to assess the performance of a complex neural network. In addition, the activation function normalizes any input result between -1 and 1, which is another advantage. Since neural networks are sometimes trained using millions of data points, the activation function must be efficient and should decrease processing time. In any neural network, the activation function decides whether a certain input or received information is meaningful or irrelevant. Look at the illustration below to help us comprehend.

The neuron is an activation function that generates an output from a weighted input.

$$\mathbf{Y = \sum (weights*input + bias)} \quad (1)$$

Y can be any positive or negative infinite integer number of the neuron to get our output.

$$\mathbf{Y = \sum (weights*input + bias} \quad (2)$$

At this time, we applied our activation function to the weight, input, and bias to get our desired result.

3.3.1. Activation Functions Types:

There are numerous varieties of activation function and describes its usage.

i. Binary step function

Every time we attempt to bind output, we are mindful of the underlying nature of this activation function. We use an effective threshold base classifier to determine whether a neuron should be triggered or muted. This function is a binary classifier that accepts a binary value. Consequently, it is frequently favored in output layers. It is not recommended to use it in buried layers since it lacks derivative learning value and will not become visible in the future. Consider then a derivative function; the linear function comes to mind immediately.

ii. Linear Function

Unlike the step function, this function produces a sequence of non-binary activation values. It enables the interconnection of many neurons. Utilizing the backpropagation method, neurons experience the learning process. This method consists of a system of derivatives. In a simple linear activation function, our contribution is proportional to the weighted sum of input or neurons. The range of activations provided by linear activation functions is greater, and their slope is beneficial. The firing rate may increase as the input rate increases. This function is a derivative constant in binary; either a neuron is responding or not.

iii. Sigmoid Function

The sigmoid activation function is often utilized because it does its job well. It is a probabilistic decision-making technique on a scale of 0 to 1. We employ

this activation function when we need to determine or forecast an output since the range is the shortest, resulting in a more accurate prediction. When large inputs between 0 and 1 are converted using the sigmoid function, the vanishing gradient problem occurs. As a result, their derivatives shrank significantly, yielding dismal outcomes. The sigmoid function is a common non-linear artificial neural network (AF) used in feed-forward neural networks. It is a differentiable objective function defined for absolute input values, with positive derivatives and some degree of smoothness. In the output layer of deep learning models, the sigmoid function is employed to predict probability-based outcomes. The sigmoid function has some significant difficulties, such as gradient saturation, convergence, severe damp gradients during backpropagation from deeper hidden layers to input layers, and non-zero-centered output, which permits gradient updates to propagate in several directions.

iv. Hyperbolic Tangent Function

This activation function outperforms the sigmoid function in terms of forecasting or discriminating between two groups. However, it only converts negative input into negative amounts ranging from -1 to 1. When it comes to training multilayer neural networks, the tanh function is far more popular than the sigmoid function. The tanh function is useful for backpropagation since its output is zero-centered. Numerous recurrent neural networks employ Tanh functions for NLP and speech recognition.

v. Rectified Linear Unit(ReLU)

ReLU, despite its design, features a derivative function and allows for backpropagation while staying computationally efficient. The major caveat here is that the ReLU function does not excite all of the neurons at once. If the linear transformation result is less than 0, the neurons will be deactivated. The fast-learning rectified linear unit (ReLU) AF offers state-of-the-art performance with impressive outcomes. When compared to other AFs, including the sigmoid and tanh functions, the ReLU function performs better in deep learning. Since the function is almost linear and keeps the features of linear models, optimizing it with gradient-descent methods is straightforward.

vi. Softmax Activation Function

In the same way that sigmoid activation is utilized, softmax is commonly employed for decision-making at the output layer. The input variables are given weighted values that are added together to equal one using the softmax. While its primary usage is in DL-based text-to-speech applications, it is also widely utilized

in regression computation issues. There are four roots in this polynomial. The main distinction is that it is favored in the deep learning models' output layer, especially when there are more than two classes to be sorted. You may use it to sort data between 0 and 1 to determine the likelihood that they belong to a specific category in order to make a prediction. Therefore, a probabilistic analysis is carried out.

vii. Leaky ReLU Activation Function

Leaky ReLU is an improved ReLU function that overcomes the Dying ReLU issue by having a little positive slope in the negative zone. The advantages of Leaky ReLU are identical to those of ReLU, with the added addition of supporting backpropagation even for negative input values. We required the Leaky ReLU activation function, as mentioned in ReLU, to address the "Dying ReLU" issue. We see that all negative input values rapidly become zero; in the case of Leaky ReLU, we do not set all negative inputs to 0 but rather to a near-zero value, which addresses the ReLU activation function's primary problem.

viii. Swish Function

The Google team developed it. Swish routinely equals or exceeds the ReLU activation function on deep networks used in various difficult applications such as image classification and machine translation. In Deep Neural Networks, the activation functions substantially influence the training dynamics and task performance. The Rectified Linear Unit (ReLU), which is f(x)=max, is now the most popular and commonly used activation function (0,x). Even though several alternatives to ReLU have been presented, none have successfully replaced it due to uneven benefits. As a result, the Google Brain Team introduced Swish, a novel activation function that is just f(x) = x sigmoid (x). Their investigations reveal that Swish outperforms ReLU on deeper models across difficult data sets. Figure. 3.4 depicts each activation function with its corresponding equation and range of positive or negative integers. Figure 3.5 above shows the plots of different activation functions.

ACTIVATION FUNCTION	EQUATION	RANGE
Linear Function	$f(x) = x$	$(-\infty, \infty)$
Step Function	$f(x) = \begin{cases} 0 \; for \; x < 0 \\ 1 \; for \; x \geq 0 \end{cases}$	$\{0, 1\}$
Sigmoid Function	$f(x) = \sigma(x) = \frac{1}{1 + e^{-x}}$	$(0, 1)$
Hyperbolic Tanjant Function	$f(x) = \tanh(x) = \frac{(e^{x} - e^{-x})}{(e^{x} + e^{-x})}$	$(-1, 1)$
ReLU	$f(x) = \begin{cases} 0 \; for \; x < 0 \\ x \; for \; x \geq 0 \end{cases}$	$[0, \infty)$
Leaky ReLU	$f(x) = \begin{cases} 0.01 \; for \; x < 0 \\ x \; for \; x \geq 0 \end{cases}$	$(-\infty, \infty)$
Swish Function	$f(x) = 2x\sigma(\beta x) = \begin{cases} \beta = 0 \; for \; f(x) = x \\ \beta \to \infty \; for \; f(x) = 2\max(0, x) \end{cases}$	$(-\infty, \infty)$

Figure 3.4 Mathematical Expressions of Activation Functions [30 -33]

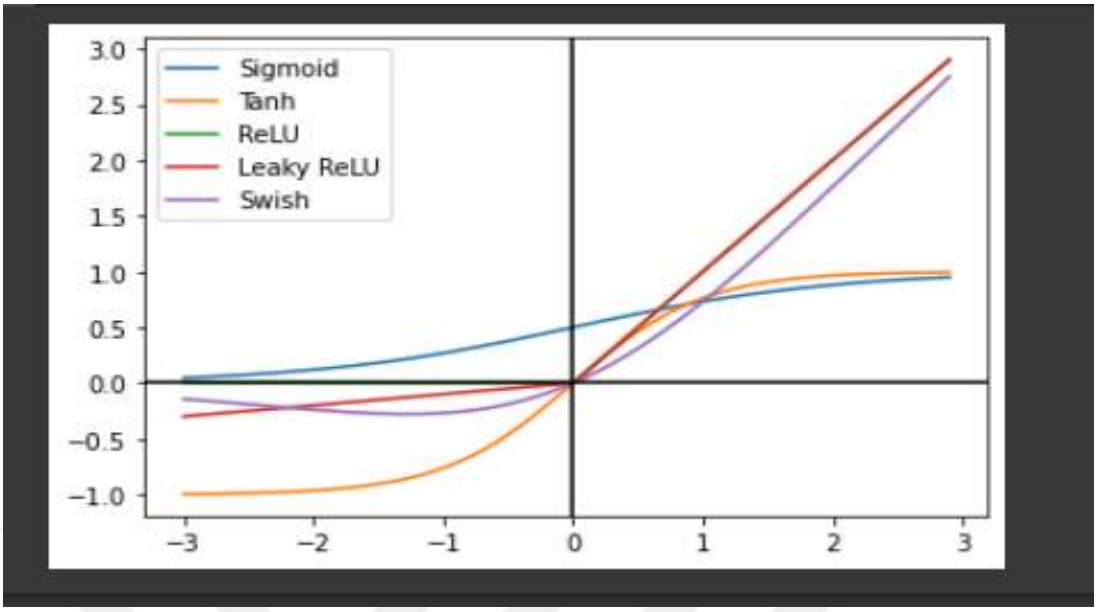

Figure 3.5. Demonstration of Activation Function

3.4. Pattern Diagnosis Concept:

As shown in Fig. 3.5, the pattern diagnostic idea consists of two stages. They have steps for identifying classes and extracting features. Feature extraction is the most important phase of pattern recognition [10, 11]. Features that are crucial to the problem are taken out using a feature extractor. If the right features aren't selected, it doesn't matter what classifier is used; classification performance will suffer. For this reason, it is recommended that the feature extractor reduce the dimension of the pattern vector. If the original feature vector had important information, it should be included in the reduced feature vector as well. Finally, the inputs of the classifier are given a reduced feature vector [12-15]. Figure 3.6 below shows the block diagram of a pattern diagnostic notion.

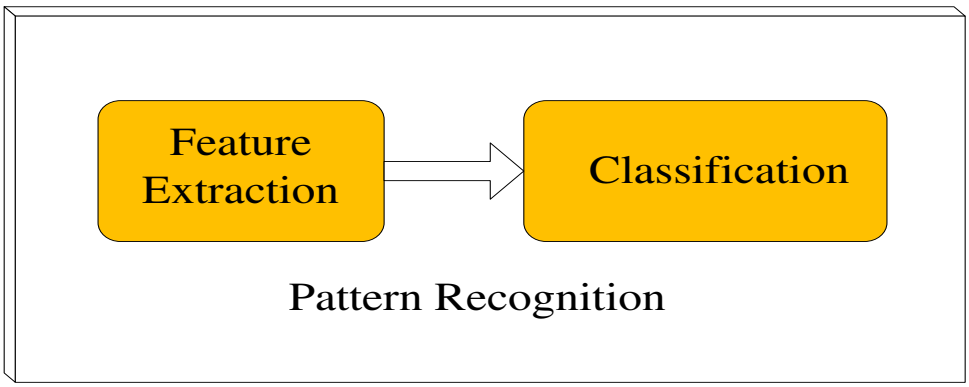

Figure 3.6. The Block Diagram of Pattern Diagnostic Notion.

3.5. Classification Algorithms

As Regression and Classification methods are the two main subsets of supervised machine learning. Regression methods were used to forecast the outcome for continuous values, whilst classification methods were required to do so for categorical outcomes. Classification is a supervised learning technique that uses previously collected data to predict the kind of newly observed data. Using a training dataset, classification software is able to automatically categorize and organize new data. In any case, structured or unstructured data may be better organized with the help of classification. A class is an approach for organizing information into groups. The fundamental objective of a classification challenge is to determine the proper category for a newly acquired data set.

3.5.1. Types of Classification Algorithms

This study explored the most popular classification algorithms as follows:

a. Linear Models
    i. Logistics Regression
    ii. Support Vector Machine
b. Non-linear Models
    i. Naïve Bayes
    ii. Kernel SVM
    iii. K-Nearest Neighbors
    iv. Decision Tree
    v. Random Forest

3.6. Logistics Regression

In the realm of supervised machine learning, logistic regression is a common technique. A set of independent factors is used to make predictions about the categorical dependent variable. Logistic regression is used to forecast the value of a dependent variable that is categorical. Therefore, the result must be either discrete or absolute. Probabilistic values between 0 and 1 are provided in place of precise integers between 0 and 1. You can take that as a yes or a no, a 0 or a 1, or a true or false statement. In this approach, the probabilities associated with the various experimental results are expressed using a logistic function. Logistic and linear regression are quite similar, except for their respective uses. Classification problems may be tackled with logistic regression, whereas regression problems can be dealt with using linear regression. Table 3.1 below gives the advantages and disadvantages of logistic regression.

Table 3.1. Advantages and Disadvantages of Logistic Regression

Advantage	Disadvantage
This use (classification) is intended for and is most useful for determining how numerous independent variables interact to impact a single result variable.	All predictors must be assumed to be uncorrelated with one another, and no missing data must exist, making this method applicable exclusively in binary prediction settings.

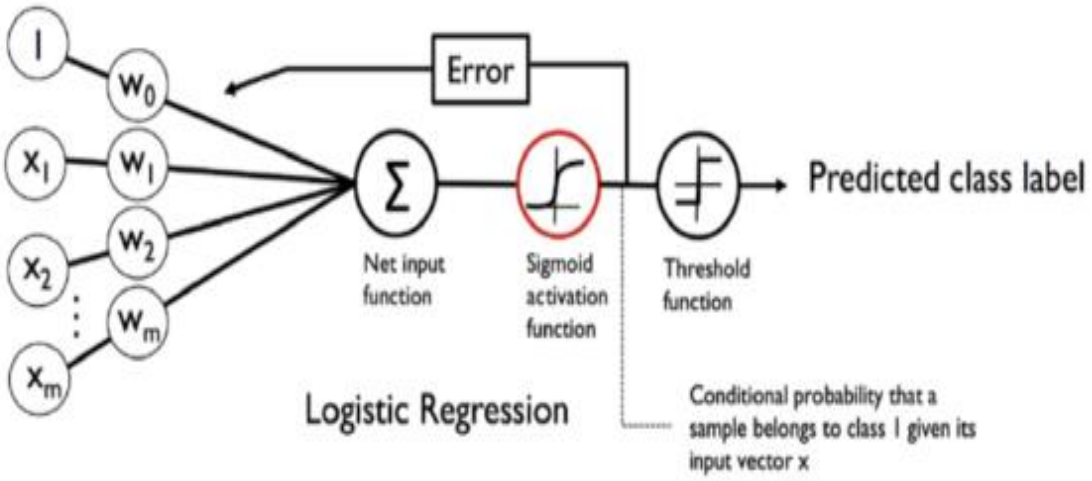

Figure 3.7. A Typical Architecture of Logistic Regression [29]

Figure 3.7. above illustrates a typical Logistic regression architecture.

3.6.1. Type of Logistic Regression

Logistic regression is classified into three categories, namely:

i. Binomial:

This unit only deals with two values, either 0 or 1, yes or no

ii. Multinomial:

The dependent variable might contain three or more unsorted types, such as 'place,' 'animal,' and 'things.'

iii. Ordinal:

Three or more such categories of ordered dependent variables exist, such as "slow," "medium," and "high.".

3.7. Support Vector Machine:

SVM is a common supervised learning method used to solve problems with classification and regression. But in machine learning, it is usually used to sort things into groups. The SVM

technique looks for the best line or decision boundary in n-dimensional space to classify a set of data. A hyperplane shows where the best decisions can be made. With the help of SVM, which picks the most extreme points and vectors to use, the hyperplane is made. The Support Vector Machine is a way of using support vectors in hard situations. The support vector machine shows the training data as a set of points in space with as much empty space as possible between them. Table 3.2 below gives the advantages and disadvantages of SVM.

Table 3.2. Advantages and Disadvantages of Support Vector Machine

Advantage	**Disadvantage**
It performs well in high-dimensional areas and is memory-efficient since it only incorporates a fraction of training points into the decision function.	Probability estimates are not available promptly but instead are calculated through a time-consuming five-fold cross-validation process.

Figure 3.8 below illustrates a typical architecture of support vector machine.

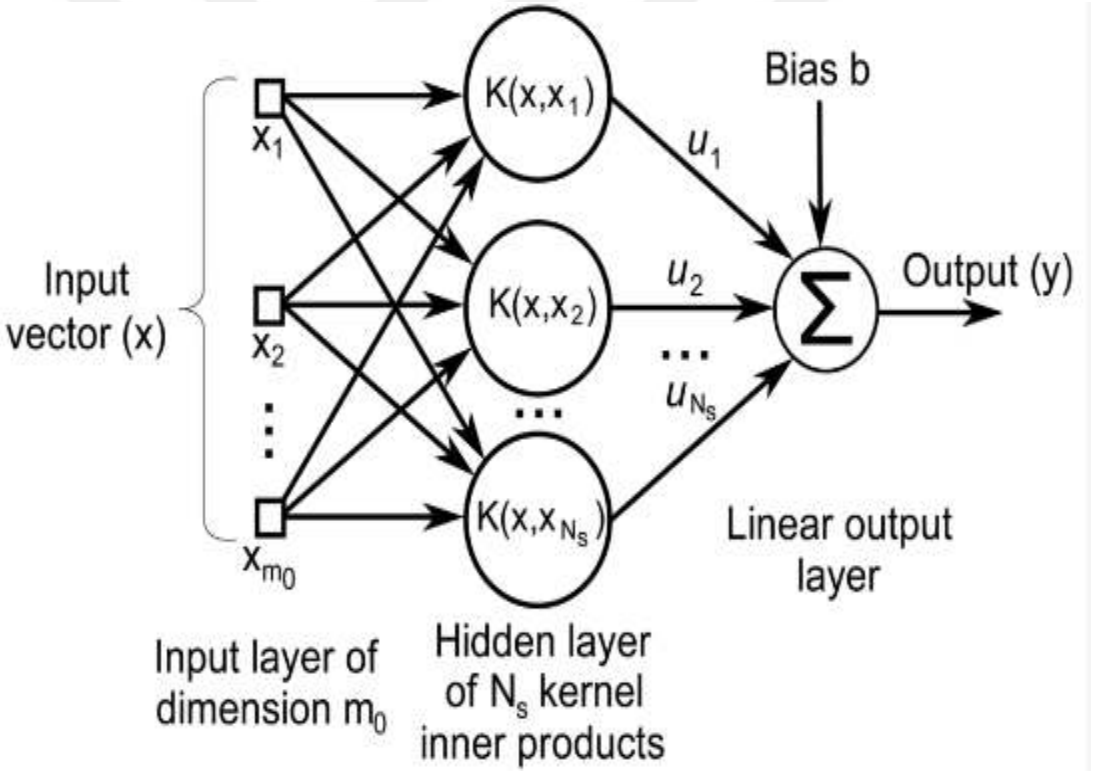

Figure 3.8. A Typical Architecture of Support Vector Machine

3.7.1. Types of SVM

i. Linear SVM:

A dataset is said to be linearly separable if it can be partitioned into two classes by a straight line, and this is the case when employing linear support

vector machines (SVMs). In this case, we employ a linear support vector machine (SVM) classifier.

ii. Non-Linear SVM:

This classifier is utilized for data that is not linearly segregated. If a dataset cannot be neatly placed into one of many predetermined categories using a straight line, it is considered non-linear, and the corresponding classifier is known as a Non-linear Support Vector Machine.

3.8. Naïve Bayes Algorithm

Supervised learning techniques are based on the Bayes theorem and are used to solve classification issues. Its primary application is in the labeling of textual data.

The Naive Bayes Classifier is a fast and simple method for classifying data that may be used to rapidly develop accurate machine learning models. It is a probabilistic classifier, which means that it uses probabilities to produce its predictions.

The Naive Bayes method is frequently used for spam filtering, sentiment analysis, and article classification. Naive Bayes is an algorithm based on the Bayes theorem and the independence assumption between each pair of features. Among its numerous practical uses, Naive Bayes classifiers excel in document classification and spam filtering. The advantages and disadvantages of Naieve Bayes algorithm is depicted in Table 3.3 below

Table 3.3. Advantages and Disadvantages of Naïve Bayes Algorithm

Advantage	**Disadvantage**
Naive Bayes is a quick and straightforward machine learning approach for making predictions over a group of datasets. It may be used for either a binary or a multiclass classification system. It outperforms the other algorithms when making predictions for several classes. It is the approach to text classification challenges that are most often employed.	Naive Bayes cannot learn about feature connections since it thinks that all features are independent or disconnected.

3.8.1 Naïve Bayes Model Types

Nave Bayes models come in three different varieties, including

i. Gaussian:

The Gaussian model predicts that characteristics are distributed normally. If predictors take on continuous rather than discrete values, the model infers that the values come from a Gaussian distribution.

ii. Multinomial:

The Multinomial Nave Bayes classifier is employed when the data is sparse. It's commonly used to describe the type of document (such as sports, politics, or education) that it belongs to when classifying papers.

iii. Bernoulli:

This classifier performs similarly to the multinomial classifier, with the key difference being that the predictor variables in Bernoulli are uncorrelated Boolean variables. The question of whether or not a certain term appears anywhere in a given document is a common approach taken when tasked with document categorization.

3.9. K-Nearest Neighbor

Assuming that the new case or data is similar to the existing cases, the basic machine learning technique is to K-Nearest Neighbor and assign the new issue to the category that is most similar to the general categories. All available data is preserved, and new data is classified according to similarities. It implies that by utilizing the K-NN method, freshly generated data may be quickly classified into an appropriate category. Although it may be applied to both classification and regression issues, it is often employed for problems involving classification.

Being non-parametric means that no presumptions are placed on the data being analyzed. Neighbor-based categorization is similar to lazy learning in that it just preserves selected instances of the training data instead of attempting to build an exhaustive internal model. Each location's closest neighbors vote on its classification using a simple majority system. Table 3.4 below gives the advantages and disadvantages of a K-NN algorithm.

Table 3.4. Advantages and Disadvantages of K-NN

Advantage	Disadvantage
It is straightforward to implement.	The ongoing determination of K's value might be challenging at times.
Noisy training data does not affect it.	
It could be more efficient if the training data is extensive.	Since the range among each training sample's data points is established, the cost is high..

3.10. Decision Tree

As a supervised learning technique, the decision tree may be applied to both classification and regression issues, albeit its primary use case is in the former. It's organized like a tree, with the

trunk representing the dataset itself, the branches representing the many steps in reaching a decision, and the leaves representing the final results.

The two types of nodes that make up a decision tree are the decision node and the leaf node. The decision-making process is represented by choice nodes, which contain many more branches than the leaf nodes, which indicate the outcomes of the decisions made. All tests must conform to the characteristics of the provided dataset. A decision tree is so named because of its tree-like structure, which develops from a central node.

CART is an algorithm that generates trees for use in classification and regression. A decision tree first asks a question and then uses the answer (yes or no) to determine where to place the next node. A decision tree generates a set of rules from a set of characteristics that may be used to classify data according to classifications. Table 3.5 below give the advantages and disadvantages of decision

Table 3.5. Advantages and Disadvantages of Decision Tree

Advantage	**Disadvantage**
Given that it employs the same methodology that a person would use to make decisions in real life, it is simple to understand. It may be quite useful in overcoming decision-making difficulties. It is useful to think of every scenario that may happen. Data purification is not as necessary with this method as it is with other algorithms. Decision trees can handle both numerical and categorical data, are simple to understand and depict, and require no data preparation.	The decision tree is hard since it has several layers. The Random Forest approach may be used to fix any overfitting issues it might have. The decision tree's computational complexity increases with the number of class labels.

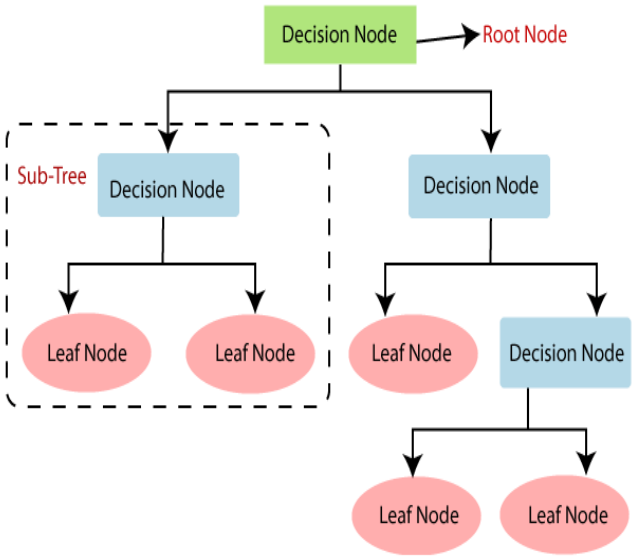

Figure 3.9. A Typical Structure of Decision Tree [34]

Figure 3.9 above depicts an architecture of a decision tree.

3.11. Random Forest

A random forest classifier is a meta-estimator that averages the results of many decision trees trained on different subsamples of information to get the best fit for a particular dataset. Averages will rise. Whenever predicting accuracy, the original input sample size is always utilized as a sub-sample size to prevent over-fitting. However, the samples are created through replacement and broken up into smaller groups. The supervised learning strategy uses the well-known machine learning algorithm, Random Forest. Both classification and regression problems in machine learning may benefit from its use. This method was inspired by ensemble learning, which entails using several classifiers to address a complex problem and boost the model's accuracy. Table 3.6 below shows advantages and disadvantages of Random Forest.

Table 3.6. Advantages and Disadvantages of Random Forest

Advantage	**Disadvantage**
Both classification and regression issues may be handled using Random Forest.	Slow real-time prediction, complicated method, and challenging to implement.
It can process massive, high-dimensional datasets with ease.	It is possible to employ a random forest to solve both the classification and regression issues. However, it does not excel at regression.
It increases model precision and prevents the overfitting issue.	
Decision trees often perform worse than random forest classifiers when over-fitting is reduced.	

3.12. J48

The J48 Iterative Dichotomiser 3 is the machine learning method's backbone. It organizes and displays data. Classification techniques perform differently across datasets. The number of instances, attributes, and types of attributes affect the classifier output quality. Other comparison tuples favor the updatable J48. Classification uses the J48 decision tree. J48 selects the tree node data property that divides tests into more effective subgroups. The division criterion is normalized information gain (contrast entropy). Standardized data shows decision-making is rising. J48 reproduces this point on sublists. We used generalized sequential pattern mining to forecast medical data and improve its performance. Algorithm J48 Quinlan's method for C4.5 updates J48 to split C4.5. Choosing an attribute divides the information into manageable subsets.

Finally, extreme normalized attribute data is used. The algorithm returns subsets. Separation tactics will end if a subgroup always sits with a comparable class. J48 creates a decision node using class estimates. The J48 decision tree can manage particular characteristics, calculate the costs of

variable attributes, and estimate lost or missing data attributes. Trimming [1] may increase precision here.

The Procedure Stage:

1. The leaf is labeled with a similar class if the instances are members of the same class.
2. Both the data obtained from the attribute test and the prospective data for each attribute will be computed.
3. Based on the current selection parameter, the best characteristic will then be chosen.

3.12.1. Limitation of J48 Algorithm

J48 is one of the most well-known algorithms. However, it has drawbacks. The following paragraphs outline some of J48's drawbacks.

i. Branchless trees: Making a tree of relevant values is one of the crucial processes in the J48 algorithm's rule generation process. Numerous nodes with values of zero or extremely close to zero have been discovered throughout our investigation. These values are not, however, any class that is created or added to the categorization process. As a result, the tree becomes more meaningful and perplexing.

ii. Careless branches: The decision tree may have the same number of splits no matter how many separate characteristics are chosen. But in actuality, not all of them are relevant to the categorization assignment. These less important branches not only make the decision tree less usable but also contribute to the overfitting issue.

iii. Overfitting: Happens when the algorithm screen gathers data with distinctive properties. Overfitting happens. The process of delivery is greatly fragmented as a result. Fragmented nodes are those with the fewest cases that are not statistically significant. This strategy works best with noiseless data and provides a perfect categorization of training samples. But most of the time, this approach results in noisy training instances.

These are:

• Let the tree over-fit the training data rather than a post-prune tree;

• If the tree expands, halt it before it reaches the maximum point of accurately categorizing the training data.

4. MATERIAL AND METHOD

In this chapter, the materials and methods deployed will be fully elaborated.

4.1. Dataset

Concerning this research, we considered a medical dataset that is a readily available and critical challenge that effect all over the world; these datasets are

x. Diabetics

xi. Lung cancer

xii. Brain tumor

4.1.1. Diabetes:

These datasets have different properties. The National Institute of Diabetes and Digestive and Kidney Diseases released the raw diabetes data. The goal is to forecast a patient's diabetes using diagnostic data from https://www.kaggle.com/datasets/mathchi/diabetes-data-set [35] These samples were selected from a database based on criteria. Every patient is over 21.

The investigated binary variable determines if a patient has diabetes according to WHO recommendations (i.e., whether the patient's 2-hour post-load plasma glucose was at least 200 mg/dl during any survey examination or normal medical care). Table 4.1 below shows the characteristics of the diabetics dataset.

Table 4.1. Displayed the attributes of Diabetics dataset

S/N	Attributes	Value	Measurement
1.	Pregnancies	Numeric values	Integer
2.	Glucose	Numeric values	Integer (2-hour oral glucose tolerance test's plasma glucose concentration
3.	Blood Pressure	Numeric values	Integer Diastolic blood pressure is a measure of blood pressure (mm Hg)
4.	Skin Thickness	Numeric values	Integer Thickness of the triceps skin fold (mm)
5.	Insulin	Numeric values	Integer Serum insulin after two hours (mu U/ml)
6.	BMI	Numeric values	Integer (weight in kg/(height in m)2)
7.	Diabetes Pedigree Function	Numeric values	Integer Diabetes pedigree function
8.	Age	Numeric values	Integer value of age
9.	Outcome	Numeric values	class value 1 is interpreted as "tested positive for diabetes"

	Class Variable	Numeric values	

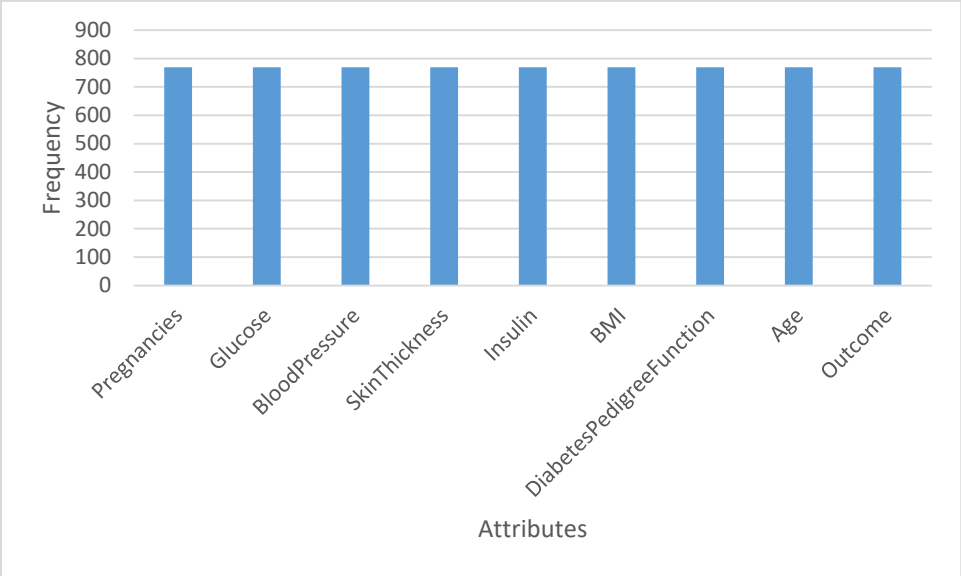

Figure 4.1. Analysis of the Diabetics Data.

The analysis of the data is shown in Figure 4.1 above. Diabetes dataset cases collected and used in this study, which every attributes have its records.

4.1.2. Lung Cancer:

Uncontrolled cell growth is a cancer symptom. Lung cancer is the deadliest malignancy. After skin cancer, lung cancer is the second most common cancer in American men and women. Lung cancer rates are falling after decades of growth as fewer people smoke and treatments improve. Tobacco use often causes lung cancer. Lung cancer can be caused by various tobacco products (such as pipes or cigars), secondhand smoke, radon or asbestos exposure, and family history. This dataset is in UCI's open-source dataset repository

The Dataset:

- Age: Patient age in years
- Gender: Sex (male 1, female 2)
- Air Pollution: Air Greenhouse gases (range 1 clean, 8 dirty)
- Alcohol use: Level of Alcohol consumption (range 1 low, 8 high)
- Dust Allergy: Allergy to environmental air(range 1 low, 8 high)

- Occupational Hazards: Workplace hazards (range 1 low, 8 high)
- Genetic Risk: Family heredity (range 1 low, 8 high)
- chronic Lung Disease: Severances of the cancer (range 1 low, 8 high)
- Balanced Diet: Food supplement of body(range 1 low, 8 high)
- Obesity: body overweighed (range 1 low, 8 high)
- Smoking: Cigarette consumption (range 1 low, 8 high)
- Passive Smoker: Inert of smoke (range 1 low, 8 high)
- Chest Pain: Level of chest pain(range 1 low, 8 high)
- Coughing of Blood: Severances of coughing of blood(range 1 low, 8 high)
- Fatigue: Weakness of body (range 1 low, 8 high)
- Weight Loss: level of body weight loss(range 1 low, 8 high)
- Shortness of Breath: Difficulty of breathing(range 1 low, 8 high)
- Wheezing: gasping of breath(range 1 low, 8 high)
- Swallowing Difficulty: Chewing difficulty (range 1 low, 8 high)
- Clubbing of Finger Nails: Body finger changing(range 1 low, 8 high)
- Frequent Cold: Level of catarrh(range 1 low, 8 high)
- Dry Cough: Level of dry cough(range 1 low, 8 high)
- Snoring: whizzing level(range 1 low, 8 high)
- Level: Disease level(range 1 low, 2 medium and 3 high)
- Result: outcome of disease(range 1 positive)

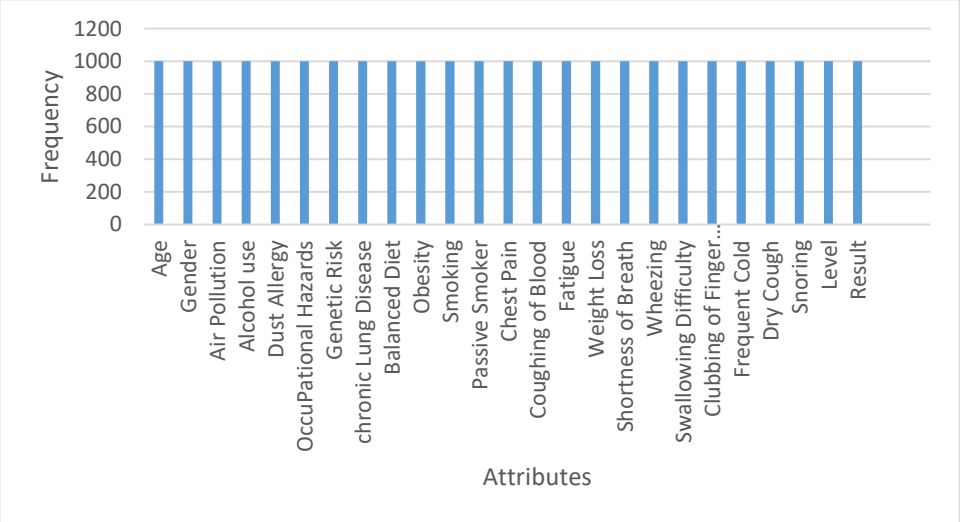

Figure 4.2. Analysis of the Lung Cancer Data.

Figure 4.2 above shows the analysis of the lung cancer cases used in this investigation, with each attribute recorded.

4.1.3. Brain Tumor:

A tumor in the brain is the development of abnormal cells. Not all tumors in the brain are malignant, although some are (non-malignant, non-cancerous, or benign). A brain tumor can originate anywhere in the CNS, including the brain, spinal cord, or cranial nerves. The brain controls our senses, our movements, our emotions, our ideas, our words, and our memories. The brain's ability to function correctly and efficiently can be compromised by a tumor. This data collection was obtained from a publicly available database at UCI's site.

Description of Dataset:

- CGGA_IDCGGA_1002: Identification number
- Gender: Sex of patients
- Age: years of Patient
- OS: Overall survival
- Censor: loss to follow-up or non-occurrence of outcome
- Radio_status: Radioteraphy status
- Chemo_status: Chemotheraphy Status
- IDH_mutation_status: Isocitrate Dehydrogenase (IDH) mutation status
- codeletion_status: prolonged survival of patient

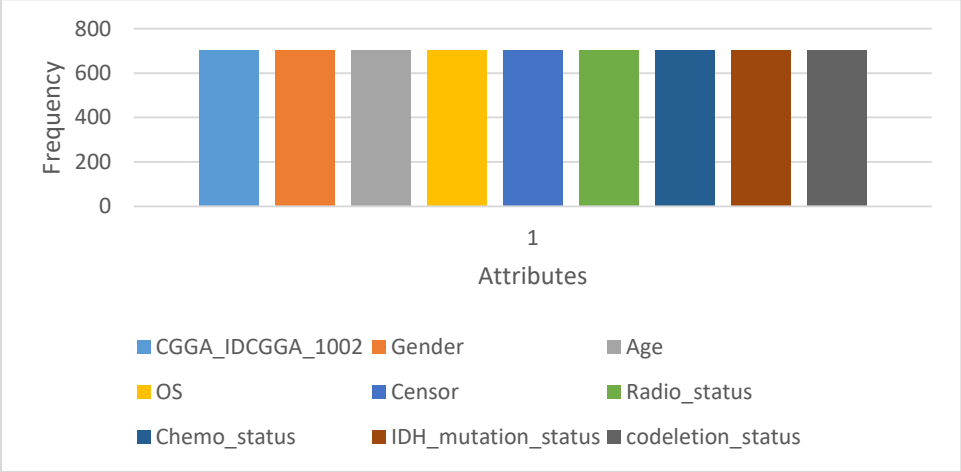

Figure 4.3. Analysis of the Brain Tumor Dataset

The analysis of the Brain Tumor cancer dataset cases that were gathered and utilized in this study is shown in Figure 4.3. above. Each characteristic has a record.

4.2. Deep feature extraction and selection

The Extraction of features and attributes is further explained below.

4.2.1. Feature extraction using pre-trained ELM

Deep convolution neural networks A dataset is kernel-filtered at each convolution offset. A filter consists of a layer of connecting weights that accepts input from the rectangular section of the layer before and a layer of convolution that functions as a feature extractor by converging the local receptive fields of the layer before with fixed-size kernel filters. Once the characteristics are collected, it's less important where they go since their location is normally retained. Each output map size is set. This layer's main goals are feature extraction and shift-invariance by scanning the whole input layer. Using the same kernel across the input layer allows weight sharing.

Batch normalization layers equalize activations and gradients throughout a network, making network training and optimization problem. After batch normalization, the Rectified Linear Unit (ReLU) applies a threshold to each input element. Any input value less than zero is set to 0. After that, each convolution layer comes to a max-pooling layer to reduce map resolution.

Since sigmoid produces results on the range's edges, it creates unambiguous splits. Its outcome may alternatively be regarded as a likelihood. Define sigmoid function:

$$\alpha(u) = 1/(1 + e\hat{}(-u)\ ). \tag{4.1}$$

Because only little modifications are performed in each training update, a NN utilizing it can quickly become stranded in low gradient areas, causing the learning process to plateau.

ReLU is often used in NNs because it makes it possible to build deep neural networks quickly [14]. The Rectified Linear Unit (ReLU) returns 0 if the state is negative; otherwise, it returns the value of the state:

$$\alpha(u) = \begin{cases} 0 & u<0 \\ u & u\geqslant 0 \end{cases}. \tag{4.2}$$

All negative values instantly become zero using ReLU, decreasing the model's ability to learn from input since all neurons with negative states immediately "turn off." This defines NN training and the backpropagation procedure. Leaky ReLU [15], Parametric ReLU [16], and Exponential Linear Unit [17] were created to avoid the problem (ReLU). Figure 4.4 below depicts the flowchart diagram of the proposed methodology.

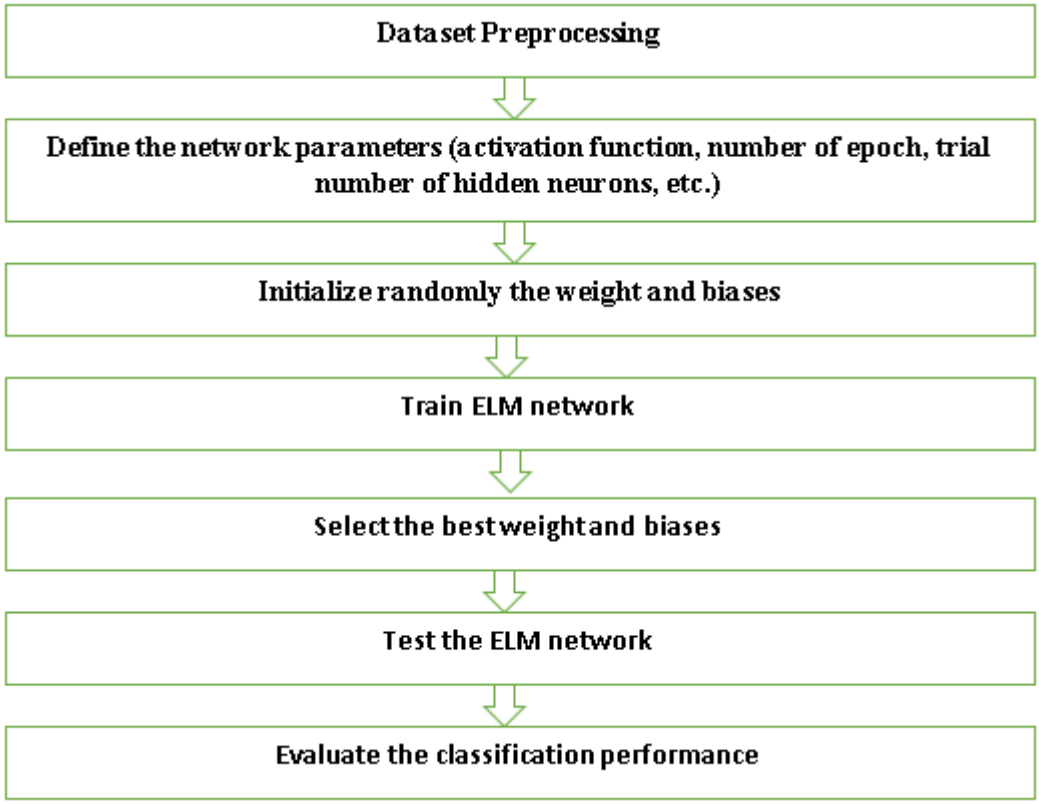

Figure 4.4. Flowchart of the proposed methodology

Table 4.2 below shows the categories of the datasets used for this study.

Table 4.2. Dataset used in this study

Dataset	Category	No. of cases
Diabetes	Medical	769
Lung cancer	Medical	1001
Brain Tumor	Medical	703
Total		2473

As soon as the features have been extracted using DLN, they are sent to the ELM classifier. In ELM, we only use a single layer of hidden neurons, and we use random selection for their weights and biases. The following diagram illustrates how the generalized least squares method is used to calculate weights for analytical output.

$$\gamma=H^{-1}T \tag{4.3}$$

The output layer's weight is denoted by, H1 is the pseudo inverse of the hidden layer's output, and T is the target matrix. The proposed network's weights and biases are initially generated at random, reducing the need for frequent tuning to achieve optimal results.

5. RESULT OF EXPERIMENTS AND ANALYSIS

In this section, we present the findings from the experiments. The findings are as a result of using the proposed model ELM. We present detailed information about the findings from the experiment.

5.1. Implementation details

This section presents setup of the presented model. This work uses Anaconda Navigator Application using Python 3.1 developers programming environment to code the proposed ensemble deep transfer-learning model. Furthermore, ELM (ReLU and Sigmoid), and others classifiers were included to the used Python environment using publically available datasets. The used configuration is presented in the table below: Table 5.1 below shows the properties of the personal computer used for the research.

Table 5.1. Properties of the used PC

Hardware/Software	Attribute
CPU	Intel @i7 9th generation
RAM	32 Gigabytes
Hard disk	1.2TeraBytes
Operating system	Microsoft Windows 11
Programming environment	Anaconda Navigator Application using Python 3.1

5.2. Accuracy evaluation

The precision might be above average. The ELM algorithm offers a high level of statistical power. The accuracy can be affected by:

- The activation function used;
- Whether or not the data is normalized;
- The nature of the characteristics; and
- The overall database distribution.

ReLU (Rectified Linear Unit) activation function is the positive component of its argument in artificial neural networks.

In the late 1960s, hierarchical neural networks used this activation function to extract visual information.

Later, biological and mathematical incentives were considered. In 2011, it was shown to provide better training in deeper networks than commonly used activation functions. Accuracy is the percentage of corrected cases. Equation (5.1) shows True Positive, True Negative, False

Positive, and False Negative. TP means the model properly predicted the positive class. TN is a model-predicted negative class. FP means the model erroneously predicted the positive class. In an FN, the model mistakenly predicts the negative class.

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN} \tag{5.1}$$

Table 5.2. Accuracy and Compilation time of Diabetes dataset using ELM (ReLU) function and other classifiers

Method	Dataset	Accuracy %	Compile Time (Seconds)
ELM ReLU	Diabetes	1	0.02s
Support Vector Machine	Diabetes	0.74	0.033s
Kneighbor Classifier	Diabetes	0.714	0.021s
Logistic Regression	Diabetes	0.746	0.22s
Bagging Classifier	Diabetes	0.753	0.28s
Naïve Bayes	Diabetes	0.77	0.27s
SGD Classifier	Diabetes	0.61	0.23s

Table 5.2. above shows the Accuracy and Compilation time of Diabetes dataset using ELM (ReLU) function and other classifier. That the ELM (ReLU) work has been validated with six other standard classifiers such as Support Vector Machine, K-neighbor Classifier, Logistic Regression,

The SGD Classifier, the Naive Bayes, and the Bagging Classifier. Approximately 70% of the data is used for training purposes, while the remaining 30% is used for testing. According to scientific analysis and reporting of performance parameters like accuracy and time, ELM is completely effective and efficient.

Table 5.3. Accuracy and Compilation time of Brain Tumor dataset using ELM (ReLU) function and other classifiers

Method	Dataset	Accuracy	Compile Time
ELM Relu	Brain cancer	0.915	0.455s
Support Vector Machine	Brain cancer	0.755	1.01s
Kneighbor Classifier	Brain cancer	0.784	0.49s
Logistic Regression	Brain cancer	0.782	0.494s
BaggingClassifier	Brain cancer	0.9	0.486s
Naïve Bayes	Brain cancer	0.773	0.499s
SGD Classifier	Brain cancer	0.384	0.494s

Table 5.3. above shows the accuracy and compilation time of the Brain Tumor dataset using ELM (ReLU) function and other classifiers, which Relu function has almost 92% accuracy and compilation time of 0.455 seconds which is still the best among its other standard classifiers.

Table 5.4. Accuracy and Compilation time of Lung Cancer dataset using ELM (ReLU) function and other classifiers

Method	Dataset	Accuracy%	Compile Time
ELM Relu	Lungs dataset	0.91	0.464s
Support Vector Machine	Lungs dataset	0.634	0.623s
K-Neighbor Classifier	Lungs dataset	0.642	0.623s
Logistic Regression	Lungs dataset	0.641	0.664s
Bagging Classifier	Lungs dataset	0.677	0.462s
Naïve Bayes	Lungs dataset	0.574	0.435s
SGD Classifier	Lungs dataset	0.634	0.799s

Table 5.4 above displays the performance of the ELM (ReLU) function and other standard classifiers on the Brain Tumor dataset in terms of accuracy and compilation time. The Relu function stands out as the best of these classifiers, with 91% accuracy and a compilation time of 0.464 seconds.

5.3. Comparative Analysis

We compare other ELM like sigmoid and observe the difference if there is any among the ELM functions with comparative difference.

Table 5.5. Accuracy and Compilation time of Diabetes dataset using ELM (Sigmoid) function and other classifiers.

Method	Dataset	Accuracy	Compile Time
ELM Sigmoid	Diabetes	0.959	0.02s
Support Vector Machine	Diabetes	0.74	0.033s
K-Neighbor Classifier	Diabetes	0.714	0.021s
Logistic Regression	Diabetes	0.746	0.22s
Bagging Classifier	Diabetes	0.753	0.28s
Naïve Bayes	Diabetes	0.77	0.27s
SGD Classifier	Diabetes	0.61	0.23s

Table 5.5. above shows the Accuracy and Compilation time of Diabetes dataset using ELM (Sigmoid) function and other classifiers. However, Sigmoid function show great value for both accuracy and time, which there is no any comparable to it.

5.3.1. Comparison based on time consumption

Time consumption analysis is displayed in table 5.3 compares the ELM Sigmoid function's time consumption to that of other common classifiers. The time consumption of both ReLU and Sigmoid is very short because of the high performance of ELM. In this study, the

time required to process such a health illness disease is paramount to the diagnoses and life-saving in the medical field, while using ELM's weights and biases and hidden neurons at 5. Figure 5.1. shows detail display of the time constraint. Figure 5.1 below shows the time comparison of ELM and other classifiers.

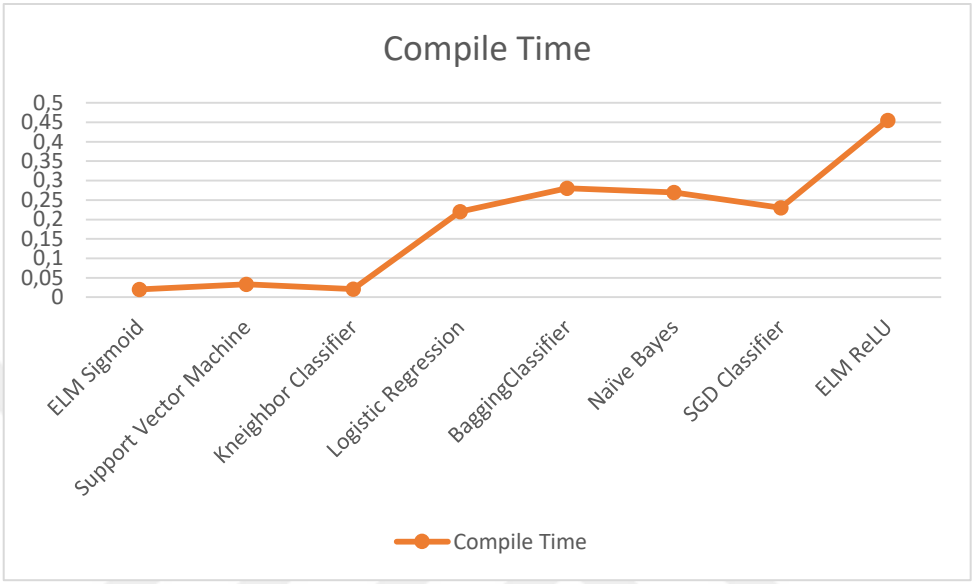

Figure 5.1. Time Comparison of ELM and Other Classifiers

5.3.2. Comparison based on accuracy

The accuracy of the dataset analysis is displayed in table 5.3. displays a comparison of accuracy with the other standard classifiers and the ELM Sigmoid function shows that ELM achieved 100% accuracy, which is an excellent performance. The accuracy of both ReLU and Sigmoid is excellent because of the high performance of ELM. Furthermore, ELM gives you the option of fine tuning the neurons to give to exceptional results. Figure 5.2. below gives display output of the result.

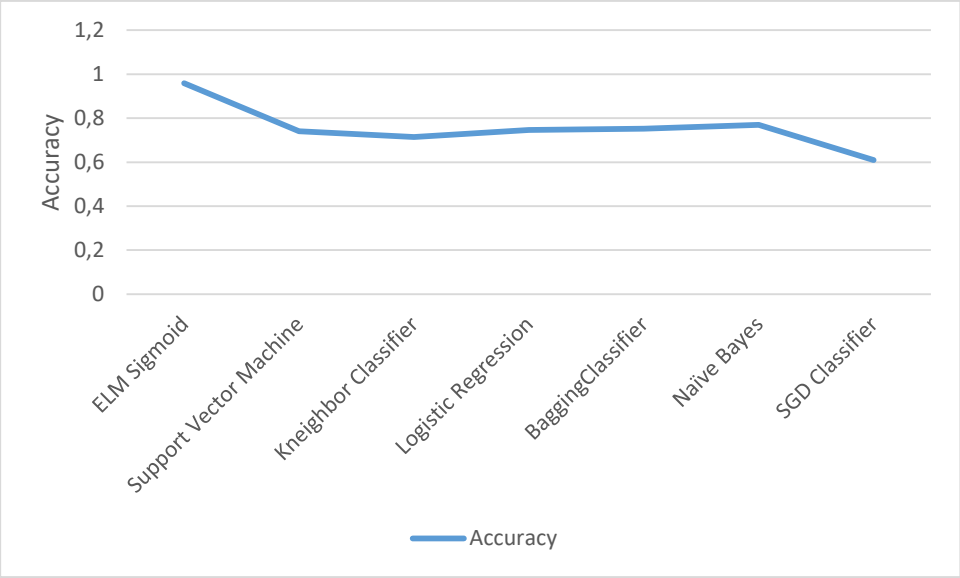

Figure 5.2. Accuracy of ELM (Sigmoid) with other classifiers

5.3.3. Comparison based on ELM functions

Among the two ELM functions that we used each one of them its time fast and achieve high accuracy, however, the ReLU function has better and higher accuracy than the Sigmoid function when it comes to Diabetes, Brain Tumor, and the Lung cancer datasets. Furthermore, Sigmoid has greater advantage when it comes to time completion, because its faster in time compilation. Any how you compare to other classifiers they are extremely fast and accurate. Figure 5.3. shows the accuracy amongst the ELM function and Table 5.6. bellows shows the time difference between the two ELM functions. Figure 5.3 depicted below shows the comparison of ELM functions using medical dataset.

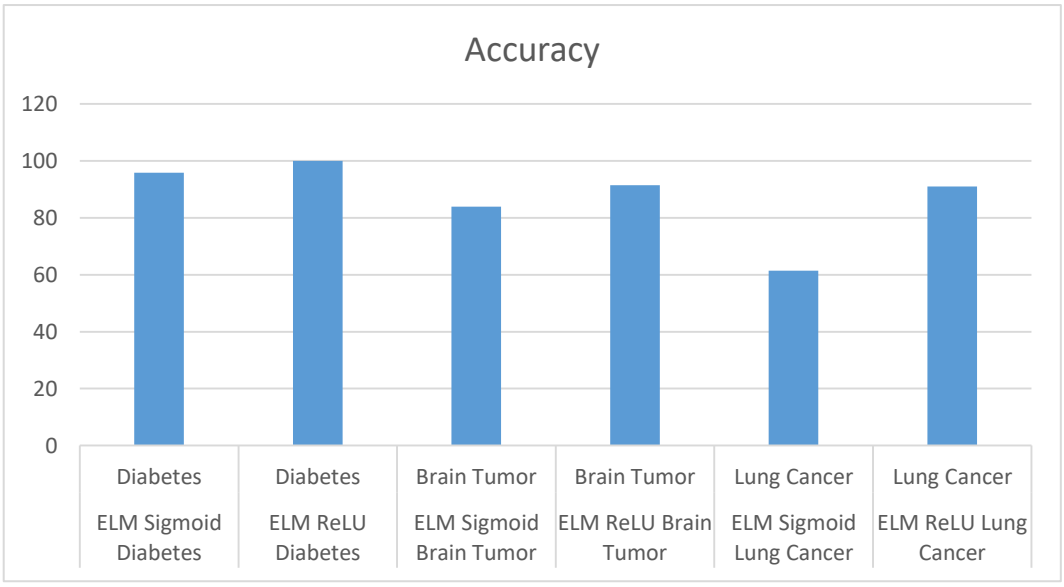

Figure 5.3. Comparison of ELM Functions Using Medical dataset

Table 5.6. ELM functions and other classifiers results using medical dataset

Method	Dataset	Accuracy	Compile Time
ELM ReLU	Diabetes	1	0.02
ELM Sigmoid	Diabetes	0.959	0.02
Support Vector Machine	Diabetes	0.74	0.033
K-Neighbor Classifier	Diabetes	0.714	0.021
Logistic Regression	Diabetes	0.746	0.22
Bagging Classifier	Diabetes	0.753	0.28
Naïve Bayes	Diabetes	0.77	0.27
SGD Classifier	Diabetes	0.61	0.23
ELM ReLU	Brain	0.915	0.455
ELM Sigmoid	Brain	0.839	0.22
Support Vector Machine	Brain	0.755	1.01
K-Neighbor Classifier	Brain	0.784	0.49
Logistic Regression	Brain	0.782	0.494
Bagging Classifier	Brain	0.9	0.486
Naïve Bayes	Brain	0.773	0.499
SGD Classifier	Brain	0.384	0.494
ELM ReLU	Lung	0.91	0.464
ELM Sigmoid	Lung	0.614	0.004
Support Vector Machine	Lung	0.634	0.623
K-Neighbor Classifier	Lung	0.642	0.623
Logistic Regression	Lung	0.641	0.664
Bagging Classifier	Lung	0.677	0.462
Naïve Bayes	Lung	0.574	0.435
SGD Classifier	Lung	0.634	0.799

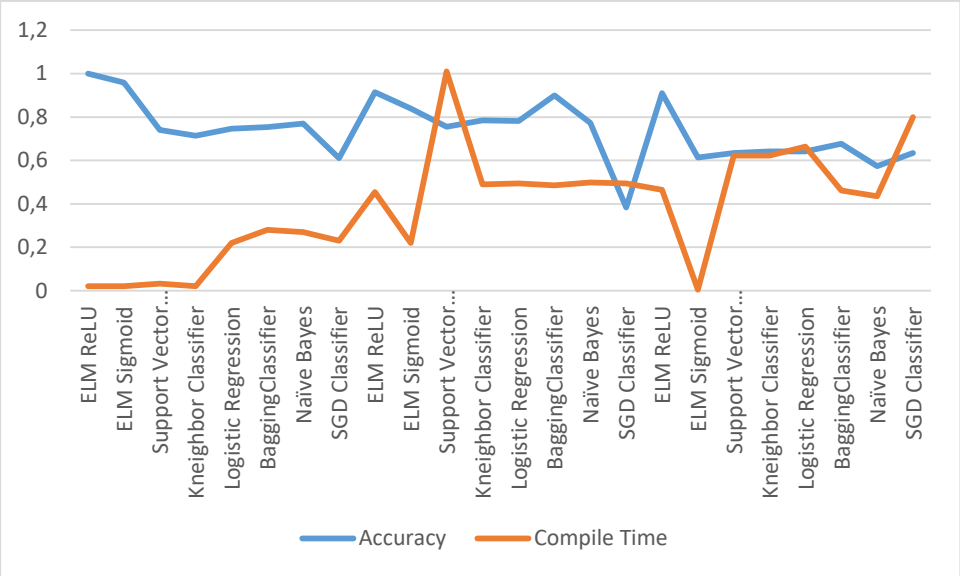

Figure 5.4. Result of ELM functions and other classifiers using medical dataset

Figure 5.4 above gives the result of ELM functions and other classifiers using medical dataset.

5.3.4. Comparison of categorization accuracy and time commitment

Finally, the overall objective of the study is to bring an experimental result to ELM functions accuracy and time fastness comparable to other classifier using medical datasets and we have achieved our objective. Table 5.6. shows the complete of the experiment conducted along its respective values for each function. Figure 5.4. displayed the good characteristics feature of time and accuracy for the attributes and it shows a good performance.

6. CONCLUSIONS

An extreme learning machine (ELM) is a simple and effective learning method for single-hidden layer feedforward neural networks (SLFNs). Three key differences separate ELM from other gradient-based learning techniques for feedforward neural networks:

ELM learns quickly. In our simulations, many applications can learn ELM in a few seconds. In the past, most (if not all) conventional learning algorithms couldn't cross a virtual speed barrier, and it's not unusual for a feedforward network to take a long time to train using classic learning methods, even for simple applications.

2. In most cases, ELM outperforms gradient-based learning techniques like backpropagation.

3. Gradient-based learning algorithms may encounter local minima, a low learning rate, and overfitting. To address these issues, traditional learning algorithms may need weight decay and early stopping. ELM often finds simple, barrier-free solutions. ELM learning seems simpler than other feedforward neural network strategies.

4. ELM learning can train SLFNs using non-differentiable activation functions, which is unlike gradient-based learning [15]. The performance of SLFNs with different activation functions is compared, but it's beyond the scope of this study. Compared with sigmoidal and RBF activation functions [13, 14], ELM could be interesting.

The recommended ELM strategy is only suitable for feedforward neural networks with a single hidden layer, but gradient-based learning techniques such as backpropagation may be used for networks with multiple hidden layers (SLFNs) [12]. SLFNs can approximate any continuous function and perform any classification. The recommended ELM method can be used in many applications. BP is usually faster than ELM at testing. In our simulations, we compared the ELM and the SVM and found that the ELM can learn thousands of times faster than the SVM. SVMs are another learning method. As shown in our simulations, trained SVM algorithms produce a significantly higher number of support vectors (computation units) than feedforward neural networks do for the same applications. This makes trained SVM algorithms slower to respond to new external observations than feedforward neural networks. In this application, it's difficult for SVMs to provide a real-time forecast because testing can take several hours, but the ELM seems to be enough when rapid prediction and response time are needed.

RECOMMENDATIONS

ELM, a simple and effective learning method for single-hidden layer feedforward neural networks, is recommended for widespread use in the medical community (SLFNs). In contrast to traditional gradient-based learning methods, which are typically used when time is of the essence when training feedforward neural networks, the proposed ELM takes a more leisurely and methodical approach.

REFERENCES

[1] Li, Bin & Li, Yibin & Rong, Xuewen. (2013). The extreme learning machine learning algorithm with tunable activation function. Neural computing and applications. 22. 10.1007/s00521-012-0858-9.

[2] Maciejewska, Monika & Szczurek, Andrzej & Ochromowicz, Ł. (2009). The characteristics of a "stop–flow" mode of sensor array operation using data with the best classification performance. Sensors and actuators b-chemical - sensor actuator b-chem. 141. 417-423. 10.1016/j.snb.2009.07.018.

[3] Damasceno, Daniela & Theodosia, Cristian & van den Bossche, Wouter Jose Luis & Van Dongen, JJM. & Orfao, Alberto. (2019). Distribution of subsets of blood monocytes cells throughout life. Journal of allergy and clinical immunology. 144. 10.1016/j.jaci.2019.02.030.

[4] Yeh, Wei-Chih & Le, Trung-Hieu. (2021). Training neural network. 10.1016/b978-0-323-90198-7.00007-0.

[5] The c4.5 algorithm j48 is used in numerous domains for data classification

[6] Man, Zhihong & Lee, Kevin & Wang, Dianhui & Cao, Zhenwei & Miao, Chunyan. (2011). A modified elm algorithm for single-hidden layer feedforward neural networks with linear nodes. Proceedings of the 2011 6th ieee conference on industrial electronics and applications, iciea 2011. 2524-2529. 10.1109/iciea.2011.5976017.

[7] Kumar, Rakesh. (2016). Human brain and artificial neural network. International journal of computer trends and technology. 33. 38-39. 10.14445/22312803/ijctt-v33p108.

[8] Shulyak, Oleksandr & Mnevets, Anton. (2022). Detalization of recognition algorithms in diagnosing patients and evaluating their effectiveness. Bulletin of kyiv polytechnic institute. Series instrument making. 82-89. 10.20535/1970.63(1).2022.260650.

[9] Yu, Wenchao & Zhuang, Fuzhen & He, Qing & Shi, Zhongzhi. (2015). Learning deep representations via extreme learning machines. Neurocomputing. 149. 308-315. 10.1016/j.neucom.2014.03.077.

[10] Yingu, Gao & Shengli, Xie. (2005). An improved form of the elms algorithm. Systems engineering and electronics, journal of. 16. 433-436.

[11] Liang Ny, Huang Gb, Saratchandran P, Sundararajan n. A fast and accurate online sequential learning algorithm for feedforward networks. Ieee trans neural netw. 2006 nov;17(6):1411-23. Doi: 10.1109/tnn.2006.880583. Pmid: 17131657.

[12] https://www.kaggle.com/datasets/mathchi/diabetes-data-set

[13] Wada, A & Tateishi, R & Sugiyama, T & Matsuda, M & Hattori, M. (1965). [lung cancer occuring in chronic pulmonary diseases]. Gan no rinsho. Japan journal of cancer clinics. 11. 637-42.

[14] Khellal, h. Ma and q. Fei, "ensemble of extreme learning machines for regression," 2018 ieee 7th data driven control and learning systems conference (ddcls), 2018, pp. 1052-1057, doi: 10.1109/ddcls.2018.8515915.

[15] Márquez-Medina, Diego & Figueiredo, Ana & Ferreira, Catarina & Ulisses, Pedro. (2015). Lung cancer overview.

[16] Rasmussen, Holger. (1987). Brain tumors can be subtle. The western journal of medicine. 147. 473.

[17] Xu, Yuesheng & Zhang, Haizhang. (2022). Convergence of deep convolutional neural networks. Neural networks : the official journal of [21] s. Baraha and p. K. Biswal, "implementation of activation functions for elm based classifiers," 2017 international conference on wireless communications, signal processing and networking (wispnet), 2017, pp. 1038-1042, doi: 10.1109/wispnet.2017.8299920.

[18] S. Ismaeel, a. Miri and d. Chourishi, "using the extreme learning machine (elm) technique for heart disease diagnosis," 2015 ieee canada international humanitarian technology conference (ihtc2015), 2015, pp. 1-3, doi: 10.1109/ihtc.2015.7238043.

[19] A. K. Alshamiri, a. Singh and b. R. Surampudi, "comparative analysis of elm and no-prop algorithms," 2016 ninth international conference on contemporary computing (ic3), 2016, pp. 1-5, doi: 10.1109/ic3.2016.7880217.

[20] T. Chouikh, h. Hellara, r. Barioul, f. Sandid and o. Kanoun, "comparative study of optimization methods proposed for the extreme learning machine (elm)," 2021 international workshop on impedance spectroscopy (iwis), 2021, pp. 137-140, doi: 10.1109/iwis54661.2021.9711902.

[21] L. Yang and q. Zhao, "comparisons of various elm based multi-view clustering methods for wdsns," 2019 ieee canadian conference of electrical and computer engineering (ccece), 2019, pp. 1-4, doi: 10.1109/ccece.2019.8861719.

[22] Torres, Renato & Ohashi, Orlando & Pessin, Gustavo. (2019). A machine-learning approach to distinguish passengers and drivers reading while driving. Sensors. 19. 3174. 10.3390/s19143174.[x]

[23] Farid, Nahla & Elbagoury, Bassant & Roushdy, Mohamed & M.Salem, Abdel-Badeeh. (2013). A comparative analysis for support vector machines for stroke patients.[y]

[24] Peycelon, R & Correard, R. (1970). [surgical treatment of sigmoiditis. Results. Indications. Apropos of 45 cases]. Annales de chirurgie. 24. 1159-64.

[25] Chen, Sitan & Klivans, Adam & Meka, Raghu. (2021). Efficiently learning any one hidden layer relu network from queries.

[26] Aguirre, Diego & Fuentes, Olac. (2019). Improving weight initialization of relu and output layers. 10.1007/978-3-030-30484-3_15.

[27] Feng, Lang & lLu, Qianhui & Tang, Huajin & Ma, de & Pan, Gang. (2022). Multi-level firing with spiking ds-resnet: enabling better and deeper directly-trained spiking neural networks. 2446-2452. 10.24963/ijcai.2022/340.

[28] Ismaeel, Salam, Ali Miri, And Dharmendra Chourishi. “using the extreme learning machine (elm) technique for heart disease diagnosis.” 2015 ieee canada international humanitarian technology conference (ihtc2015) (2015): n. Pag. Web.

[29] Torres, Renato & Ohashi, Orlando & Pessin, Gustavo. (2019). A machine-learning approach to distinguish passengers and drivers reading while driving. Sensors. 19. 3174. 10.3390/s19143174.

[30] Ruiz-gonzalez, Ruben & Gomez-gil, Jaime & Gomez-gil, Francisco & Martínez-martínez, Víctor. (2014). An svm-based classifier for estimating the state of various rotating components in agro-industrial machinery with a vibration signal acquired from a single point on the machine chassis. Sensors (basel, switzerland). 14. 20713-35. 10.3390/s141120713.

[31] https://www.javatpoint.com/artificial-neural-network

[32] https://www.analyticsvidhya.com/blog/2014/10/ann-work-simplified/

[33] Chen, Shu & Ma, Ling & Cen, Shan. (2020). Progress of slfn family proteins in tumor and virus infection. Hereditas (beijing). 42. 444-451. 10.16288/j.yczz.19-311.

[34] https://vitiya99.medium.com/what-is-a-decision-tree-in-ml-5bd76efc2232

[35] https://www.kaggle.com/datasets/mathchi/diabetes-data-set

[36] Liu, Xinwang & Wang, Lei & Huang, Guang-Bin & Zhang, Jian & Yin, Jianping. (2013). Multiple kernel extreme learning machine. Neurocomputing. 149. 10.1016/j.neucom.2013.09.072.

[37] Ismaeel, Salam & Miri, Ali & Chourishi, Dharmendra. (2015). Using the extreme learning machine (elm) technique for heart disease diagnosis. 10.1109/ihtc.2015.7238043.

[38] Huang, Guang-Bin & Ding, Xiaojian & Zhou, Hongming. (2010). Optimization method based extreme learning machine for classification. Neurocomputing. 74. 155-163. 10.1016/j.neucom.2010.02.019.

[39] Bacanin, Nebojsa & Stoean, Catalin & Zivkovic, Miodrag & Jovanovic, Dijana & Antonijevic, Milos & Mladenovic, Djordje. (2022). Multi-swarm algorithm for extreme learning machine optimization. Sensors. 22. 4204. 10.3390/s22114204.

[40] Wibawa, Helmie & Malik, Indra & Bahtiar, Nurdin. (2018). Evaluation of kernel-based extreme learning machine performance for prediction of chronic kidney disease. 1-4. 10.1109/icicos.2018.8621762.