THE RNN-CNN BASED CANCER PREDICTION MODEL FOR GENE EXPRESSION

Thesis Submitted for the Award of the Degree of

DOCTOR OF PHILOSOPHY

in

Computer Science and Engineering

By

Tanima Thakur

Registration Number: 41800130

Supervised By
Dr Isha Batra (17451)
Computer Science and Engineering
(Professor)
Lovely Professional University

LOVELY PROFESSIONAL UNIVERSITY, PUNJAB
2024

DECLARATION

I, Tanima Thakur hereby declared that the presented work in the thesis entitled "The RNN-CNN based Cancer Prediction Model for Gene Expression" in fulfilment of degree of **Doctor of Philosophy (Ph. D.)** is outcome of research work carried out by me under the supervision of Dr. Isha Batra, working as Professor, in the School of Computer Science and Engineering of Lovely Professional University, Punjab, India. In keeping with general practice of reporting scientific observations, due acknowledgements have been made whenever work described here has been based on findings of other investigator. This work has not been submitted in part or full to any other University or Institute for the award of any degree.

(Signature of Scholar)

Name of the Scholar: Tanima Thakur

Registration No: 41800130

Department/School: School of Computer Science and Engineering

Lovely Professional University, Punjab, India

CERTIFICATE

This is to certify that the work reported in the Ph. D. thesis entitled "The RNN-CNN Based Cancer Prediction Model for Gene Expression" submitted in fulfillment of the requirement for the reward of degree of **Doctor of Philosophy (Ph.D.)** in the School of Computer Science and Engineering, is a research work carried out by Tanima Thakur, 41800130, is bonafide record of her original work carried out under my supervision and that no part of thesis has been submitted for any other degree, diploma or equivalent course.

(Signature of Supervisor)

Name of supervisor: Dr. Isha Batra

Designation: Professor

Department/school: School of Computer Science and Engineering

University: Lovely Professional University

ABSTRACT

The term Artificial Intelligence (AI) refers to the intelligence which is man-made. It means that it is the technology with the help of which artificial machines can be made that can pretend to work like humans without any biological requirement. There are various applications of AI such as gaming, intelligent robots, natural language processing, pattern recognition, speech recognition, and many more. The main motive of AI is to use human intelligence in such a way that machines can use it very easily to solve problems from simple to complex levels.

Machine learning (ML) is the subsector of AI and similarly deep learning (DL) is the subsector of ML. These fields provide solutions to various real-life problems including healthcare. Healthcare is one of those areas that need proper attention. In healthcare, deep learning has applications in medical imaging, electronic health records, genomics, drug development, and many more. Convolutional neural networks (CNN) and recurrent neural networks (RNN) are two of the deep learning methods utilized for disease prediction. Machine learning models are also used for such problems. However, there exist some disadvantages of machine learning approaches over deep learning approaches. Machine learning approaches are not suitable for very large amounts of data having a greater number of dimensions because they may suffer from the overfitting problem.

Among various diseases, cancer is one of the most fatal diseases. However, if it is timely detected and treated, it can increase the probability of the patient's recovery. It can be detected after performing various medical tests. But at the same time, this requires time, money, and resources. Well-timed detection and treatment of cancer is also possible by diagnosing it at the genetic level. A larger number of features, or genes, are present in gene expression data despite the small sample size. That is why gene expression data suffers from a problem of high dimensionality. This problem needs to be handled well for further classification.

Therefore, this research work has been conducted so that gene expression data can be handled properly by reducing the dimensionality of the data and then different types of cancer have been classified. To achieve this, the whole research work is done to (1)

study and analyze the existing deep learning approaches to address the challenges. (2) develop suitable Feature Extraction Method for Cancer Prediction. (3) design the hybrid model for cancer prediction using RNN and CNN for gene expression. (4) compare and evaluate the proposed hybrid model with existing approaches using standard metrics.

Computational analysis, artificial neural networks (ANN), principal component analysis (PCA), fisher analysis (FA), support vector machines (SVM), particle swarm optimization (PSO), recurrent neural networks (RNN), lightweight CNN, CNN, long short-term memory (LSTM) neural networks, and many more are studied in order to meet the first objective. Some of the machine learning methods namely gradient boosting, random forest, logistic regression, support vector, and extreme gradient boosting (XGB), and deep learning methods such as Visual Geometry Group with 16 and 19 layers (VGG16 and VGG19), simple RNN, Residual Network with 50 layers (ResNet50), LSTM, Inception V3, and gated recurrent unit (GRU) from literature are also compared by performing classification on one dataset which consists of 5 different classes: LUAD (lung adenocarcinoma), LUSC (lung squamous cell carcinoma), UCEC (uterine corpus endometrial carcinoma), BRCA (breast invasive carcinoma), and KIRC (kidney renal clear cell carcinoma).

Among machine learning models, extreme gradient boosting performs the best concerning various performance metrics. Among given convolution neural network models, VGG16 performs the best and for recurrent neural network models, gated recurrent unit performs the best. From the literature, it has been found that deep learning models are more effective than machine learning models on gene expression data.

As discussed above, gene expression data suffers from a problem of high dimensionality so it must be handled well. So, to tackle this problem, we proposed a method known as sandwich stacked bottleneck feature extraction. From the literature, it has been found that pre-trained models VGG16 and VGG19 perform better feature extraction than other techniques. When the number of samples is less in the dataset then pre-trained models are more useful, otherwise, deep learning approaches require a large amount of data to get trained. Gene expression data consists of a lesser number of samples, so we decided to use pre-trained models for feature extraction.

The VGG19 model is assembled between two VGG16 models in the proposed architecture. The extracted features from VGG16 are passed to VGG19 and then to

again VGG16 to extract the most prominent features. The presented feature extractor is correlated with current feature extractors namely VGG16, Inception V3, VGG19, and ResNet50. Various performance metrics namely accuracy, precision, recall, f1 score, and mse are used to evaluate the presented method on three different datasets. One dataset contains five classes of cancer as mentioned above. Dataset 2 also contains five classes of cancer: PRAD (prostate adenocarcinoma), KIRC, LUAD, COAD (colon adenocarcinoma), and BRCA. Basal, HER, Luminal_A, Luminal_B, Cell_Line, and Normal are the six classes in total that are present in Dataset 3. The proposed method gives the maximum accuracy value of 0.954 on dataset 1, 0.931 on dataset 2, and 0.967 on dataset 3 among all the other existing models. However, on dataset 1, theproposed model yields the lowest MSE of 0.187, 0.186 on dataset 2, and 0.032 on dataset 3, which makes the presented feature extractor the best among the given methods.

The third objective is to predict cancer from gene expression data using the classifier. From the literature survey done so far, it has been found that convolution neural networks and recurrent neural networks perform better classification than existing techniques. For this, a novel method based on these two methods is developed. The extracted features from the dataset using the bottleneck feature extractor are passed to the proposed classifier. Three convolution blocks, one LSTM layer, and one simple RNN layer make up the suggested classifier.

The provided classifier is compared with many state-of-the-art deep learning classifiers, including VGG16, VGG19, ResNet50, Inception V3, and MobileNet, to assess its performance. It has been evaluated using above said performance measures. The proposed classifier has been evaluated on three different datasets. On dataset 1, it yields the maximum accuracy of 0.995, 0.994 on dataset 2, and 0.924 on dataset 3. On the other hand, it gives the lowest MSE of 0.029 on dataset 1, 0.006 on dataset 2, and 0.125 on dataset 3. From the research work carried out, it has been found that both the proposed models performed best among existing deep-learning models in terms of feature extraction and cancer classification.

ACKNOWLEDGEMENT

I would like to present my deepest gratitude to Dr. Isha Batra for her guidance, advice, understanding and supervision throughout the development of this thesis and study. Despite her busy schedule she has been available at every step, devoting time and energy and the much-needed counsel and advice. This enabled me to sail through the tough times and complete this enormous task.

I would like to thank the research project committee members for their valuable comments and discussions. A special thanks to the management of Lovely Professional University for their support in academic concerns and letting me involve in research study. The doctoral programme of LPU has made it possible for me to pursue my dream of research and upgrade my knowledge.

My sincere feeling of gratefulness also goes to my parents and family members who always motivated me in all the endeavors of my life including this research work in LPU. I am also thankful to my husband Sumit Kumar for offering full support to me during the entire period of my research work. Finally, I would like to thank each and every person who has directly and indirectly helped and motivated me in this journey.

TANIMA THAKUR

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVIATIONS

Abbreviation	Description
A	Adenine
AAE	Adversarial Autoencoder
ADA	AdaBoost
AE	Auto-Encoder
AGI	Artificial General Intelligence
AI	Artificial Intelligence
ALL	Acute Lymphoblastic Leukemia
AML	Acute Myeloid Leukemia
ANN	Artificial Neural Network
ANNs	Artificial Neural Networks
AS	Attribute Selection
ASR	Automatic Speech Recognition
BGA	Between Group Analysis
BGA	Between Group Analysis
BN	Bottleneck features
BPSO	Binary Particle Swarm Optimization
BRCA	Breast Invasive Carcinoma
C	Cytosine
CALGB	Cancer and Leukemia Group B
cDNA	Complementary DNA
CNAs	Copy Number Alterations
CNN	Convolution Neural Network
CNS	Central Nervous System
COAD	Colon Adenocarcinoma
CRC	Colorectal Cancer
CSE	Consistency-Based Subset Evaluation
DAI	Disagreement and Agreement Index
DBPs	DNA-Binding Proteins
DEGs	Differentially Expressed Genes
DFNForest	Deep Flexible Neural Forest
DHPCA	Dual Hypergraph Regularized PCA
DL	Deep Learning
DNA	Deoxyribonucleic Acid
DNN	Deep Neural Network
DRBPs	Both DNA RNA Binding Proteins
DT	Decision Tree
DWT	Discrete Wavelet Transform
ECDHE	Elliptic Curve with Diffie-Helmen and ephemeral key
FA	Fisher Analysis
G	Guanine
GA	Genetic Algorithm
GB	Gradient Boosting
GBDT	Gradient Boosting Decision Tree

GBM	Glioblastoma multiforme
GeneXNet	GeneeXpression Network
GO	Gene Ontology
GPGPU	General Purpose GPU
GPUs	Graphical Processing Units
GRU	Gated Recurrent Unit
GSVD	Generalized Singular Value Decomposition
HiGCN	Hierarchical Graph Convolution Network
IBGAFG	Improved Binary Genetic Algorithm with Feature Granulation
IFS	Incremental Feature Selection
INRSG	Improved Neighbourhood Rough Set with Sample Granulation
KIRC	Kidney Renal Cell Carcinoma
KNN	K Nearest Neighbor
KPLS	Kernel Partial Least Squares
LDA	Linear Discriminant Analysis
LINCS	Library of Integrated Network-Based Cellular Signatures
LISP	List Programming Language
LR	Logistic Regression
LSLS	Locality Sensitive Laplacian Score
LSTM	Long Short-Term Memory
LUAD	Lung Adenocarcinoma
LUSC	Lung Squamous Cell Carcinoma
MDNNMD	Multimodal Deep Neural Network with Multi-Dimensional Data
ML	Machine Learning
MLP	Multiple Layers Perceptron
MMC	Maximum Margin Criteria
MOPSOHA	Multi-Objective Particle Swarm Optimization
MRF	Markov Random Fields
mRMR	Minimum Redundancy Maximum Relevance
mRNA	messenger RNA
NB	Naïve Bayes
NLP	Natural Language Processing
NMF	Nonnegative Matrix Factorization
NN	Neural Network
OCR	Optimal Character Recognition
PCA	Principal Component Analysis
PLS	Partial Least Squares
PLS	Partial Least Square
PLS	Partial Least Square
PNN	Probabilistic Neural Network
PRAD	Prostate Adenocarcinoma
PSO	Particle Swarm Optimization
RBPs	RNA-Binding Proteins
ResNet50	Residual Network 50
RF	Random Forest
RNA	Ribonucleic Acid

RNN	Recurrent Neural Network
RPCA	Robust Principal Component Analysis
RUR	Rossum's Universal Robots
SBN	Stacked bottleneck
SGD	Stochastic Gradient Descent
SHAP	Shapely Additive explanation
SMOTE	Synthetic minority oversampling technique
SOM	Self-Organizing Map
SRBCTs	Small, Round Blue Cell Tumors
ST	Single Transcription
SVC	Support Vector Classifier
SVM	Support Vector Machine
SVM-RFE	Support Vector Machine-Recursive Feature Elimination
T	Thymine
TCGA	The Cancer Genome Atlas
TFs	Transcription Factors
tRNA	transfer RNA
U	Uracil
UCEC	Uterine Corpus Endometrial Carcinoma
UDA	Uncorrelated Discriminant Analysis
UT	Unlimited Transcriptions
VGG16	Visual Geometry Group 16
VGG19	Visual Geometry Group 19
XGB	Extreme Gradient Boosting
XOR	Exclusive-OR

CHAPTER 1

INTRODUCTION

By grasping the fundamental ideas of deep learning and machine learning, this chapter develops the information needed to comprehend the research work's issue statement. This chapter gives an introduction to the present work. It explains gene expression, artificial intelligence, the history of artificial intelligence, and much more about machine learning, deeplearning, and their various techniques. It also gives information about the thesis organization.

1.1 Gene Expression

Transcribing DNA sequences into RNA is known as a gene expression which is later used for protein composition [1]. The method by which the gene's DNA or the genome code is used to compose protein and to produce a cell's structure is known as gene expression. There are two main key stages for gene expression:

- **Transcription:** It is the first stage of gene expression. The messenger RNA (mRNA) is produced by RNA enzyme, polymerases to form an RNA strand.
- **Translation:** It is the stage in which mRNA is used for direct protein formation and further processing of the protein molecule [2].

Turning the information contained in a gene into an end gene product, such as transfer RNA (tRNA), ribosomal RNA, small nuclear RNA, or protein, is a crucial process known as gene expression. There are several sub-processes involved, including initiation, translation, termination, and post-translational processing. These processes are arranged in a sequence and include transcription and translation. The basis for numerous events involved in the evolution of life, including the growth and shape-changing and specialized tasks of basic cells, is known as gene expression. This process can be managed to adapt and obtain the desired functional protein. This process offers the framework for many other

vital procedures such as environmental adaptation, etc.

With the continuing development in technology, the relevance of gene expression is quickly across a range of life sciences. In general, this analysis is used to clarify the steps that must be undertaken to locate the target gene. It makes it possible for us to play around with different genes and the characteristics that they cause in an organism. It may be possible to create hybrid or altered organisms through certain gene expression processes. When identical cells are put together in a group, they behave as an organ because they possess the same goals [3].

1.1.1 Deoxyribose Nucleic Acid (DNA):

DNA holds the genetic architecture of an organism. It has the genetic information required to regulate an organism’s protein production. Nucleotides are DNA's fundamental building blocks. They consist of four nitrate groups: adenine (A), cytosine (C), thymine (T), and guanine (G), along with phosphates and deoxyribose sugars. These are usually passed on from parents to their children and hold the genetic details needed for the children to grow and develop. The characteristics of an organism are determined by the sequence in which these groups arrange themselves; this ordered arrangement is known as a gene, and it is necessary to produce proteins. The sugars and phosphates are linked together to form a double helical structure which is joined by nitrogenous base pairs. These DNA strands are long and are fitted into a structure called Chromosomes as shown in Fig 1.1. Humans contain 23 pairs of chromosomes in every cell.

1.1.1.1 Types of DNA:

- **A-DNA:** This particular form of DNA is a double helical in the right direction. When DNA is malnourished, dehydrated, or bombarded by greater ionic concentrations, it adopts this form.

- **B-DNA:** The DNA sequence, which has 10 bases per rotation, is what preserves it in its typical configuration beneath normal circumstances, where life flourishes.
- **C-DNA:** It is also referred to as complementary DNA and is produced by a unique process known as reverse transcription, which is assisted by the catalyst transcriptase.
- **D-DNA:** At present, very little is known about this highly unusual configuration.
- **Z-DNA:** The following kind of DNA is a left-handed double helical type. Higher salt concentrations drive DNA to take on this structure. Its structure is left-handed, yet it operates similarly to A-DNA.

1.1.2 Ribose Nucleic Acid (RNA):

One element that is principally in charge of protein production is RNA. Its structure is helical, with a single strand as shown in Figure 1.1. This allows it to fold readily upon itself to generate other compounds. Phosphates, ribose sugar, and four nitrogenous bases—adenine (A), guanine (G), cytosine (C), and uracil (U)—make up its composition.

1.1.2.1 Types of RNA:

- **tRNA:** t-RNA, an abbreviation for transfer RNA, is a type of molecule that is used to convert a messenger RNA (mRNA) into a protein. These carry amino acids to the end of the amino acid chain during the process of translation.
- **mRNA:** Messenger RNA, or mRNA for short, transports information from DNA to the cytoplasm, where it is converted into proteins.
- **rRNA:** Abbreviation for ribosomal RNA, or rRNA, ribosomes are components that synthesize proteins, which then undergo processing to become actual proteins.
- **snRNA:** Small nuclear RNA, or snRNA, is a vital component of RNA processing and the process of splicing introns. [4].

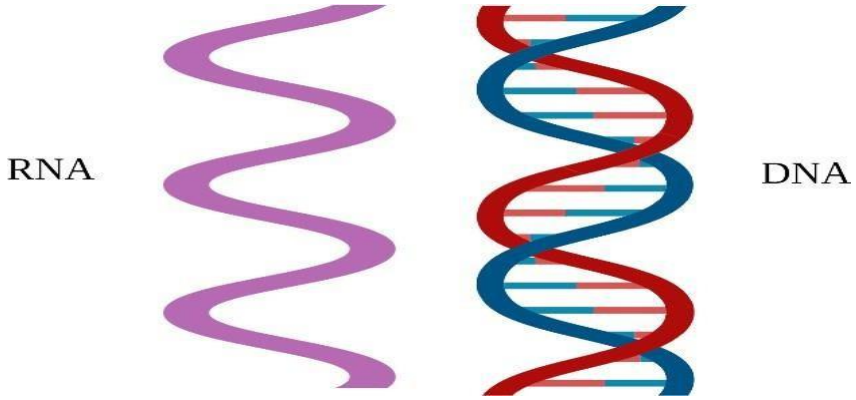

Fig 1.1 Structures of RNA and DNA

1.2 Artificial Intelligence

The engineering and science of developing intelligent machines, primarily computer programs with intelligence - John McCarthy

Artificial Intelligence is composed of two terms: artificial and intelligence. Artificial means that is man-made and intelligence means the property to learn something and use that knowledge to perform some tasks. With the help of artificial intelligence, a machine (computer, a robot, or any product) can think like a human being. Whenever a problem comes to the human, he follows one major approach having four main steps: think, learn, decide, and work. This study is called AI, and the final product of this study is the intelligent machine.

The term artificial intelligence refers to a technology that allows for the creation of fully artificial machines that can mimic human behavior without using any biological things as a basis for their growth.

There are certain applications of AI as shown in Fig 1.2:

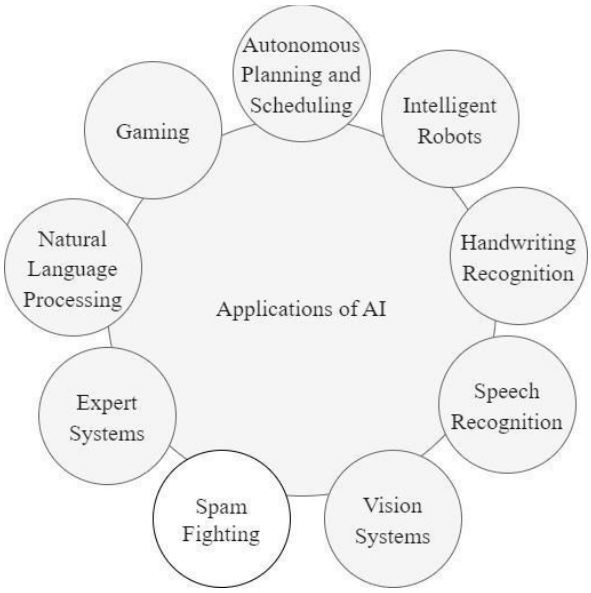

Fig 1.2: Applications of AI

- Gaming: AI is crucial for machines to generate a huge number of potential positions based on an in-depth understanding of strategic games. For instance, N- queen's issues, river crossing, chess, and so forth.
- Natural Language Processing: Interact with a machine that can comprehend human words.
- Expert Systems: The users receive explanations and recommendations from machines or software.
- Vision Systems: On the computer, systems comprehend, clarify, and describe visual input.
- Speech Recognition: Some AI-based voice recognition systems are capable of hearing what people are saying, expressing it as sentences, and comprehending what they are saying.
- Handwritten Recognition: The text written on paper is examined by the handwriting recognition program, which identifies the letter shapes, and translates the text into editable format.

- Intelligent Robots: Robots are capable of carrying out the commands supplied by humans [5].
- Autonomous planning and scheduling: There are various autonomous planning programs such as NASA's Remote Agent Program, Successor Program MAPGEN, and MEXAR2 which perform various scheduling tasks such as detecting, diagnosing, etc.
- Spam Fighting: Various ML and DL algorithms are in the market that are filtering messages and emails into ham and spam to save users time deleting them [56].

The method of developing machines in a purely artificial way that can behave like human beings without harming any living organism is known as Artificial Intelligence. These products are like humans and can behave like them such as having sentiments, predicting, and making decisions [6]. The intelligent system can learn from past experiences. Depending upon the learning experiences, when a new input is given to the intelligent system, the system adjusts itself accordingly and performs the task as humans do. Deep Learning and Natural Language Processing (NLP) play a very important role in day-to-day examples of AI. Computers are instructed depending upon these technologies to perform different activities by using immense amounts of data and also identify the patterns in the data to gain experience for future inputs. AI is important for multiple reasons. Some of these are listed below:

- Artificial Intelligence automates the processes of data-driven discovery and repetitive learning. While involving humans, AI accomplishes challenging automated jobs. The systems must be customized by people.
- AI making things that already exist smarter. Numerous technologies may be enhanced by combining automation, conversational platforms, bots, and intelligent robots with vast volumes of data.
- AI adapts by letting the data handle the programming by utilizing progressive learning techniques. Artificial Intelligence explores regularities and structures in data so that algorithms may learn.

- Massive amounts of data are analyzed by AI using neural networks with multiple hidden layers. Deep learning models require a large quantity of data to train because they derive their knowledge directly from the data.
- Deep neural networks achieve commendable accuracy in the medical field as well. Cancer can also be detected from medical images through AI techniques of deep learning [7].

Building intelligent computers that can carry out activities that need human intelligence is the ultimate goal of computer science's artificial intelligence (AI) field. Although artificial intelligence (AI) spans many academic fields, advancements in machine learning and deep learning are particularly promising in the technical domain. By deciphering the Nazi encryption device Engima, mathematician Alan Turing contributed to the victory of the Allied Forces in World War II and after this in less than adecade; he came up with the question “Can machines think?” The branch of computer science known as AI gives the answer to Turing’s question. It’s the attempt at replicating or simulating human intelligence in computers.

AI, as defined by Stuart Russell and Peter Norvig, is the study of agents that take in information from surroundings and act on it. Four different methods were used to define AI initially. Stuart Russell and Peter Norvig also inspected these methods. These are:

- Thinking humanly
- Thinking rationally
- Acting humanly
- Acting rationally

The two initial approaches are all about thought patterns and reasoning, and the next two are about behavioral patterns. Norvig and Russell are especially focused on rational agents who tried to obtain good output [8].

Table 1.1 shows different definitions of AI. The definitions just at the top focus on thinking and thought processes, whereas the definitions at the bottom deal with behavior. While definitions on the right assess performance in relation to an ideal performance criterion, those on the left examine performance in terms of fidelity to human performance.

Table 1.1: Definitions of AI

Thinking Humanly The intriguing new initiative to infuse computers with thoughts—that is, machines with minds in the truest sense of the word—Haugeland, 1985 the automation of cognitive functions such as learning, decision-making, and problem-solving (Bellman, 1978).	Thinking Rationally computational models are utilized to investigate mental faculties (Charniak and McDermott, 1985). the study of the equations needed for perception, reasoning, and action.(Winston, 1992).
Acting Humanly The method of building machines to carry out things that humans would typically accomplish with intelligence (Kurzweil, 1990) The study of programming computers to carry out tasks for which humans are now more proficient—Rich and Knight, 1991	Acting Rationally The discipline of developing intelligent agents is known as computational intelligence.-Poole et al., (1998) (Nilsson, 1998) Artificial Intelligence is the study of intelligent behavior in artifacts. [57]

The process with the help of which human intelligence is simulated in machines and the machines are coded in such a way that they can think and act like human beings is called Artificial Intelligence. AI is also used to refer to any machine that shows features that are related to the human mind such as thinking and problem solving. One of the roles of AI is to rationalize and make decisions that are used to attain a specific objective.

Whenever people hear about the term artificial intelligence, they start comparing the term with robots. The reason is that high commercial movies and novels present tales about machines that are like humans. There could be nothing further from the facts, however. The main objective of AI is to use human intelligence in such a way that machines can easily interpret it and can perform simple to simple and even complex tasks easily. There are three main goals of AI. These are learning, reasoning, and perception.

With the advent of technology, the initial specifications set for artificial intelligence have been outdated. Previously, the systems that were used to perform basic functions or that were used to identify the characters using Optimal Character Recognition (OCR) were considered as the Intelligent Systems. But later on, due to advancements, these features have been added to all the computers [9].

1.2.1 History of Artificial Intelligence

Artificial intelligence is influencing ever-larger areas of human life. AI chips are the latest craze with their applications in smartphones. Conversely, the early 1950s saw the introduction of technology at Dartmouth College in the United States through the Dartmouth Summer Research Project on AI. The source can be traced back to Allen Turing's work, which in turn can be traced back to Allen Newell and Hebert A. Simon's Turing Test. In 1996, IBM created the Deep Blue chess computer, which proved to be effective in defeating the renowned World Chess Champion of the time, Garry Kasparov, in a match. This made artificial intelligence grab popularity across the globe. AI techniques have been used for several years in data centers and on mainframes.

It is proven that in the ancient Greek age, diverse ideas were carried out about robots that have their appearance similar to human beings. An example of this is Daedelus, the king of mythology of that time, who tried to create an artificial human. The objective of explaining the philosopher's system of human thought in that era has given a spark to modern artificial intelligence. In 1884, Charles Babbage worked on one machine with the hope that that machine would act similarly to human beings. But with time after getting results, he suspended his work so that it would be impossible for him to develop a machine that would be behaving like human beings. But in 1950, Claude Shannon came up with the idea that a computer system can play a mess.

The evolution of artificial intelligence officially started in 1956. An AI conference took place in 1956 at Dartmouth College in the United States of America. The problem of artificial intelligence modeling will be addressed in a generation, according to Marvin Minsky, the author of Stormed Search for Artificial Intelligence. The first applications based on artificial intelligence using logic theorem and chess games were introduced during this time. The geometric shapes utilized in the intelligence test differed from the programs created in the same time frame. This finding contributed to the belief that intelligent systems can be developed.

To find out if a system is intelligent or not, Allen Turing experimented in 1950. In those days, passing the test indicated a sufficient level of intelligence. John McCarthy created the artificial intelligence functional programming language known as LISP (List Processing Language) in 1957. LISP is one of the oldest and most potent programming languages, which can be used to construct customizable programs that carry out simple tasks using list structures.

The period of 1965-1970 is known as the Dark Period for artificial intelligence due to the very little work in this field. Due to the impractical expectations, the abrupt and on the cloud nine kind of attitude has forced the mind to think that it would be so easy to discover systems with intelligence. But this duration was marked as the Dark Period due to the

unsuccessful idea of making a system intelligent by uploading data only. Artificial intelligence gained traction from 1970 through 1975. Thanks to the performance of artificial intelligence systems, built and built on topics such as diagnosis of disease, has been laid the cornerstone of present artificial intelligence. It was established between 1975 and 1980 that artificial intelligence could be backed up by other scientific fields such as psychology.

In the 1980s, artificial intelligence started to be used in large-scale initiatives with practical applications. Artificial intelligence has been altered to address problems in the actual world the next time the sun sets. Furthermore, more affordable technology and resources have made artificial intelligence applications possible, even in fields where traditional approaches are still able to meet client expectations.

The history of Artificial Intelligence in sequential order is given below:

- One of the forerunners of cybernetic science, Ebru Iz Bin Rezzaz Al Jezeri, developed water-powered automated controlled machines in 1206.
- Wilhelm Schickard in 1623 developed a mechanic and also one calculator that was used to perform four operations.
- Gottfried Leibniz invented a numbering system that is known as the binary numbering system in 1672. This binary system acts as a base for the computers that we are using today.
- Charles Babbage created a mechanical calculator between 1822 and 1859. During this period, Ada Lovelace developed so many algorithms on his machine using the punch cards of Babbage. Ada Lovelace is regarded as the first programmer asa result of this work.
- Karel Chapek introduced the idea of robots in 1923 at one of the performances held at Rossum's Universal Robots (RUR) theater.
- In 1931, the theory of deficiency was presented by Kurt Godel. In 1936, a programmable computer system, namely Z1 with a memory size of 64K was developed by Konrad Zuse.
- In 1946, the first computer, namely Electronic Numerical Integrator and Computer

has started working with a room size of 30 tons.

- The concept of self-duplicating programs has been presented by John Von Neumann in 1948.
- In 1950, the discoverer of computer science Allen Turing presented the Turing Test concept.
- In 1951, the first ever artificially intelligent programs were written for the Mark 1 computer system.
- The first artificial intelligence system, named the Logic Theorist (Logic Theory - LT), was created in 1956 by Neweell, Shaw, and Simon. This system was used to perform various mathematical calculations.
- LISP was developed by John MacCarty in 1958.
- JCR Licklider in 1960 presented the relationship between human beings and machines.
- In 1962, the company named Unimation was rooted. This company became the first company to produce robots that were used at the commercial level.
- In 1965, ELIZA, the first artificial intelligence program was written.
- The “Shakey” was developed at Standford University. It was known as the first animated robot.
- In 1973, the TCP/IP protocol was developed by Darpa.
- People started using the internet for the first time in 1974.
- Herbert Simon proposed Rationality Theory and won the Nobel Prize in 1978.
- The first personal computer was made by IBM in 1981.
- The production of a robot namely Cog was started in 1993. The robot was a look-alike human being.
- In 1997, popular world-level chess player Kasporav was defeated by the supercomputer namely Deep Blue.
- The very first player of artificial intelligence, Furby, was presented in the market in year 1998.
- The Kismet robot was introduced in 2000. This robot can use expressions and mimic motions during conversation.

- In 2005, a new robot named Asimo was developed. This robot can use the skills andabilities of human beings that make it the nearest robot to humans.
- Then in 2010, Asimo started using the power of mind [6].

1.2.2 Types of Artificial Intelligence

As human beings can do multiple tasks at the same time, similarly Artificial General Intelligence system (AGI) makes a machine capable of doing multiple jobs at a time. AI is oftwo types:

- Weak AI
- Strong AI

Weak AI: The main idea behind Weak AI is that the machines behave as if they are smart. With AI, it has already been proved that artificial qualities such as thought, speaking, and moving can be achieved by any system if it is programmed in this way. So that's why in the Chess game, the computer is responsible for all the moves and games. A computer can play and move the players accordingly. It doesn't mean that a machine can think and make decisions. The reason is that the machines are programmed or trained in this way sothat they always make the right movement.

Strong AI: The main logic for Strong AI is that, in the future, the machines may perform calculations, may think, and may also anticipate the answers. Example- IBM invented the artificially intelligent supercomputer, namely "WATSON". That's why we can predict that in the future there may be such systems or robots that will do their job and will be more powerful than human beings [10].

There are different types of AI based upon the capability and functionality of AI as shown in Fig 1.3 and Fig 1.4. The given figure represents the various types:

- Narrow AI: It is a kind of AI that can perform a committed task intelligently. Narrow AI is the most popular and currently accessible AI in the Artificial Intelligence environment. Because narrow AI was created solely to accomplish one particular task, it is unable to function properly outside of its bounds. This is why

it's often referred to as a weak AI. Narrow AI may fail in unforeseen ways if it is used outside its bounds. Speech recognition, image recognition, self-driving cars, and playing chess are some of the examples of Narrow AI.

- General AI: It is a kind of AI that can perform tasks in an efficient manner like human beings. The main principle behind the general AI is to create a system that can think like humans and can act smart. There is currently not a single system that lies in the category of General AI. The research is still going on to develop a General AI.
- Strong AI: It is a level of system intelligence at which a machine can overpower the intelligence of humans and can execute any job with cognitive propertiesbetter than humans. Strong AI is the end product of General AI. There are variousfeatures of AI such as thinking ability, decision-making, learning, planning, and communicating. Super AI appears to be a hypothetical Artificial Intelligence term. The development of these types of machines is still an evolving challenge for the world.
- Reactive Machines: These are the basic types of AI systems based on functionality. They cannot store previous experiences, and they can't even store their memories so that using those experiences they can act in future. These systems revert to the best solution for the current problem. One of the examples of Reactive Machines is the Deep Blue System developed by IBM.
- Limited Memory: Such type of AI system has a smaller memory that can store the details regarding recent activities for a smaller amount of time. Self-driving cars are one example of such a system. These vehicles may keep data about road navigation, including the speed limit, distance from other vehicles, and the recent speed of cars in the neighborhood.
- Theory of Mind: This type of AI system should grasp human feelings, personalities, and interests and be able to communicate socially like human beings. Although a great deal of research and development is still in progress, these kinds of AI robots have yet to be developed.

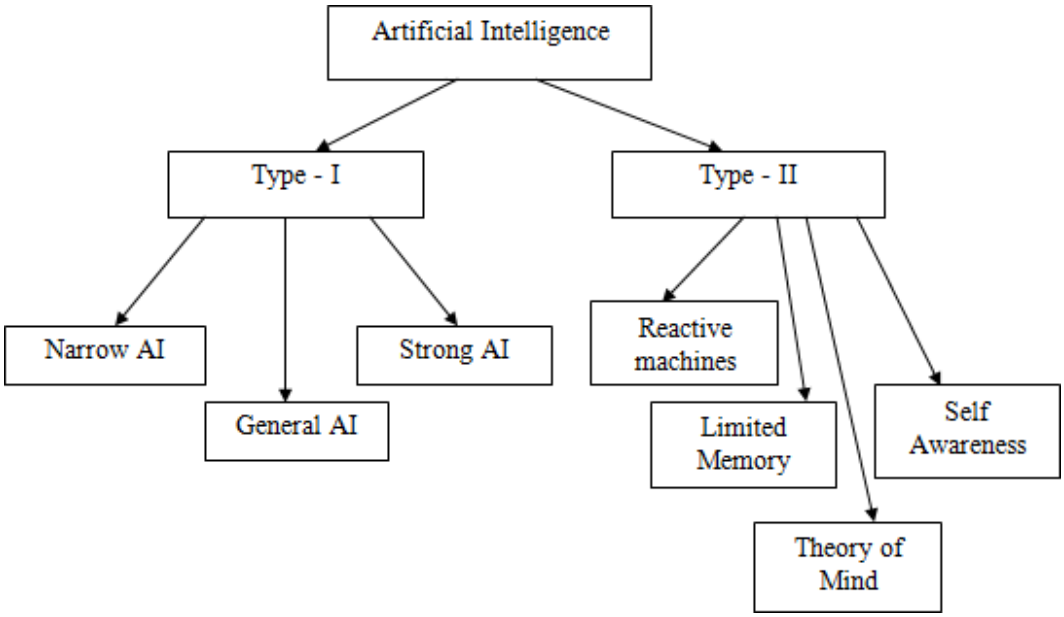

Fig 1.3: Types of AI

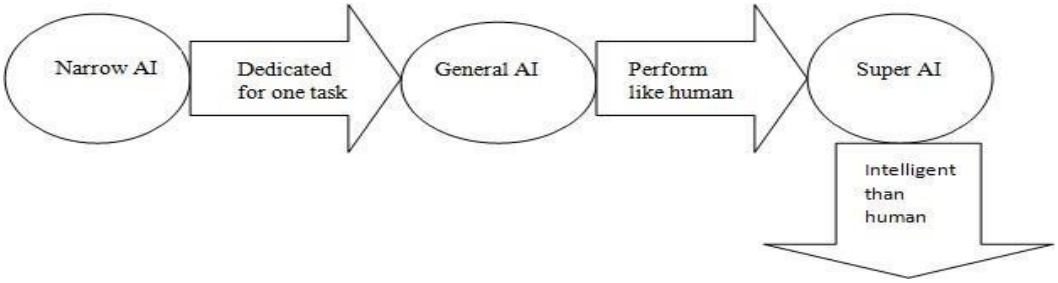

Fig 1.4 AI Based on Capabilities

- Self-Awareness: This is the future of AI. Such systems will be super smart and will be having their own consciousness, emotions, and awareness. These systems are going to be more intelligent than humans. Self-awareness AI does not exist, and it is purely a hypothetical term [11].

1.3 Machine Learning

The discovery of artificial intelligence can create a war between machines and their creators if we believe science fiction stories. Initially, a system could play some games like tic-tac-toe and chess. But with time we started giving the control of traffic lights as well as the control of military drones and missiles. When the computer would be having consciousness of them and would learn self-teaching, at that time the evolution of the machines would take a sinister turn. Also, there would be no need for human programmersand there would be no humanity.

Since the moment we were born, data has surrounded us. Raw data is constantly being received by the body's sensors, including the tongue, ears, nose, eyes, and nerves. Brain activity has changed this raw input into visuals, auditory, taste, smell, and tactile perceptions. Language had been utilized to express these feelings to other folks. Therefore, at first, the data was manually recorded. But these days, with so many technological breakthroughs, this information recording process is automated and digitized. For this, sensors are being employed. They work repeatedly without a break and never become tired like humans do. Data is recorded in one manner or another by government agencies, businesses, and even individual citizens. We have been surrounded by an abundance of data. We can also call it the era of Big Data. Because most of the data is easily obtainable with only a single browser click, a vast amount of data is readily available. We can make meaningful information out of that available data. Given this, machine learning (ML) is the field that focuses on creating algorithms that may transform data into actions and meaningful information. ML emerged in an area where the data, statistical methods, and computational power developed rapidly and simultaneously. Data growth requires additional computational power. This also encouraged the development of statistical methods that are required to analyze large datasets. It produced a period of growth, allowing for the collection of even larger and more important data. Fig 1.5 shows the cycle of advancement on these three parameters.

Data mining is closely related to machine learning, which is used to draw unique insights from huge databases. But still data mining is different from machine learning in some

contexts. Machine learning instructs computers on how to use data to solve problems. On the other hand, data mining trains machines to recognize patterns in data that humans can utilize to solve problems. Although not all machine learning applications use data mining, most machine learning applications make use of data mining. As an example, machine learning may be used to look for patterns of traffic data related to accident rates. On the other hand, this is simply machine learning and not data mining if a machine is teaching itself how to drive the car.

The very first computer to beat a world champion in chess is Deep Blue, and in the television game show Jeopardy, a computer named Watson beats two human competitors. Keeping in mind such miraculous performance of the machines, some people havethoughts that humans will be replaced by machines and robots in the fields and assembly line respectively, the same way humans will be replaced by computer intelligence in different IT applications as well.

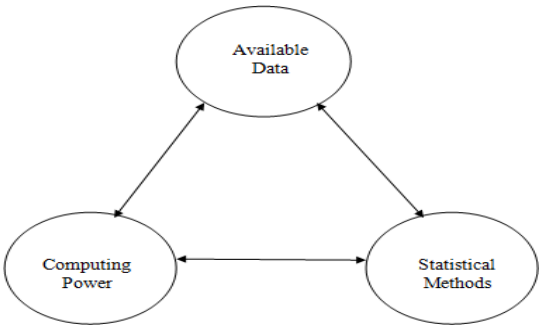

Fig 1.5 Cycle of Advancement in ML

But the reality is, even if the machines are much advanced still require human interference to perform tasks. For example, from the massive volume of data, a machine can identify significant patterns but still humans are required for the proper analysis of those patterns and also to convert those patterns into substantial results. Machines don't know what to ask or even how to ask anything. But if the questions are designed well, machines can provide answers to those questions.

1.3.1 Applications of Machine Learning

Almost all fields can benefit from machine learning. It simplifies the patient's treatment by the physicians. It supports the creation of smart houses by engineers and programmers. Furthermore, it aids in the development of intelligent societies by social workers. Given are some of the applications of ML:

- ML is used to find spam mail in e-mail.
- It is used to segregate the behavior of customers so that advertisements can be done according to their tastes.
- It also gets used in weather forecasting and to know about climate change.
- It also controls suspicious transactions on credit cards.
- It is used to estimate financial losses during storms and natural disasters.
- It is also used as a predictor for the election results.
- Different algorithms have been developed using machine learning for auto-piloting drones and self-cars.
- It is used to optimize the use of energy in homes and offices.
- It is used to identify the criminal areas.
- It is used to identify the genes that are responsible for any disease.

1.3.2 Learning Process of Machine Learning

According to the definition given by Tom M. Mitchell state for ML, a machine can learn by experiences so that the performance of a machine may be improved by using those experiences in the future. It means a machine has to learn. There are four basic steps for learning as in Fig 1.6; either it is a machine or a human. These are:

- Data Storage: It is a phase where the data is collected. It can be collected from any of the available sources. The stored data is processed further for reasoning.
- Abstraction: In this phase, the stored data has been converted into deeper illustrations and concepts.
- Generalization: It is used to create knowledge and assumptions from the data that is withdrawn at the abstraction level.

- Evaluation: Whatever is learned by a machine, this phase is used to identify the learned data and provide feedback accordingly. If any improvements are required, these are also being communicated.

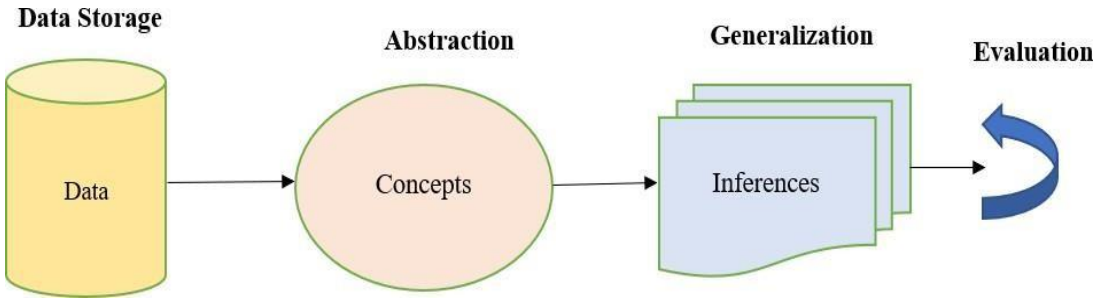

Fig 1.6 Learning Process

1.3.3 Steps to Apply Machine Learning Algorithm

There are mainly five steps as given in Fig 1.7 that are required to implement machine learning algorithms in real life. These are:

- Data Collection

  The data has to be collected which has to be used by the learning algorithm to get the information. The collected data can be organized in any of the following forms: text file, database, or spreadsheet.
- Data Exploration and Preparation

  The quality of the collected data must be good to get the fine results of the machine learning algorithm. This step includes cleaning of data and removing the unimportant data.
- Model Training

  After preparing the data, the next step is to select the task of machine learning which in turn is used to find the appropriate algorithm that is used to model the data.
- Model Evaluation

  After training the model, the next step is to test it on the dataset to find its accuracy. It is also very important to find in what manner the algorithm has learned.

- Model Improvement

  After evaluating the model, if improvements are required, then it is important to use the enhanced methods. It may be the possibility that after the evaluation process, switching to another model is the next step.

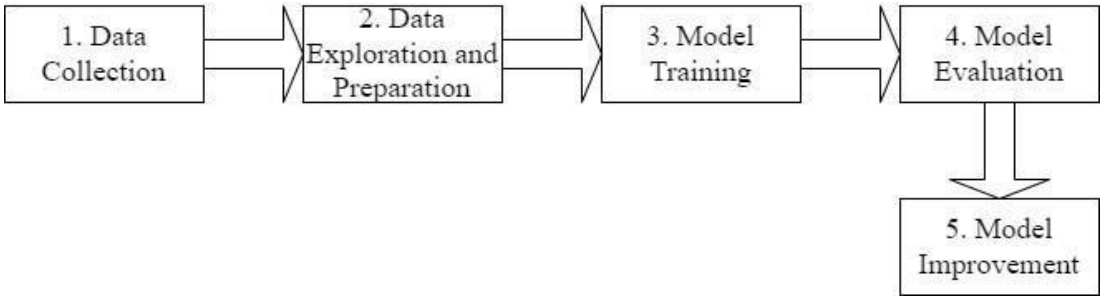

Fig 1.7 Steps to Apply Machine Learning

1.3.4 Types of Machine Learning Algorithms

Machine learning algorithms exist in several forms depending upon their functionality as shown in Fig 1.8.

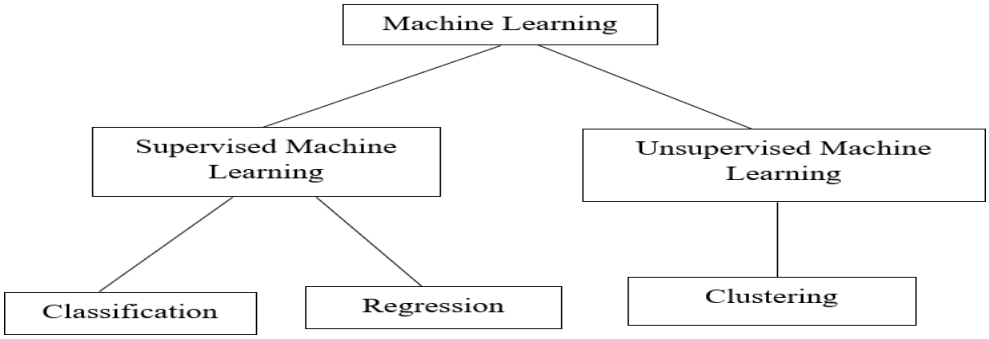

Fig 1.8 Types of Machine Learning

1.3.4.1 Supervised Learning

In a dataset, the predictive model is applied when one value is being used to predict another value. Finding the relationship between the target feature and the other independent features is done using the algorithm. The predictive model helps to make predictions for

the target feature. These models are not used to predict the upcoming or future values. But this model is used to predict past events such as the conceiving date of a mother can be predicted using the levels of the hormone of the current day. For managing traffic lights during peak rush hour, predictive models can also be employed. Supervised learning is the process of training a predictive model. Supervised learning means what and how to learnis already defined. Even though there is no human interaction in this type of supervision, the learner can determine how well the machine has learned. One of the methods used to determine which category an example belongs to is classification. In supervised learning, regression modeling—another technique for numerical prediction—is used to forecast numerical values. Fig 1.9 shows the process of supervised machine learning. The model is trained with features and labels and then evaluated with test data to make predictions.

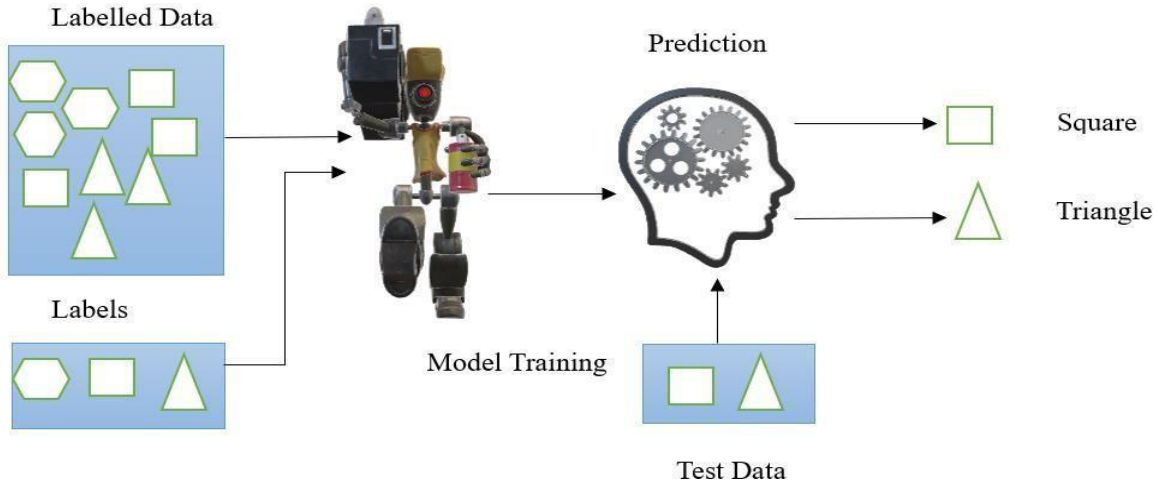

Fig 1.9 Supervised Machine Learning

1.3.4.2 Unsupervised Learning

Unlike predictive models, descriptive models are used to identify all the features instead of any single feature. Here all the features are very important. The training method used by the descriptive model is known as unsupervised learning. There is one method available for descriptive modeling known as pattern discovery. It is used for market basket analysis on the invested data by the retailer. This can be used to find out the interest of the persons visiting the store to increase the sale. For example, if it is found that along with the purchase

of the sunglasses, the customers are also interested in the swimming trunk, then the swimming trunk can be placed in a more approachable area to earn more profit. Clustering is one of the techniques for descriptive modeling that is used to divide the datainto the same type of groups known as clusters. In this case, a machine has created clustersbut still, to choose for a particular cluster human interference is required [12]. Fig 1.10 shows the clustering process. Unlabeled data has been given to the machine, unlike supervised machine learning. In the given example, some random data has been given to the machine and then three clusters are formed. In the first case, the cluster is based on color, and in the second, the cluster is based on shape.

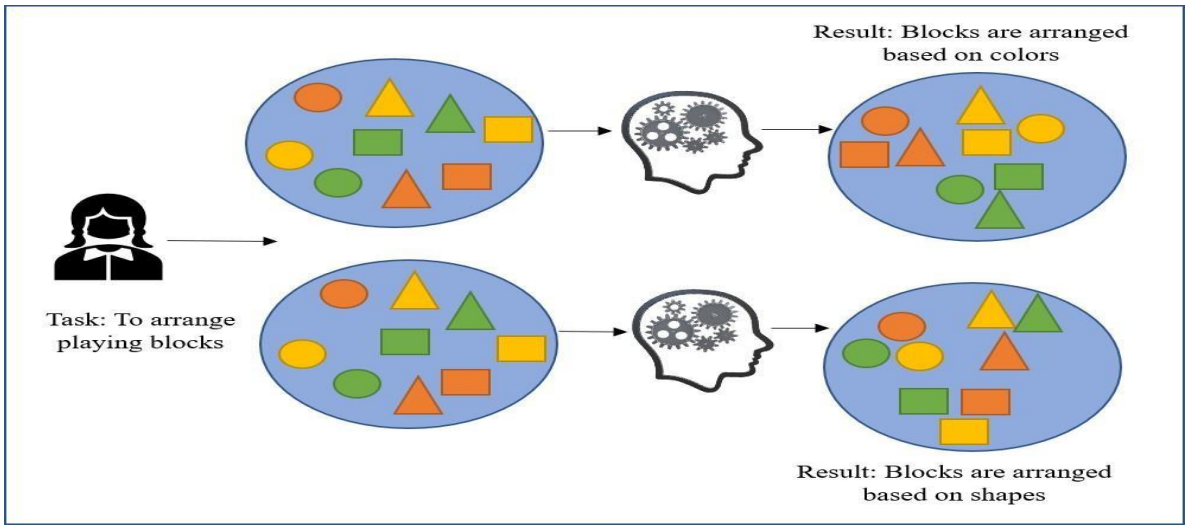

Fig 1.10 Clustering (Unsupervised Machine Learning)

1.4 Deep Learning

Deep learning is one field that emerged from machine learning. There are a plethora of applications for deep learning, involving cancer detection, elephant identification, and game production. The DL algorithms offer promising outcomes in resolving complicated issues, the data and resources needed to get the results are easily accessible, and a growing number of efficient algorithms are being implemented daily. These are just a few of the reasons why researchers used to be so keen on DL [13]. Deep Learning is one kind of machine learning. Additionally, there are two categories of machine learning algorithms:

supervised and unsupervised. Stochastic gradient descent (SGD), an optimization process, is an essential requirement for deep learning algorithms. Machine learning algorithms perform incredibly well in a variety of scenarios. Nevertheless, these algorithms haven't shown good performance in the main AI tasks, such as speech and object recognition. This issue catalyzes DL's development [14].

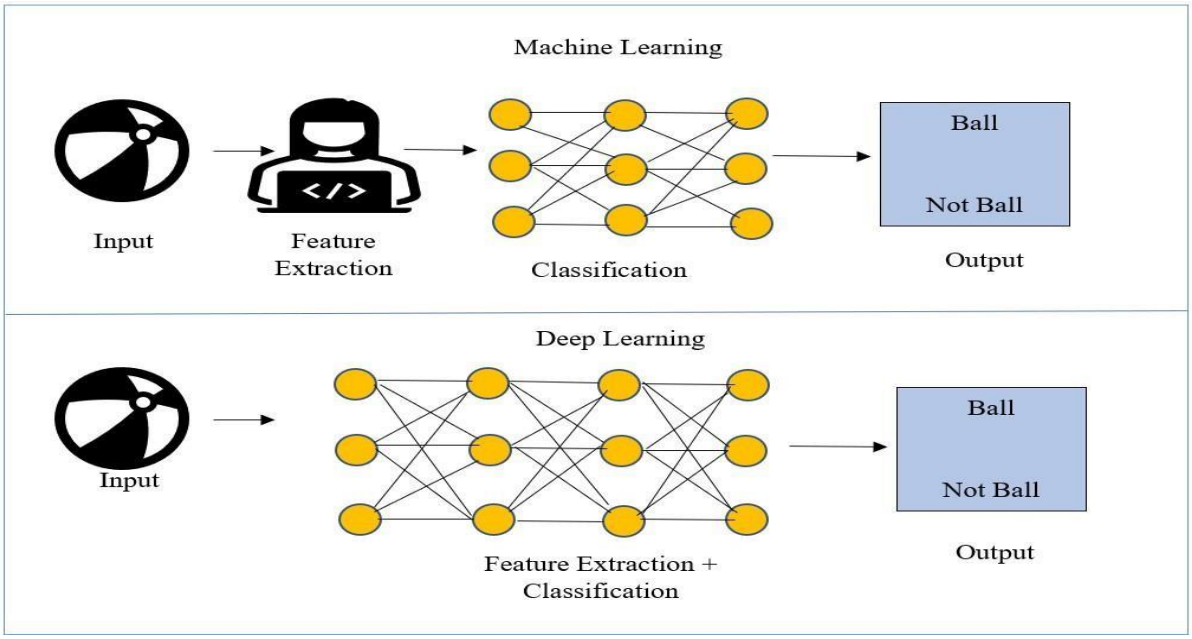

Fig 1.11 Machine Learning Vs Deep Learning

Machine Learning algorithm works well with small data, and it also requires manual feature extraction from the data as shown in Fig 1.11. On the other side, the Deep Learning model works for big data and does not require human interference for feature extraction. It automatically performs feature extraction and classification for the data. However, with machine learning, a model must be developed to extract features first, followed by another model for classification. One method of machine learning is called deep learning.It enables computers to learn by illustration, something that comes with such ease to us. Deep learning is now recognized as a key breakthrough in technology in self-driving vehicles, enabling them to recognize stop signs and distinguish pedestrians from light poles. It is the key to regulating sound on electronics for consumers, such as TVs, phones,

laptops, and hands-free speakers. With good reason, deep learning has been receiving a lot of attention lately. It is producing outcomes that were not previously achievable. A computer model learns to do tasks like categorization directly from images, text, or audio through deep learning. Deep learning approaches can attain exceptionally high levels of precision, frequently surpassing human performance. Models are trained using vast collections of labeled data and multi-layered neural network topologies.

Deep learning obtains identification accuracy at a higher level than earlier approaches. In addition to being essential for safety-sensitive systems like autonomous vehicles, it aids electronic devices in achieving customer standards. The latest developments in deep learning have improved to the point that deep learning can now perform tasks, including object classification in photos, better than humans. Deep learning has gained popularity recently, despite being first proposed in the 1980s, for two key reasons:

- The methods of deep learning demand a lot of labeled data. For instance, numerous photographs and many hours of video would be required to create driverless cars.
- Deep Learning calls for a massive processing capacity. Deep learning benefits substantially from the layered architecture found in high-performance Graphic Processing Units (GPUs). When combined with clusters and cloud computing, this architecture helps design teams reduce the training duration of a deep learning network from weeks to hours or less.

Although neural network architecture is utilized by most deep learning models, these models are also referred to as deep neural networks. The number of hidden layers in the model is represented using the term deep. While a deep neural network can have 150 layers, conventional neural networks only have two or three hidden layers. Deep neural networks and neural networks that automatically extract features from data without any human involvement have been trained on tremendous quantities of labeled data.

1.4.1 Applications of Deep Learning:

Deep learning has a wide range of uses in practically every industry sector, including medical equipment and automated driving. The following defines a few of the applications:

- Automated Driving

Automotive developers utilize deep learning to automatically identify artifacts such as stop signs and traffic signals. Additionally, it reduces the number of injuries by using deep learning to identify pedestrians.

- Aerospace and Defense

  Deep learning is utilized for both secure and dangerous site classification for forces, as well as the classification of objects that track regions of concern from satellite imagery.

- Medical Research

  Algorithms that utilize deep learning are being deployed by cancer researchers to detect cancerous cells. Teams from the University of California, Los Angeles (UCLA) have constructed one microscope. This microscope has exceptional accuracy when it comes to detecting cancer cells because it was trained using a high-dimensional dataset.

- Industrial Automation

  Applications of deep learning may also be found in industrial automation as well. It has been used to figure out the danger area for those operating large, heavy machinery.

- Electronics

  Our voice can be detected by a variety of electrical devices we use across the house. Deep learning is the bedrock upon which all those gadgets operate [15].

1.4.2 Neural Network classification

Neural networks are one type of machine learning approach. It is analogous to the human nervous system and brain. A neural network comprises various layers. They are the output layer, hidden layer, and input layer. Each layer has different nodes, and these nodes are connected to other nodes of that layer and those nodes are connected to the adjacent layers. There are so many areas where we can use neural networks such as pattern recognition, classification, clustering, computer vision, regression, and natural language processing. Frank Rosenblatt developed a prototype known as perceptron in 1957 for the Neural Network. It was initially having two layers. But it did not perform well. It was not capable

of learning even XOR (Exclusive-OR function). To perform any continuous operation, hidden layers are required that become the motivation to develop DNN (Deep Neural Network). Another factor behind the DNN was the development of a backpropagation algorithm. The automatic feature extraction quality of DNN makes it different from other learning algorithms.

A type of neural network known as DNN has MLP (multiple layers perceptron). DNN is trained with the help of algorithms without manual extractor designs so that it can learn representations from different datasets. The deep learning model contains multiple higher and deeper layers as compared to the shallow learning model. Shallow learning models have not performed well for the mapping of complex and non-linear functions. This contributes to the development of DNN. The emergence of GPGPU (General-Purpose Graphic Processing Unit) and big data also played a key role in the growth of DNN. Although CPUs are more efficient than GPUs, there are a greater number of parallel processing cores in GPGPU that make them good for DNN. There are different types of classifications of Neural Networks:

- Recurrent Neural Network
- Feed Forward Neural Network
- Kohonen Self-Organizing Neural Network
- Modular Neural Network
- Radial Basis Function Neural Network

1.4.2.1 Feed Forward Neural Network

Information can flow from the input layer to the output layer in a single direction—forward—when using a feed-forward neural network. No loopbacks and circles are created in such a network [16]. There is not any feedback from the output layer to the input layer in a feed-forward neural network. Feedforward neural networks come in two types: single-layer and multilayer. In a single layer, there is only an input and output layer. In a multilayer feed-forward neural network there exists a hidden layer between the two as shown in Fig 1.12 and 1.13 [57].

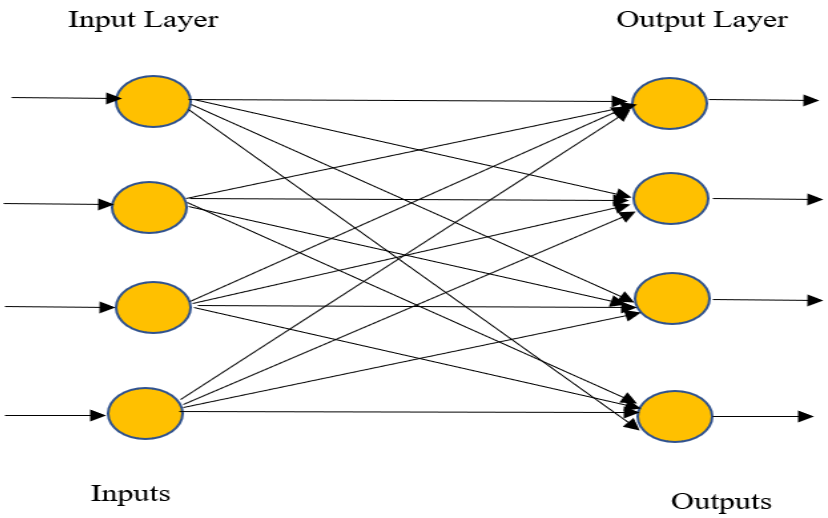

Fig 1.12 Single Layer Feed Forward Neural Network

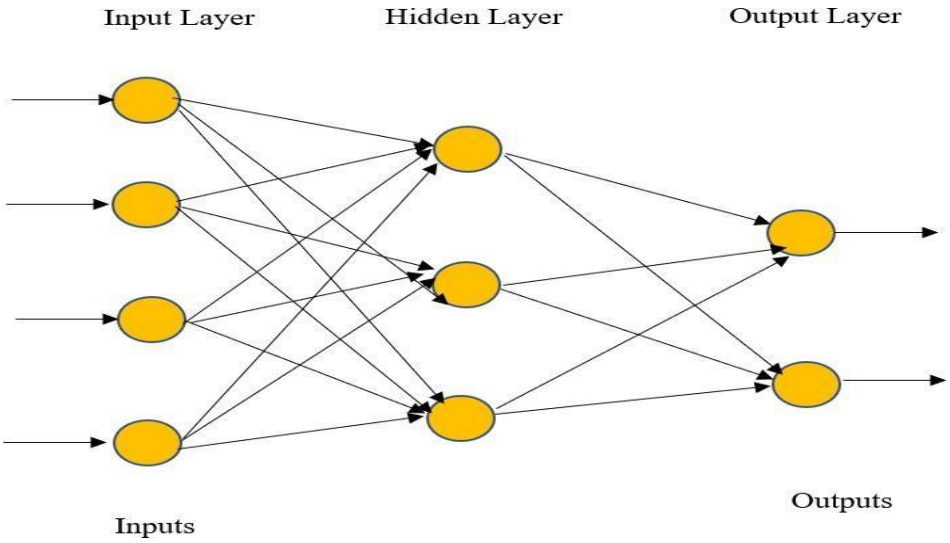

Fig 1.13 Multilayer Feed Forward Neural Network

1.4.2.2 Recurrent Neural Network

The processing unit forms a loop in RNN. The output of one-layer feeds back into

itself to create a feedback loop, and the output feeds into the next layer, the only layer in the network. This improves the network's ability to store knowledge about prior states and use that information to regulate the output. RNN can work on the input sequence and create the output sequence. This makes it ideal for various applications such as recognition of speech, and classification of videoframe by frame. For example, if an input consists of a three-word sentence sequence, then each word would pertain to a layer, and the network would then be unraveled and unzipped three- times into a three-layer RNN.

1.4.2.3 Radial Basis Function Neural Network

This type of neural network works in problems such as classification, function approximation, prediction problems based on time series and so many. It is made up of layers, containing input, output, and hidden or secret layers. In the secret or hidden layer, each node is represented as a cluster center along with a radial basis function that is defined as a Gaussian function. The network learns how to distribute the input to the center point by combining the weight parameters for classification and inference with the outputs of the radial basis function in the output layer.

1.4.2.4 Kohonen Self-Organizing Neural Network

This type of neural network uses unsupervised learning to arrange the network model into the input data. The input and output layers are the two fully connected layers that make up this structure. The output layer of the network is structured as a double-dimensional grid. It does not contain any function of activation, and the output layer attributes are represented using weights. In this case, the Euclidian distance between each output layer node and the input data is calculated using weights.

1.4.2.5 Modular Neural Network

In this type of neural network, a large network has been broken into smaller modules that are independent of each other. Each module does some tasks and

then at a later stage, all those outputs are merged to get a single output[16].

1.5 Thesis Organization

The thesis has been organized into various chapters. Chapter 1 discusses the introduction part. This chapter explains the topics that provide the basis for my topic of research. It explains artificial intelligence, its history and application, machine learning and its various types, deep learning and its applications, neural network and its various types.

Chapter 2 discusses the Literature review of all the research and review papers that have been studied. This review of the literature has been required to formulate the research gaps so that the necessary objectives of the research have been made. Several approaches for deep learning and machine learning have been explored and various reasons to work with deep learning models have been highlighted. By the end of this chapter, various research gaps have been shown to support the necessary research objectives.

Chapter 3 discusses the problem formulation that I have formulated so far andthe objectives of my work. There are four objectives of my research work around which my whole thesis revolves. A hybrid approach based on a deep learning modelfor feature extraction has been devised after a variety of machine learning and deep learning algorithms have been examined for the first objective. Subsequently, an additional hybrid approach to classification has been suggested. Lastly, acomparison has been made between various state-of-the-art methodologies and our proposed solution.

Chapter 4 discusses the proposed methodology to achieve my objectives of research which are framed in chapter 3. Detailed steps of methodology along with a proper flowchart have been given.

Chapter 5 gives a brief overview of the work done in the field of dimensionality reduction for gene expression datasets using different types of techniques. It also gives an outline of the proposed sandwich stacked ensemble model using VGG16 and VGG19 and compares it with various existing models using different visualizations.

Chapter 6 discusses various techniques that are based on RNN and CNN that are used for

classification. It also gives an overview of the hybrid method that has been designed for classification and comparison with various existing models in the form of various visualizations.

Chapter 7 marks the conclusion of the research work done and states the future scope as well.

CHAPTER 2

LITERATURE SURVEY

This chapter summarizes the work done in the field of tumor prediction for gene expression datasets using a variety of machine learning and deep learning algorithms. It gives an outline of the whole of the literature that has been done in support of the research work.

2.1 Introduction

Cancer is one of the major threats to mankind. The death rate is high in cancer as compared to other diseases. But if cancer has been detected on time, it can save the life ofthe patients. Although the early signs of cancer are not always clear, it is difficult to diagnose the disease. However, because it can identify cancer at an early stage, genetic cancer diagnosis has become more and more common in recent years. The literature coversseveral machine learning and deep learning approaches that are used to find cancer in the gene expression data for different cancer types, notably kidney, lung, and breast cancer. Some of those techniques are reviewed here in the next section.

2.2 Machine Learning and Deep Learning-based Classification Techniques for Gene Expression Data

In [17] A method that attempts to classify cancer based on gene expression has been stated by T. R. Golub et al., and DNA microarrays have also been used to monitor the proposed strategy. Human acute leukemias have been used to test the suggested strategy. The class discovery approach automatically distinguishes between AML (acute myeloid leukemia) and ALL (acute lymphoblastic leukemia), even in the absence of prior knowledge of such instances. It has been revealed that the method provided is viable for classifying cancer and that it offers a plan for identifying and predicting cancer classifications without prior expertise.

In [1] Donna K. Slonim et al., the work has been done to classify the samples of cancer patients as this is one of the most difficult steps. For classifying the samples, the author provided an approach that relies on computational analysis of gene expression data. Class prediction and class discovery are the two aspects that collectively make up the classification problem. Class discovery is the method of classifying samples into distinct classes with identical features and behaviors. Class prediction is the process of assigning new examples to the established classes. Classifying bone marrow and blood samples of 38 acute leukemia patients has been accomplished by applying the proposed cancer prediction method. Moreover, 34 samples have gone through testing using this approach. Furthermore, the author presents a method for using predictors to confirm the reliability of novel classes. The suggested approach opens the door to further research using molecular classification.

To classify tumors according to certain gene expression signatures, Javed Khan etal. [18] developed an Artificial Neural Network (ANN) based approach. Small, round blue cell tumors, or SRBCTs, have been utilized to train the suggested approach. The four categories of cancer are defined by the SRBCTs, and the diagnosis of these diseases mightresult in complications for the patient throughout treatments. But because of this novel approach, all the samples have been appropriately categorized and their genes have been discovered. The procedure uncovers the genes involved in the SRBCTs model. This proposed approach has been brought to the test using the new samples to assess the method's efficacy.

In [19] Michael D. Radmacher et al. proposed one of the frameworks depending upon gene expression profiles to predict the classes of predefined tumors. With the help of this framework, firstly the correctness of the class prediction has been evaluated for the current dataset. After this the prediction method has been selected and using the method cross-validation for the class prediction has been performed. The method of compound covariate prediction has been implemented. At last, the method's performance has been assessed through permutation testing. The proposed strategy has been used to classify 22 breast

cancers using gene expression in a real dataset. Although the method is working well, there are still certain issues with the predictor's classifications.

In [20] Scott L. Pomeroy et al. presented a solution for the problem of embryonal tumors of the central nervous system (CNS). There is one of the types of malignant brain tumor known as Medulloblastomas which has been addressed by the author. The author has presented a categorization approach based on DNA microarray gene expression data. This data has been taken from 99 samples. Using PCA, it has been determined that medulloblastomas differ molecularly from different kinds of cancers. The suggested strategy additionally confirms previously reported findings on cerebellar granule cell-derived medulloblastomas.

In [21] A PCA and FA-based approach has been proposed by Weng Shifeng et al. to handle one of the issues concerning supervised learning. The proposed method carries out Supervised Learning with two kinds of attributes namely nominal and interval-scaled attributes. The suggested strategy is suitable for supervised learning's gene expression analysis as well. The gene expression for lung adenocarcinomas has similarly been found to be dispersed in a high-dimensional space, with linear discrimination possible among the features.

In [22] A feature selection mechanism based on the backward elimination method has been proposed by Kai-Bo Duan et al. The Support Vector Machine-Recursive Feature Elimination (SVM-RFE) method employs a backward elimination procedure that is comparable. Nevertheless, the feature ranking score is calculated by the proposed approach using the statistical analysis of several linear SVM weight vectors that were generated using the initial training data subsamples. Evaluating it on four gene expression cancer datasets revealed that the suggested approach performs more effectively than the original SVM-RFE version. Furthermore, it has been demonstrated that a performance efficiency metric for gene expression-based cancer classification may be selected from a variety of test sets and training partitions.

With an increasing number of biological measures, the challenge of integrating and interpreting different kinds of genomic measurements has become essential. A technique that sets the patterns of variation in two biological inputs using Generalized Singular Value Decomposition (GSVD) was created by John A. Berger et al. in [23]. Analysis and iterative building of data on various angles identified by the GSVD projection angle have been done on the gene expression and copy number data. Implementing the approach in two genome-wide investigations of breast cancer has been proven to be effective. It discovers genes that exhibit notable variations in expression and copy number amongst various cell lines and tumor samples. The suggested strategy is also helpful for examinations that rely on expression and a huge amount of shared copy numbers.

Rui Xu et al. have suggested a method in [24] that analyses gene expression profiles to categorize tumor tissues into more than two groups using semi-supervised Ellipsoid ARTMAP and PSO. The method provides excellent results on three publically available data sets, specifically with the dataset NCI60. The method has been compared with the other four machine learning methods, but it performed well on three other data sets and the variation in classification accuracy is highly important. A few of the problems are linked to dimensionality and noise.

A prediction approach based on a radial basis function neural network and rough-based feature selection method was presented by Jung-Hsien Chiang et al. in [25]. It can find the specific features without prior information about the number of clusters and define centers approximating the correct ones. Naïve Bayes and Linear Support Vector Machinesare used as classifiers. This novel method has been validated on numerous data sets and is indicated to have a superior classification rate.

A method based on the perturbation methodology, the cluster ensemble approach, and the cluster validity index was presented by Zhiwen Yu et al. in [26] to ascertain the class from gene expression data. Four cancer datasets and three synthetic datasets have been employed to test the suggested technique. Disagreement and Agreement Index (DAI) has

shown that it is successful in recognizing the internal structure of most cancer datasets as well as all synthetic datasets. Additionally, when applied to gene expression data, DAI provides a greater validity index than other cutting-edge approaches.

The histopathological classification technique does not provide good results for predicting non-small cell lung cancer. The technique centered around genome-wide gene expression has been described by Jun Hou et al. in [27] and applied to the analysis of 91 instances. 91 tumors and 65 normal lung tissue adjacent to this have been used. A set of predictor genes is also defined using expression profiles. Tumor signatures of 5 genes aswell as histology signatures of 75 genes have been identified. The correlation analysis has also identified 17 genes that provide an association with the survival time after the surgery. The method has been tested against data from Duke University with 96 samples. 17signatures performed very well compared to the previous methods. It has also been found that identified gene signatures can act as a very good tool for histo-pathological classification of non-small cell lung cancer.

Ramon Salazar et al. have proposed a method in [28] which employs gene expression to predict cancer. The purpose of this work is to develop a reliable gene expression classifier that can anticipate an early disease relapse in patients with colorectal cancer (CRC). 188 patients with stage I to IV colorectal cancer who underwent surgery had their freshly frozen tumor specimens examined with Agilent 44K oligonucleotide arrays. A follow-up period of 65.1 months was typical, and 83.6 percent of the patients did not get chemotherapy. A prognostic classifier (ColoPrint) was built and an ideal collection of 18 genes has been discovered. To verify the genuineness of the signature, 206 patients with stage I, II, and III colorectal cancer were incorporated into the sample. Forty percent of the patients have been classified as high risk, and sixty percent as low risk. The coloPrint greatly improves the prediction accuracy of pathological parameters and MSI in patients with stage II and III CRC and enables the identification of those with stage II cancer who can be safely treated without chemotherapy.

A scientific approach for learning an interaction network between sparse DNA copy number areas and their associated transcription targets in breast cancer was suggested by

Yinyin Yuan et al. in [29]. Although it has the advantages of concurrent selection and feature efficiency, the sparse network's inference performance on simulated and real data isn't notably worse than other models. Distinguishing between expression changes that are copy-number-dependent and those that aren't is made simpler by the DNA-RNA interaction network. It's been demonstrated that the recommended approach produces a quantitative dependent score for copy counts that distinguishes between cis- and trans- effects.

Haseong Kim et al. presented a reverse engineering method in [30] that aims to bring together all admissible models representing gene interactions. This approach is based on the average of the Bayesian Model. This proposed method with a Gibbs distribution-dependent prior offers an efficient means of combining multiple biological data sources. The study indicates greater sensitivity in a simulation analysis with a maximum of 2000 genes than existing elastic-net and Gaussian graphic models with a specificity of 0.99. It has been found that the presented method outperformed the other methods on a DREAM dataset created by non-linearity stochastic methods. In brain tumor analysis, the regulatory genes found in the tumor networks may be the primary genes that trigger disease by regulating other genes, provided that most of the Gene Ontology (GO) words are strongly connected to the process of regulation and growth. Findings have additionally shown that the proposed strategy would be extremely beneficial, as it offered information that was unavailable from conventional DEGs (Differentially ExpressedGenes) approaches.

Ahmad B. Ashraf et al. have shown how to segregate breast cancers using multi-channel Markov Random Fields (MRF) [31]. Multichannel MRF (Area Under Curve: 0.97) performs better in segmentation than single-channel MRF (Area Under Curve: 0.89), as do other segmentation methods such as the structured segmentation cut approach. In particular, it is also distinguishable between the women with high and low breast cancer recurrence risk when tumors segmented by the proposed method have been fed to the SVM classifier.

Locality Sensitive Laplacian Score (LSLS), a technique that relies on supervised gene selection, has been proposed by Bo Liao et al. in [32]. The method integrates

discriminatory information into the local geometric structure by reducing local in-class information and simultaneously enhancing local in-class details. The LSLS and wrapper method are combined to create a two-stage feature selection method that is used to detect superior subsets of genes. Six datasets have been utilized for evaluating the presented strategy, and the results revealed that it outperforms other cutting-edge techniques.

A method based on Robust Principal Component Analysis (RPCA) has been presented by Jin-Xing Liu et al. in [33] to use gene expression data for categorizing tumor samples. It acts as a model for the gene's characteristic of a specific biological cycle. Then, Linear Discriminant Analysis (LDA) in conjunction with RPCA and RPCA is used to identify the features. Finally, the gene expression data for the tumor samples can be classified using Support Vector Machines (SVM) based on pertinent criteria. Seven datasets have been utilized to evaluate the method, and the results indicated that it is both practical and appropriate for classifying tumors.

Yifei Chen et al. introduced D-GEX, a deep learning method, in [34] to extract target gene expression from landmark gene expression. GEO dataset based on microarray has been used for the training purpose. The suggested method outperforms the linear regression strategy with a comparative improvement of 15.33 percent and lowered error in 99.97 percent of the target genes. This method's efficacy was also assessed using the RNA-Seq-Based GTEx dataset, where it outperformed the linear regression method overall with a relative improvement of 6.57 percent and reduced error in 81.31 percent of the target genes.

In [35], Su-Ping Deng et al. proposed a fused network to identify the stages of kidney renal cell carcinoma (KIRC) using DNA methylation and gene expression data. The potential of the fused network has been tested with data from two networks. It has been found that the proposed method has performed well with data of two types as comparedto data of two patient networks from one type of data. Moreover, using network-based features from different types of data has been shown to improve disease diagnosis.

Lei Chen et al. in [36] identified the genes that are expressed in 32 normal tissues/cancers

(i.e., widely expressed genes; FPKMN1 in all samples) and those that are not identified (i.e., rarely expressed genes; FPKMb1 in all samples) based on the wide gene expression data to investigate the functional differences between rarely expressed and widely expressed genes. To differentiate between the genes, the supervised classifier RNN and the incremental feature selection method have been employed. The proposed approach helps to map the landscape of gene expression and demonstrates how gene expression modifies tissues and the cancer microenvironment.

Murtada K. Elbashir et al. suggested a lightweight CNN breast cancer classification system in [37] based on RNA-Seq gene expression data. Targeting the high dimensionality and small amount of gene expression data, the method is strictly limited to two convolutional layers. Grid search along with cross-validation are used in the process of choosing CNN's hyper-parameters. It has been observed that the presented approach performs more accurately (98.76 percent) than other existing approaches.

Nour Eldeen M. Khalifa et al. presented a deep learning-based method in [38] that uses gene expression data for the tumor RNA sequence (RNA-Seq) to distinguish between different forms of cancer. The proposed method is based on CNN and binary particle swarm optimization with decision trees (BPSO-DT). This method has three phases namely preprocessing, augmentation, and deep CNN architecture. The proposed approach outperforms the other identical approaches and acquires a testing accuracy of value 96.90 percent. It has also been observed that the method requires less time and is less complicated.

In [45], Chun-Hou Zheng et al. described a method for categorizing cancers based on gene expression data. The method relies on nonnegative matrix factorization (NMF), or sparse NMF, to choose genes. The method has been tested on three datasets, and the findings show that it is a very effective and efficient way to discriminate between normal and malignant tissues. The selected genes are also studied for their biological terms. The dimensionality reduction of the gene expression dataset has also been studied.

To bicluster the tumor gene expression data, Xuesong Wang et al. stated a method in [47]

which is based on Dual Hypergraph Regularized PCA (DHPCA). Two hypergraphs have been constructed namely sample and gene hypergraph. Four real-world applications have been utilized to evaluate the suggested approach, and the results indicated that it is a magnificent tool for biclustering. Further, it has been discovered that this approach finds gene clusters with similar biological roles. The method's incomplete evaluation is one of its shortcomings. More evaluation is required for gene clustering.

The LGEPM model was created by Huiqing Wang et al. in [48] and is used to extract the non-linear features that influence gene expression. Neural networks having long-short-term memory act as the model's backbone. Target gene prediction depends on the features that have been extracted. It has been observed that this new method successfully fixed the issue that the LINCS (Library of Integrated Network-Based Cellular Signatures) program was facing. As a work of the future, this LGEPM can be combined with the latest models of prediction to get more accuracy.

Tarek Khorshed et al. presented the GeneeXpression Network (GeneXNet), a multi-layer CNN-based framework, in [49]. It serves to classify cancer's several tissues. The specified model's visualization has been produced to promote the application of deep learning to biological applications. As a work of the future, ensemble models can also be designed across Omics data to increase the performance of the classifiers. The method has been tested against 33 different types of cancer and it has been found that the method achieves classification accuracy of 98.9 percent.

A Partial Least Squares (PLS)-based gene-selection procedure has been presented by Guoli Ji et al. in [50] to identify the genes from a high-dimensional small sample to eradicate cancer promptly. Three parameters namely VIP, VEG, and IEG are defined and are used to find the effects of combined genes and their correlation. Two datasets have been used to evaluate the technique. This method has proven to be reliable and effective. It delivers superior classification outcomes in multi-category datasets as well.

Fang-Han Hsu et al. proposed a single transcription model (ST) in [51] that relies on the Laplace-Stieltjes transform and numerical analysis. Transcribing factors (TFs) can be

uploaded using this manner following transcription specifications. The functional problems caused by copy number changes (CNAs) are evaluated using mathematical models and simulations. However, the latter was not feasible with the Unlimited Transcriptions (UT) method.

Muhamed Wael Farouq et al. used a fusion-based method [52] to profile gene expression in non-small cell lung cancer. The primary contribution of the suggested method is the merging of data from different samples in the dataset under examination using evidence-based Demspter theory-based data fusion. As an outcome, the definitionof genes linked to non-small cell lung cancer has been enhanced by decreasing uncertainty and boosting decision validity and reliability.

Serkan Kiranyaz et al. released research in [53] on 1D convolutional neural networks, their applications, and a comparison with 2D convolutional neural networks. Whenever low-cost implementation and a minimal amount of training data are needed, 1D CNN performs well. In such cases, there is little training data, which prevents 2D CNN from working. It has also been observed since 1D CNN operates on limited data, it is less complex and requires less labor during training. Only low-processing systems, which include mobile and handheld ones, can use 1D CNN. However, if an enormous dataset is provided and intricate computations must be performed, 2D CNN must be utilized. In addition, it has been discovered that 1D CNN applications deploy fewer hidden layers with fewer parameters (less than 10 K). However, more layers with parameters larger than 10 M are used in 2D CNN. This demonstrates that 2D CNN is preferable to 1D CNN for large datasets as it has more parameters and can extract more features.

Yunan Wu et al. presented a method in [54] for identifying images of ECG signals as normal or abnormal based on a 2D convolutional neural network. The accuracy of the proposed approach is 98%. Furthermore, a study shows that 2D CNN performs far better than 1D CNN. In comparison with 1D CNN, the 2D CNN approach is more reliable and accurate. The input parameters in 1D CNN are less than 2D CNN which makes CNN work fine for the limited data. On the other hand, as number of input parameters in 2D CNN is more which is why it creates the problem of overfitting when it applies to small or limited

data, and it also affects the accuracy of the model.

The CNN architecture's most vital component is the convolution layer. The first hidden layer prioritizes low-level features, the second hidden layer focuses mostly on high-level features, and so on. CNN performs extremely well with the images due to its hierarchical structure. In CNN, the subsequent layers are not entirely connected, and the weights are repeatedly used. It has fewer parameters than networks with all connections. CNN has certain advantages over other fully connected networks, particularly a reduction in training time, training data requirements, and overfitting. Once CNN's kernel has learned a feature, it can recognize it in many places. Images are made up of different iterative elements. As a result, CNN can extract high-dimensional features from images and works well with images that have less training data. In 1D CNN, large-size filters are used. It signifies the fact that there would only be 7 feature vectors if a filter of size 7 is used. However, if a size 7 filter is applied to a 2D CNN, the result would be 49 feature vectors. Because of this, 2D CNN offers higher dimensional features than 1D CNN [55].

Several centrality and machine-learning methods have been used to determine which proteins are required for a cell's survival and evolution. However, a limitation of these approaches is that they necessitate an extensive amount of prior knowledge. To overcome this problem, Min Zeng et al., proposed a deep learning method based on node2vecwhich automatically identifies the essential proteins [63]. In females, breast cancer is very common and a major threat to their health. Various deep-learning models of CNN are used for detecting breast cancer from pathological images. Min Liu et al. also proposed a method known as AlexNet-BC for detecting breast cancer. Data augmentation has been performed on BreaKHis dataset to avoid the overfitting problem [64].

Subtypes of cancer have been classified using deep flexible neural forest (DFNForest). Along with this, dimensionality reduction has been done using a proposed method based on the amalgamation of neighborhood rough set and fisher ratio. DFNForest provides better accuracy for the classification of cancer subtypes than other methods [65]. There is another deep learning-based approach namely CapsNetMMD to discover genes for breastcancer. This proposed method uses Capsule Network Based Modelling of Multi-Omics Data [66].

Table 2.1 Various ML and DL Techniques of Classification for Gene Expression Data

Sr. No.	Paper Name	Author	Year	Objective	Techniques/ Tools	Dataset	Findings
1	[17]	T.R. Golub et al.	1999	To develop a method for classifying and predicting cancer classes	Self-organizing maps, DNA microarrays, and neighborhood analysis	Eleven adult AML samples and 27 ALL samples from the Dana-Farber Cancer Institute and Leukemia Group B (CALGB leukemia cell bank	A feasible approach. Experimentation necessitates careful handling.
2	[1]	Donna K. Slonium et al.	2000	To categorize cancer samples based on gene expressiondata	Neighborhood analysis, Genecluster software, Affymetrix oligonucleotide arrays, and computational analysis	38 samples of leukemia (11 AML, 27 ALL), 34 samples for testing (14 AML, 20 ALL)	The results are better for genes without correlation, with a median prediction strength of 0.86.

3	[18]	Javed Khan etal.	2001	To analyze their gene expression to determine different cancer groups	DeArray Software, cDNA microarrays, and ANNs	National Institutes of Health, NCI, MSKCC, CHTN, DZNSG, ATCC	It is also usable for non-linear characteristics. It is powerful. High sensitivity and specificity are also attained.
4	[19]	Michael D. Radmacher et al.	2002	To create a framework for predicting predetermined classes of tumor	BRB ArrayTools, Compound Covariate Prediction	22 patients' hereditary breast cancer dataset[39]	Excellent setup to compare prediction methods. Require some improvements
5	[20]	Scott L. Pomeroy	2002	To introduce a classifying approach for DNA microarray gene expression data	SOMs, PCA, KNN, Cluster and TreeView Software	Several datasets, particularly one with 99 samples and other with 42 samples, have been used.	Gene expressions offer an advantageous technique for medulloblastoma diagnosis.

6	[21]	Weng Shifen get al.	2004	To demonstrate a method for categorizing data according to interval-scaled characteristics	PCA, FA, and Fuzzy FA	A subset of the full dataset used in [40] consisting of 203 samples	Efficiently employed in supervised learning. Compared to surgical-pathological staging, FA delivers greater information
7	[22]	Kai-Bo Duan etal.	2005	To provide a strategy for selecting gene features	Multiple SVM-RFE	The Kent Ridge Bio-Medical Data Set Repository offers four gene expression datasets.	MSVM-RFE has better classification accuracy than SVM-RFE. The performance of SVM has gotten better.
8	[23]	John A. Berger et al.	2006	To provide an infrastructure for addressing the issue of	Singular Value Decomposition in Generalized Form	American Type Culture Collection's Fourteen Breast Cancer Cell Lines	Useful for analyzing both gene expressions and copy number data. Other data forms can also be used with

				integrating various data kinds			greater efficiency.
9	[24]	Rui Xu et al.	2007	To present an approach for identifying tumor tissues with diverse gene expression profiles	PSO and ssEAM	NC160, ALL Dataset, Acute Leukemia	PNN, ANN, LVQ1, and KNN are beaten by ssEAM atthe 0.05 significance level.
10	[25]	Jung-Hsien Chiang	2008	To provide a gene expression data analysis selection method	Rough Based Feature Selection Method, RBF Neural Network, Naïve Bayes, and Linear SVM	Datasets for Lung Cancer, Prostate Cancer, ALL, and AML available at http://sdmc.lit.or	99.8% is the best classification accuracy rate.

						g.sg/GED atasets /Datasets.	
11	[26]	Zhiwe n Yu	2009	To provide a framewor k for class discovery of cancer from gene expressio n	Cluster ensemble, permutation approach, and cluster validity index (DAI)	3 synthetic, 4 real datasets (Novartis multi-tissue [41], Leukemia [17], Lung Cancer [40], St. Jude [42])	While finding the correct number of classes, DAI works better than other current techniques.
12	[27]	Jun Hou	2010	To develop a gene expressio n-based method for NSCLC classificat ion	SpotFire Decision Site, Hierarchic al Clustering, and Proportion alHazards Model	Six normal lung tissues and 91 NSCLC were obtained from GSE3526 (Duke University).	The most efficient method for histopathologic al classification involves gene signatures.
13	[28]	Ramon Salazar et al.	2011	To propose a classifier that can	Unsupervis ed Hierarchic al	206 testing samples (Institute Catalad'Onc	In the test dataset, 86% of the patients are classified as

				be used to predict disease in CRC patients	Clustering, Kaplan-Meier Method, and Agilent 44K Oligonucleo tide Arrays	ologia, Spain) and 188 training samples (NCI, LUMC, SGH)	low-risk. The first CRC prognostic method
14	[29]	Yinyin Yuan etal.	2011	To present a method that incorporates expression and copy number information across the genome	Local and global search strategies, L1 and L2 constrained regression	89 samples from the California Pacific Medical Center and UG San Francisco breast cancer dataset [43]	beats other current approaches in terms of accuracy
15	[30]	Haseong Kim	2012	To suggest a model that integrates many	Bayesian Model, ANOVA Test, Gibbs Distribution, and	GSE4290, Dataset DREAM	A 0.99 specificity has been obtained. Superior to those of Enet and VAR

				models to explain gene interactions	GPU/CPU Parallel Programming		
16	[31]	Ahmed B. Ashraf	2013	To present an expanded structure for breast tumor segmentation	Gaussian Mixture Model, kinetic observation model, and multichannel MRFs	DCE breast cancer MRI images	Using multichannel MRF, an AOC of 0.9 has been achieved as opposed to 0.89 with single-channel MRF. Applying SVM to segmentation yields better results.
17	[32]	Bo Liao	2014	To provide a method for gene selection	SVM, LSLS, and Wrapper Method	6 datasets are available at the Kent Ridge Bio-Medical Data Repository	In terms of accuracy, precision, recall, F-score, and AUC, LSLS outperforms KW and SPFS.
18	[33]	Jin-Xing Liu	2015	To present a novel method	LDA, RPCA, and SVM	Nine distinct datasets that are publicly accessible	Performance has been assessed using

				that is used to classify tumor samples from gene expression data		(data on acute leukemia [17], data on colon cancer, data on glimos, data on medulloblastoma, data on prostate cancer, 11_Tumor Data, and data on brain tumors)	accuracy, AUC, and LOO-CV. A practical and efficient technique
19	[34]	Yifei Chen etal.	2016	To propose an approach based on deep learning to ascertain target gene expression	D-GEX	RNA-Seq-Based GTEx Dataset and Microarray GEO Dataset	beats both KNN and linear regression (15.33 relative improvement). reduced error rate (81.31%) in most of the genes

20	[35]	Su-Ping Deng	2017	To develop a fused network for identifying stages of KIRC	Data on DNA methylation and gene expression, Sparse Partial Least Square Regression, SNF, SNFTool, and LASSO Label Prediction Method	The KIRC data for the cancer genome atlas (TCGA Data Portal)	The greater prediction accuracy compared to WDC, MLW, and KNN. It is sturdy.
21	[36]	Lei Chen	2018	To categorize both frequently and infrequently expressed genes	mRMR, RNN, and the incremental feature selection method	Gene Expression Dataset which is available at The Human Protein Atlas [44]	The functional level makes use of KEGG and GO terminology. Youden's indices for normal and cancerous tissues are, respectively, 0.739 and 0.639.
22	[37]	Murtada K.	2019	To introduce	CNN, Array-Array	Breast Cancer data	Achieves 98.76%

		Elbashir et al.		a simplified CNN for breast cancer classification	Intensity Correlation, R-Studio, Batch Normalization	from the Pan-Cancer Atlas	accuracy. Performance has been evaluated based on sensitivity, accuracy, precision, specificity, and F-measure
23	[38]	Nour Eldeen M. Khalifa et al.	2020	To present a deep learning-based approach for classifying various cancer kinds	CNN, Deep Learning, BPSO-DT	Cancer Types: Mendeley datasets provide RNA sequencing values from tumor samples and tissues.	It attains a 96.90% accuracy rate. The assessment criteria include F1 score, recall, and precision.
24	[45]	Chun-Hou Zheng	2011	To present a tumor classification method based on NMF	NMF, SVM, and SNMF	Acute leukemia dataset, medulloblastoma dataset, and colon cancer dataset [46]	It operates well and efficiently. There is little impact from sparseness.

25	[47]	Xuesong Wang	2018	To propose a model for biclustering of tumor gene expression data	PCA, DHPCA, and GLPCA	Colon Cancer, Medulloblastoma, SRBCT, 11_Tumors	Comparable include PCA, GLPCA, GNMF, ONMTF, and NMTFCoS. It gives better accuracy than other models
26	[48]	Huiqing Wang	2019	To build a model which employs non-linear information to predict gene expression	Algorithm for Unsupervised Clustering, L-GEPM, and Neural Network LSTM	GEO Data from GTEx, LINCS cloud, and 1000G RNASequence Data	outperforms LR-L1, KNN-R, and D-GM. Extracted target genes are significantly closer to the expression of the gene. NL features that are better and more flexible.
27	[49]	Tarek Khorshed et al.	2020	To present a multi-tiered model for classifying multiple cancer tissues	Back-propagation, RNA sequencing, CNN, supervised learning, and stochastic gradient	11093 samples from TCGA	AUC of 0.99 and total accuracy of 98.93 percent have been obtained.

					descent optimization		
28	[50]	Guoli Jiet al	2011	To suggest a method of gene selection capable of categorizing tissues in datasets with multiple categories	PLS, MATLAB, OSU_SVM 3.00 Toolbox Linear SVC, SVM, and Linear Support Vector Classifier	SRBCT datasets, MIT AML, and ALL datasets	It is reliable and strong. It performs brilliantly on datasets with two or more categories.
29	[51]	Fang-Han Hsu	2012	To provide an ST model for determining the impact of CNAs	Circular binary segmentation, Laplace Stieltjes transform and numerical analysis, Transcriptional Oscillation, Transcription	GEO/NCBI database	It illustrates the practical application of mathematical theories to analyze result and improves awareness of cancer biology.

					al Bursting, and Dynamic Modeling		
30	[52]	Muhammed Wael Farouq	2019	To demonstrate a multifusion-based method for profiling gene expression after non-thermal plasma therapy.	Fuzzy C-Means Clustering Method, Dempster-Shafer Method, and Matlab R2016b	GEO's NCBI Gene Expression Omnibus (GSE59997)	Boosts reliability and decreases uncertainty. Using C-means, one can determine the way various non-thermal plasma treatments affect gene expression.
31	[53]	Serkan Kiranyaz et al.	2020	To provide an overview of 1D CNN and its uses	NA	NA	1D CNN performs well in situations requiring fewer computations and tiny amounts of data. When low-cost implementation is necessary, it

							also functions well.
32	[54]	Yunan Wu et al.	2018	To suggest an ECG signal image classification technique based on 2D CNN	The NVIDIA GTX1080 GPU, CNN, and Intel 17-5930K CPU	MIT-BIH Arrhythmia Database	1D CNN performs inferior to 2D CNN. 2D CNN has higher robustness and accuracy. With little data, 1D CNN performs well.
33	[63]	Min Zeng etal.	2021	To create a deep learning framework to determine which proteins are necessary	Node2vec, LSTM, ReLU	PPI Network from BioGRID Database, GSE3431, Subcellular localization dataset of yeast	The proposed method performs better than existing Centrality methods and ML methods.
34	[64]	Min Liu et al.	2022	To detect breast cancer using a deep learning approach from	Data Augmentation, AlexNet, Computer with an Intel i7- 9700 3.0 GHz CPU and an	BreaKHis dataset, Invasive Ductal Carcinoma (IDC) dataset, UCSB Bio-	The proposed method gives high accuracy on pathological data. Dividing lesion area is a future scope.

				pathological images	NVIDIA Quadro RTX 4000 GPU	Segmentation Benchmark dataset	
35	[65]	Jing Xu et al.	2019	To classify subtypes of cancer using DFNForest	Fisher Ratio, Neighborhood Rough Set, Deep Flexible Neural Network Forest	Lung from TCGA, GBM (glioblastoma multiforme), and BRCA (breast invasive carcinoma)	Both the proposed methods provide better performance than existing feature selection and classification methods
36	[66]	Chen Peng et al.	2020	To discover cancer-related genes for breast cancer	Capsule Network Based Modelling	Multi-Omics Data for Breast Cancer	Performs better than ML models. It can't be used for diseases with unknown genes

2.3 Dimensionality Reduction for Gene Expression

To detect cancer several tests can be performed in the lab. It requires a lot of money, time, and resources to predict cancer in a laboratory. Computational methods are becoming complementary techniques to assess human physicians in the diagnoses of many diseases. Cancer can be diagnosed at the genome level, which allows for early detection of the patient's illness and treatment. In the case of gene expression, there usually exist few samples but many dimensions. [67], [68]. There are thousands of genes, where every gene is a feature in gene expression data. Some genes are more responsible for cancer than

others, which is why feature selection (gene selection) is required. Feature selection has proved its worth in the classification problems particularly when the dimensionality is high. This process is known as dimensionality reduction [69], [70].

In [69], the author proposed a method for the classification of cancer using gene expression data which is based on the Probabilistic Neural Network (PNN) and neighborhood rough set method which reduces the dimension of the genes. Ensemble PNN provides more accurate classification results than single PNN. Genes are selected using the Simba algorithm based on their ranking. The method based on a genetic algorithm and neighborhood rough set has been used in [70] to select the prominent features with smaller redundancy and higher discrimination from gene expression data. This method is efficient and provides higher accuracy for classification.

The inclusion of several classifiers and the diversity of base classifiers are the two key challenges when handling ensemble models. Consequently, a decision group employing K Nearest Neighbor (KNN), Naïve Bayes, and Decision Tree has been created toimprove the variety of the basic classifiers. The use of Genetic Algorithm (GA) and AdaBoost-based ensemble technique has been implemented to increase the accuracy of cancer classification by gene expression [71]. To classify the subtypes of cancer Deep Flexible Neural Forest was used in another work [65]. They have utilized a fisher ratio in combination with the neighborhood rough set to reduce the dimensions in gene expression data to boost classification accuracy [65].

To get the features from the gene expression data, LDA along with a deep learning-based autoencoder neural network was implemented [72]. They have achieved an accuracy of 98.27% [72]. Deep learning techniques have been used during classification to create an ensemble model for classification. To prevent cancer as well as for its treatment, its timely detection and diagnosis is of much importance. Various methods are available for this purpose, but these suffer from a problem of low diagnostic ability and bad generalization. Several goals have been examined with the hybrid method based onmulti-objective Particle Swarm Optimization (MOPSOHA), like determining the number of features, increasing accuracy, and computing entropy-based metrics. They have achieved cutting-edge

outcomes for the classification of cancer [73].

The outcomes of a study presented in [74] suggest that treating lung adenocarcinoma (LUAD) is hard. To successfully predict lung adenocarcinoma, the authors have used a number of machine learning methods, including ttest, K-Mean and hierarchical clustering, sensitivity analysis, and Self-Organizing Map (SOM) neural network. The data's dimension has been reduced using K-Means and T-Test. Sub-types of LUAD have been found using SOM and hierarchical clustering.

To improve the quality of prediction and to decrease the time of computation, dimensionality reduction is done while working with gene expression data. In [75], two dimensionality reduction techniques namely Attribute Selection (AS) and PCA are used during the pre-processing step. Then for classification, consistency-based subsetevaluation (CSE) and minimum redundancy maximum relevance (mRMR), known as CSE-mRMR have been used. Gene expression data has high dimensionality posing difficulties for the classical machine learning models. To cope with the curse of dimensionality, Attribute Selection, and dimensionality reduction techniques such ascorrelation analysis and PCA have been widely known. PCA came out to be the best unsupervised learning-based reduction method [75].

Partial Least Square (PLS) is a supervised learning-based dimensionality reduction technique used in literature. Gene expression data suffers from two main problems, high dimensions, and small samples. Thus, dimensionality reduction is the solution to this problem. There are two main methods namely PCA and Between Group Analysis (BGA) that are used for dimensionality reduction. The authors in [76] have proposed an automatic and non-parametric method known as Uncorrelated Discriminant Analysis (UDA) to extract features gene expression data. The proposed method UDA is based on maximum margin criteria (MMC). This method acts as a good solution to the problem of a small sample size.

A feature extraction method autoencoder followed by a dimensionality reduction method PCA has been implemented in [77]. For the classification of breast cancer, they have used

an ensemble classification technique named AdaBoost. PCA along with an autoencoder neural network has been used as a feature extraction method. These two are combined to add deep learning advantages to the feature extractor for extracting representative features from gene expression data. One problem with the proposed method is that it can be tough to identify the most salient features that offer support for the predictions.

Gene expression datasets are usually sparse and imbalanced. A new method based on the "Fuzzy Granular Support Vector Machine and Recursive Feature Elimination Algorithm" has been developed [78] to select the most prominent features from the data. This method is a combination of multiple methods namely statistical learning, fuzzy clustering, and granular computing, and provides 100% accuracy on eight genes as compared to previous methods that provided an accuracy of 86% with sixteen genes [78].

For cancer subtyping, conventional approaches use statistical techniques and also there occurs a problem of overfitting due to the small size of the sample. A Deep autoencoder has been used to extract features from gene expression data of hematopoietic cancer. This method outshines PCA and Non-Matrix Factorization. The author also states that there is no other research work that is related to hematopoietic cancer using deep learning techniques of feature extraction. In comparison to prior techniques, the suggested method performs better in terms of F1-measure, G-Mean, accuracy, precision, recall, and specificity [79].

The T-Test method and Kernel Partial Least Squares (KPLS) are also used to extract features from gene expression. However individually, these two approaches do not provide promising outcomes. When used in pairs, these methods act as a reliable method for feature extraction.In [80], the author used a t-test method for pre-processing in which irrelevant and noisy genes are filtered out, and then features are extracted using KPLS and finally classifier has been applied for the final classification.

Due to the development in DNA Microarray technology, it is becoming possible to study genes so that relevant information can be extracted. Relevant features have also been retrieved from the data using the deep learning-based GSEnet model. The model is

composed of “pre-conv, SE-Resnet, and SE-conv” modules. The proposed feature extractor has been tested with nine different classifiers namely DT, Naïve Bayes (NB), KNN, RF, AdaBoost (ADA), and Gradient Boosting Decision Tree (GBDT). It provides good performance [81].

With the help of deep learning techniques, it is possible to work with high dimensional genomic data as these techniques help to extract the most prominent features that are required to classify various diseases. AFExNet is one of those methods that has been developed as a dual-stage method for extraction of the features from breast cancer data. It is known as dual-stage as it consists of unsupervised pre-training and supervised fine-tuning. The suggested approach uses an Adversarial Autoencoder (AAE)-based neural network. Twelve distinct classifiers were used to test this strategy, and the results are encouraging [82].

In [83], the author proposed a stacked ensemble method to classify breast cancer from gene expression data. Various deep learning prediction models are used in a framework of stacked ensemble. The proposed stacked ensemble model is a two-stage model in which CNN has been used in the first stage for feature extraction and then extracted features from the first stage are passed as an input to the ensemble models used in the second phase. RF is used as a final classifier in the second phase to classify breast cancer.

Lung and colon cancer are two of the primary causes of death in humans. In [146], the author proposed a hybrid feature extractor based on transfer learning models such as VGG16, VGG19, MobileNet, DenseNet169, and DenseNet201. The extracted features are passed to the various ML models such as RF, SVM, XGB, light gradient boosting (LGB), logistic regression (LR), and multi-layer perceptron (MLP) for the classification of lung and colon cancer. The top three ML models have been filtered out using high performance filtering method and an ensemble classifier has been made for the final classification and evaluated based on various performance measures.

Gene expression data consists of non-linear relationships among the genes. PCA is used for linear dimensional reduction as it uses Euclidean distance, which in turn cannot find

the non-linear interactions among genes. In [147], PCA and isometric mapping (ISOMAP) are compared, and it has been concluded that non-linear dimension reduction (NDR) methods such as ISOMAP provide promising results. NDR methods have been applied to five different datasets as a pre-processing step. After that clustering methods namely K-means and Hierarchical clustering have been applied to the selected features.

Single-cell RNA-sequencing data suffers from a problem of noise and data sparsity. In [148], authors coined a solution to this problem by proposing scDMAE, which is a denoising model based on autoencoders. The topological and low-dimensional features are extracted by using two autoencoders. Later, these features are fused to make new features. Noise and missing values are handled by the masking strategy which has been applied in the autoencoder to make the gene data free from irrelevant and missing values. Refined genes are combined with original genes to create final gene expression profiles free from noise.

Enhancer promoter interactions (EPIs) play a significant role in gene expression regulation and prediction of genetic-related diseases. To handle the predictions of EPIs well, the author proposed a new method known as EPIMR [149]. DNA sequences are converted into images using the Hilbert curve. Multi-scale ResNet deep learning model has been used to extract features at different levels of abstraction. The extracted features are connected using a heuristic matching technique which is required to find the interactions between enhancer and promoter regions.

In [150], the author proposed enhanced VGG-19 (E-VGG19) for extracting features from the data, which are then passed to the machine learning models for the classification of skin cancer. Pre-trained deep learning models provide automatic feature extraction which improves the accuracy of the classification. ML classifiers have been compared with pre-trained models in terms of accuracy, recall, precision, sensitivity, and F1 score which proves that pre-trained models are better than existing ML models. E-VGG19 has also been compared with basic VGG19 and provided better results.

In [151], the author compared three DL models namely VGG16, LGBM, and Inception V3

for feature extraction from the CT images and three ML models namely SVM, RF, and XGB for the classification of pancreatic ductal adenocarcinoma. VGG16 provided the best feature extraction results and XGB provided the best classification results. The extracted features from VGG16 have been passed to the XGB for final classification, making it a hybrid of VGG16 and XGB. The proposed model provides an accuracy of 0.97.

Table 2.2 Dimensionality Reduction for Gene Expression

Sr. No.	Paper Name	Author	Year	Objective	Technique/ Tool	Dataset	Findings
1	[67]	Chaowen Zhong et al.	2019	To perform fault diagnosis using limited faulty training data	Generative Adversarial Network	NA	It achieves a classification accuracy of 90.44%. The method is unsupervised but still it requires supervised learning for classification.
2	[69]	Shu- Lin Wanget al.	2009	To develop a method for cancer classification based on gene reduction	Probabilistic Neural Network, Neighbourhood Rough Set Method, Simba Algorithm	Leukemia Dataset [17], Colon Tumor Dataset [46] and Small Round Blue Cell Tumor Dataset [18]	The method is stable and accurate. Genes selected from this method are the most important that improve the accuracy.

3	[70]	Hongbin Donget al.	2018	To propose a feature selection method	Improved Binary Genetic Algorithm with Feature granulation n (IBGAFG), Improved Neighbour hood Rough Set with Sample Granulatio n (INRSG)	http://csse.s zu. edu.cn/staff/ zhuzx/Datas ets.	The method is efficient and provides higher accuracy for classification by using granular information
4	[71]	Huijuan Lu etal.	2021	To classify cancer using the hybrid ensemble method	AdaBoost , GA, KNN, Naïve Bayes, Decision Tree	Breast, Lung, Colon, Leukemia, and Brain Datasets from UCI	The proposed ensemble method outperforms existing ensemble methods on small samples
5	[72]	Xinfen g Zhang et al.	2020	To classify different features from gene	LDA, Auto-Encoder (AE)	GEO Database	Good predictions with an accuracy of 98.27%

				expression data	Neural Network		
6	[73]	Yunhe Wang et al.	2021	to identify subtypes of cancer using gene expression	Multiobjective PSO	35 Cancer Gene Expression Datasets	The method has a high power of diagnosing subtypes of cancer
7	[74]	Fuyan Hu et al.	2020	To propose a method for the classification of subtypes of LUAD Cancer	hierarchical clustering, SOM neural network, t-test, sensitivity analysis, and k-means clustering	LUAD Datasets from TCGA	Extracted genes are important for cancer subtype classification.
8	[75]	Jovani Taveira De Souza et al.	2019	To reduce dimensions in gene expression data	AS, PCA,CSE, mRMR	Lung Cancer Michigan, Ontario, and Harvard Databases	CSE-mRMR provides better classification. PCA performs better than AS.
9	[76]	Wen Hui Yang et al.	2007	To extract features from high-dimensional gene	PCA,BGA, UDA	Feret Database, ORL database, CMU PIE	The proposed method is effective and stable for

				expression data		database (Image Data), And Gene Expression data (SRBCT, ALL, Lymphoma, Brain, Colon, Leukemia)	feature extraction.
10	[77]	Dejun Zhang et al.	2015	To propose an unsupervised feature extraction method	PCA, Autoencoder NN, AdaBoost, MAS5.0, RMA	Breast Cancer data from NCBI GEO datasets	Extracted features help to get more classification accuracy.
11	[78]	Yuchu n Tanget al.	2008	To extract the features and classify cancer from gene expression data	Granular computing, statistical learning, and fuzzy clustering, MATLAB	Prostate, AML/ALL, and Colon Cancer available at http://sdmc.lit.org.sg/GEDatasets/Datasets.html	100% accuracy with eight genes

12	[79]	Kwang Ho Park etal.	2021	To extract the features using deep learning techniques for hematopoietic cancer	Deep Autoencoder, Synthetic minority oversampling technique (SMOTE), Shapely Additive explanation (SHAP)	TCGA Hematopoietic Cancer datasets	Deep learning-based feature extraction provides good results.
13	[80]	Shutao Li et al.	2006	To develop a feature Extraction method	t-test, KPLS,	Breast cancer, ALL/AML, Leukemia, Colon Cancer	t-test in combination with KPLS provides better results.
14	[81]	Kun Yu et al.	2022	To provide a deep learning approach for feature extraction from gene expression data	GSEnet (combination of ResNet andSEnet)	GSE99095 Data set (NCBI Data)	The proposed model is superior as compared to other models in terms of accuracy and precision.

15	[82]	Raktim Kumar Mondol et al.	2022	To extract relevant genes for the classification of breast cancer from gene expression data		BRCA dataset from cBioportal	The proposed method outperforms other feature extraction methods.
16	[83]	Nikhilanand Arya et al.	2022	To classify breast cancer using a deep-learning ensemble model	Neural Network, AAE, SMOTE CNN, SVM, RF, NB, Logistic Regression (LR)	https://github.com/USTC-HIlab/MDN NMD, TCGA-BRCA	It outperforms other existing prediction methods based on uni-modality and multi-modality.
17	[146]	Md. Alamin Talukder et al.	2022	To classify lung and colon using a hybrid feature extractor and ensemble classifier	VGG16, VGG19, MobileNet, DenseNet169, DenseNet201, RF, LR, LGB, XGB, MLP, SVM	LC25000 lung and cancer dataset	Provides an accuracy of 99.05% for lung cancer, 100% for colon and 99.30% for both.

18	[147]	Jinlong Shi et al.	2010	To compare linear and non-linear dimensionality reduction	PCA, ISOMAP, K-Means, Hierarchical Clustering	SRBCT, Diffuse large B-Cell lymphoma, Nine Tumors, Brain Tumor and Leukemia Dataset	NDR such as ISOMAP provides better results than PCA for gene expression datasets.
19	[148]	Wei Liu et al.	2024	To denoise and recover scRNA gene expression data by feature extraction	Autoencoder	Nine different datasets	The proposed method performs well in clustering and gene identification
20	[149]	Qiaozhen Meng et al.	2024	To enhance the predictions for EPIs in gene expression data	Hilbert Curve, ResNet, Matching Heuristic	Benchmark datasets such as IMR90, HUVEC, HeLa-S3, K562, GM12878, NHEK	Performs better than other existing models such as SPEID, SimCNN and EPIsHilbert
21	[150]	Irfan Ali Kandhro et al.	2024	To perform feature extraction using E-	VGG19, Nvidia K80 GPU	ISIC 2020 Dataset	Provides better feature extraction which in turn

				VGG19 for improved classification of skin cancer			provides better classification
22	[151]	Wilson Bakasa et al.	2023	To extract features using VGG16 and classification using XGB to find pancreas cancer	VGG16, LGBM, Inception V3, RF, SVM, XGB	The Cancer Imaging Archive (TCIA) Dataset	Provides accuracy and F1 score of 0.97

2.4 CNN and RNN-Based Methods for Gene Expression

The changes in gene expressions show the response of the human body to certain environmental conditions, for example, any disease such as cancer in the body. While some genes are easily recognized, others are not, making them rare and widely expressed genes. To build an ideal RNN classifier and identify distinguishing features for differentiating often expressed genes from infrequently expressed genes, the incremental feature selection approach in conjunction with a supervised classifier RNN has been applied [84].

Gene expression data is unsuitable for deep CNNs due to its high dimensions and smaller sample sizes. A lightweight convolutional neural network (CNN) with two layers of convolution is used to classify breast cancer from gene expression data with an accuracy of 98.76%, overcoming the issues with the small size and high dimensionality of the gene expression data. The genomic data has been converted into 2D images before applying CNN. During preprocessing, sample outliers are removed, and then normalization is applied to remove any kind of bias. Finally, a filtering step is applied, and data has been passed to CNN for classification [37].

L-GEPM, which anticipates the target genes by utilizing learned features to capture the non-linear elements influencing gene expression, has been implemented. It is based on

LSTM neural networks. It can yield very good fitting outcomes with little error [48]. Using a deep learning strategy based on BPSO-DT and CNN, several cancer types have been diagnosed with 96.90% accuracy using tumor RNA sequence (RNA-Seq) gene expression data [38].

The creation of a multi-tissue cancer classifier using whole transcriptome gene expressions gathered from several tumor types covering various organ sites has led to the introduction of a deep learning framework for cancer diagnosis. A CNN architecture called GeneXNet has been designed particularly to deal with the complexity of gene expressions [49]. The brief length of gene expression time series affects several clustering algorithms, or they fail to take into account the temporal structure of the data. Overcoming these problems, a novel deep learning-based technique called DeepTrust is used to classify gene expression time series. To provide richer data representations, DeepTrust first converts time series data into images. Subsequently, the images undergo an application of deep neural clustering [86].

Utilizing a hierarchical graph convolution network (HiGCN), information from the feature and sample spaces has been combined. Studies using both simulated and actual data revealed that, even in the presence of few labeled data points, HiGCN enhanced the performance of the downstream tasks (which include things like survival analysis and classification). HiGCN also obtained more efficient representations of the gene expression data. The proposed method can also extract features from noisy and low-complexity data. [88].

Improved prevention, detection, and treatment of breast cancer can be achieved by deeper research of the molecular features of the disease, made possible by the advent of deep learning techniques and multidimensional data. A multimodal deep neural network that makes use of multi-dimensional data (MDNNMD) has been put into practice for breast cancer prognostic prediction. This method predicts the time of survival of breast cancer. The proposed DNN consists of four hidden layers along with an input and an output layer [89].

Various machine learning and deep learning models are available that are used for gene

expression data. However, in [90], the author stated that deep learning models provide more promising results as compared to conventional machine learning models inthe classification of cancer for gene expression data. Gene expression datasets still suffer from a problem of low sample size. But DL methods require large amounts of data for which certain methods are used such as regularization, data augmentation, decreasing the complexity of the NN, etc. But still, these things do not solve this issue completely.

Microarray data has risen in size over time, but its challenging characteristics—such as the small size of the sample and high complexity—have remained steady. More sophisticated algorithms for gene selection and classification must be designed to address these challenges. To accurately diagnose cancer, a hybrid technique utilizing deep fuzzy neural networks has been developed to preprocess data and select critical genes. This method outperforms other traditional classification techniques [91].

DeepDRBP-2L, a two-level predictor, has been proposed by combining CNN and LSTM. The ability to differentiate between RNA-Binding Proteins (RBPs), DNA-Binding Proteins (DBPs), and DRBPs (both DNA RNA Binding Proteins) is unique to this computational approach. DeepDRBP-2L may identify DRBPs beyond the limitations of the current approaches, as demonstrated by extensive cross-validations and independent testing [92].

Two issues are directly related to the diagnosis of cancer. One is preserving data privacy and the second is a diagnosis of cancer itself. The MRI images of cancer patients are encrypted to save them from intruders for the patient's privacy. When any operation such as cancer diagnosis has to be performed on those images, then before this, images are decrypted using various techniques. So, this job is quite tedious. In [93], the author has proposed a method based on CNNs for this purpose. The proposed method has been compared with conventional ML models such as DT, NB, RF, and SVM and it provides an accuracy of 98.9%.

The correct segmentation of liver tumors is required so that tumors are diagnosed and treated on time. The segmentation also helps surgeons to make the correct decisions during surgeries. This motivates the author of [94] to develop a method based on 3D CNN and

convolution LSTM to perform liver tumor segmentation in MRI scan images. The MRI scan images are taken from two hospitals and the research work has been validated by the research ethics committee of one of those hospitals.

In [152], the author compared various DL techniques such as CNN, LSTM RNN, hybrid CNN-RNN, and GRU RNN to classify cancer from gene expression data. The data is normalized and then Bayesian optimization has been applied for tuning of hyperparameters which are further required to strengthen the classification process by DL models. The DL models are ensembled with a decision tree which is a classification and regression tree (CART) for final decision-making. Among all the models, two-layer GRU RNN achieved the highest accuracy of 97.8%.

Gene expression data contains fewer samples than other datasets, so data augmentation is one of the solutions to generate synthetic samples. In [153], the author integrated uniform distributive augmentation (UDA) and Wasserstein – Generative Adversarial Network (W-GAN) for data augmentation. Then classification has been done by Capsule neural networks (CapsNet). The proposed method provides promising results as compared to other DL methods.

In [155], the author proposed a deep learning-based classifier to classify cancer from the gene expression dataset. For the proposed work, data has been taken for five different cancers. Features have been normalized using min-max normalization. Then the most significant and non-redundant genes were selected using an enhanced chimp optimization (ECO) algorithm. Finally, depth-wise separable CNN (DSCNN) has been applied to the selected genes for the classification of cancerous and non-cancerous classes. The proposed approach has been compared with other DL methods such as CNN, DCNN, LSTM, and GRU and proved to be the best with 99.18% accuracy.

Prostate cancer is also one of the major reasons for increasing mortality rates. However, it should be early detected to reduce the mortality rate. In [156], the author proposed an AI-based technique for classifying prostate cancer from gene expression data. Data has been normalized first and then passed to LSTM – deep belief network (LSTM-DBN) for classification. The proposed model has been fine-tuned using wild horse optimization

(EWHO) hyper-parameter to increase the performance. The proposed model can handle noisy and missing data well. The proposed model has been compared with various existing ML and DL models.

In [157], the author proposed two new methods to classify cancer types and sub-types from the gene expression dataset which is available on Kaggle and contains 130000 images of various cancer types. Two novel CNN methods have been proposed namely Vception and Vmobilenet which contain VGG and Inception, and VGG and MobileNet each of which has been integrated later with LSTM for both spatial and temporal pattern detection. These models predict cancer type and sub-type together. But classification has been done in other ways as well where cancer type has been predicted first and then cancer subtype. In this classification approach, the author used KNN and PCA for cancer subtype classification.

Table 2.3 CNN and RNN Methods for Gene Expression

Sr No	Paper Name	Author	Year	Objective	Technique /Tool	Dataset	Findings
1	[84]	Lei Chenet al.	2018	To classify rarely and widely expressed genes using RNN	RNN, Incremental Feature Selection (IFS)	TCGA [85]	Genes are successfully classified as normal and cancerous.
2	[37]	Murtada K. Elbashir et al.	2019	To classify cancer using lightweight CNN	CNN, Normalization, Filtering, R Studio	Breast cancer dataset from Pan-Cancer Atlas	CNN provides good accuracy as compared to other current approaches such as PCA and SVM.

3	[86]	Ozan Firat Ozgul et al.	2021	To propose a deep convolution-based clustering method for time series gene data	Convolutional Autoencoder, ReLU, Adam Optimizer	Dataset from [87]	DeepTrust provides reliable clusters than other methods for the time series data.
4	[88]	Kaiwen Tan et al.	2021	To propose a graph method based on convolution network for finding the information in feature as well as sample space	Graph Convolution Network	2 simulated and 2 real-time datasets from Pan Atlas	Provides good results even with small training data.
5	[89]	Dongdong Sun et al.	2019	To propose a DNN-based model for	mRMR, DNN, Batch Normalization	METABRIC Dataset available at http:// www.cbiopor	Proposed model can be applied to other similar

				multi-dimensional cancer gene expression data		tal.org/study?id¼brca_metabric#summary.	type of diseases
6	[90]	Mahmood Khalsan et al.	2022	To compare various ML and DL models using gene expression data for cancer prediction	Various ML and DL techniques are studied	Multiple datasets are studied available on TCGA and GEO	DL techniques provide better results than ML techniques for cancer prediction in gene expression data.
7	[91]	Thosini K. Bamunu Mudiyansela ge et al.	2020	To propose a deep neural network-based method for cancer detection	Relevance Indexed Based Filtering, K-Means Clustering ,SVM, DT,NB	Cancer datasets of colon, leukemia, prostate, CNS, ovarian, and breast	The proposed method is better and more reliable than other traditional methods
8	[92]	Jun Zhang et al.	2021	To propose a predictor method based on	CNN, LSTM	Benchmark and independent dataset available at http://bliulab.	It solves the problem of existing predictors that DBPs and

				CNN and LSTM		n et/DeepDRBP-2L	RBPs and vice-versa.
9	[93]	Mujeeb Ur Rehman et al.	2022	To develop a DL-based model for preserving privacy while diagnosing cancer	CNN, Discrete Wavelet Transform (DWT)	MRI Scan Images	The CNN-based model provides more accuracy than conventional ML models
10	[94]	Rencheng Zheng et al.	2022	To propose a model based on 3D CNN and convolution LSTM for the segmentation of liver tumor	3D CNN, 4 Layer Convolution LSTM	MRI scan images from First Affiliated Hospital of Zhejiang University, Test data from Fudan University Affiliated Zhongshan Hospital	The proposed method provides correct segmentation results.
11	[152]	Sergii Babichev et al.	2024	To compare various DL models for cancer	One- and two-layer CNN, LSTM RNN, GRU	TCGA	Two-layer GRU RNN performed best among all with 97.8% accuracy

				classification	RNN, hybrid CNN RNN, Bayesian Optimization, CART classifier		
12	[153]	U. Ravindran et al.	2024	To perform smart data augmentation and classification	UDA, W-GAN, Pearson's Correlation Coefficient, CapsNet	Gene Expression dataset at [154]	Provides 99.32% accuracy, 98.56% recall, and 100% precision
13	[155]	Anju Das et al.	2024	To perform gene selection and classification using deep learning	Normalization, ECO, DSCNN, Python	ALL AML, brain, breast, colon and prostate cancer dataset	Provides overall accuracy of 99.18% for all the 5 datasets
14	[156]	Bijya Kumar Sethi et al.	2024	To classify prostate cancer from gene expression data	Normalization, DBN, LSTM, EWHO, Python	Prostate cancer dataset	Provides accuracy of 98.75%

15	[157]	Jabed Omar Bappi et al.	2024	To classify cancer type and sub-type using DL and ML methods	VGG, Inception, MobileNet, LSTM, KNN, PCA, X-OR Gate	Kaggle Dataset for breast, colon, kidney, lymphoma, oral, brain, cervical and lung cancer	Provides 99.25% accuracy for cancer type and 97.20% for cancer subtype classification

2.5 Results and Discussion

We examined several deep learning and machine learning models for the gene expression dataset available at https://data.mendeley.com/datasets/sf5n64hydt/1. The dataset consists of 2086 samples with 971 genes for each sample. There are 5 types of cancer that we focused on. These include LUAD (lung adenocarcinoma), LUSC (lung squamous cell carcinoma), UCEC (uterine corpus endometrial carcinoma), BRCA (breast invasive carcinoma), and KIRC (kidney renal clear cell carcinoma). We implemented our work in Python on Google Colab. Fig 2.1 to 2.3 represents the raw data which is downloaded from the link given above. Samples are represented with numbers from 0 to 2085 which makes the count of patients 2086. Genes or the features are represented with numbers from 0 to 970 which makes the features, or the genes count as 971. The last column, 971, is the class column that represents the different cancer types with numbers from 1.0 to 5.0. Column 971 has been renamed as a target column to specify the given column as a class column of the dataset based on which the final classification has been done. Fig 2.4 shows thetarget column highlighted with red color. Instead of 971, it is written as a target. As a nextthing in preprocessing, 10 columns are removed from the dataset as these consist of repetitive values. Those 10 columns are 961 to 970 making the total count of features/genesequal to 960. The modified dataset after removal of the 10 columns is shown in Fig 2.5 and 2.6.

	0	1	2	3	4	5	\
0	44.023542	9.216286	11.319078	33.215176	16.901427	9.031338	
1	29.746157	9.765600	40.540128	30.169134	20.047393	32.237287	
2	35.799315	9.884781	3.886043	29.984211	17.135946	21.273727	
3	26.490401	7.085828	10.804003	23.482255	17.044085	14.880104	
4	27.632466	7.642971	3.670265	16.584843	20.375321	22.174600	
...	...	...	...	...	...	...	
2081	23.758989	15.394219	10.883932	21.691852	20.209391	30.078659	
2082	27.990210	28.998590	8.701462	27.579071	29.770012	15.744797	
2083	14.665414	20.195646	6.703477	19.648529	12.530305	24.321260	
2084	31.116022	14.562066	9.121585	13.831678	15.535040	41.278765	
2085	74.046628	13.003962	9.472274	37.000833	19.295983	13.590392	

	6	7	8	9	...	962	963	\
0	1.109961	20.017821	16.724363	10.494192	...	21.706486	16.315579	
1	2.460624	17.029112	28.346167	17.017284	...	12.815215	10.150965	
2	1.501203	20.598204	25.855152	12.275738	...	14.344729	11.224647	
3	1.299056	14.978582	31.214294	10.015235	...	13.660995	9.730124	
4	1.553541	14.909150	54.435490	13.392213	...	16.650019	8.584938	
...	...	...	...	...	...	...	...	
2081	0.978559	13.002383	28.629486	6.387968	...	10.900725	4.188131	
2082	3.759037	13.468529	20.548527	5.557939	...	13.409906	8.146828	
2083	2.263398	9.642926	30.248579	22.157856	...	4.310888	7.920039	

Fig 2.1 Sample of dataset from sample (rows) 0 to 2083 and genes (columns) from 0 to 963

2084	1.044817	8.012867	8.701291	4.777847	...	5.303415	3.716170
2085	0.861233	16.029999	31.036571	7.036183	...	12.484386	9.730580

	964	965	966	967	968	969	970	\
0	4.224009	8.602081	23.762341	8.302416	1.408731	4.295620	8.768768	
1	8.914809	6.797915	15.379187	11.420690	6.599729	3.819019	5.758501	
2	7.870991	7.724003	25.762396	8.628786	4.104879	4.382387	5.306177	
3	7.804760	5.030966	8.964868	7.990036	4.251886	3.702483	7.500498	
4	7.485410	5.945771	9.205302	8.761025	4.656969	3.827945	7.939863	
...	...	...	...	...	...	...	...	
2081	2.499783	2.422833	10.061337	3.966998	2.323246	42.449235	2.661792	
2082	3.390904	3.621281	18.706152	4.272848	1.199481	3.990134	1.665184	
2083	2.731255	4.168946	4.470333	3.064729	1.341491	1.830216	2.355292	
2084	1.802522	1.466401	4.209371	2.633839	0.908784	1.176131	1.891297	
2085	6.713517	3.953366	9.271268	4.530131	4.252967	4.337182	9.333152	

Fig 2.2 Sample of dataset from sample (rows) 0 to 2085 and genes (columns) from 964 to 970

```
      971
0     1.0
1     1.0
2     1.0
3     1.0
4     1.0
...   ...
2081  5.0
2082  5.0
2083  5.0
2084  5.0
2085  5.0

[2086 rows x 972 columns]
```

Fig 2.3 Target Column (Classes of Cancer)

```
1     8.914809  6.797915  15.379187  11.420690  6.599729   3.819019  5.758501
2     7.870991  7.724003  25.762396   8.628786  4.104879   4.382387  5.306177
3     7.804760  5.030966   8.964868   7.990036  4.251886   3.702483  7.500498
4     7.485410  5.945771   9.205302   8.761025  4.656969   3.827945  7.939863
...        ...       ...        ...        ...       ...        ...       ...
2081  2.499783  2.422833  10.061337   3.966998  2.323246  42.449235  2.661792
2082  3.390904  3.621281  18.706152   4.272848  1.199481   3.990134  1.665184
2083  2.731255  4.168946   4.470333   3.064729  1.341491   1.830216  2.355292
2084  1.802522  1.466401   4.209371   2.633839  0.908784   1.176131  1.891297
2085  6.713517  3.953366   9.271268   4.530131  4.252967   4.337182  9.333152

      target
0        1.0
1        1.0
2        1.0
3        1.0
4        1.0
...      ...
2081     5.0
2082     5.0
2083     5.0
2084     5.0
2085     5.0

[2086 rows x 972 columns]
```

Fig 2.4 Renaming of column 971 as target

	0	1	2	3	4	5	\
0	44.023542	9.216286	11.319078	33.215176	16.901427	9.031338	
1	29.746157	9.765600	40.540128	30.169134	20.047393	32.237287	
2	35.799315	9.884781	3.886043	29.984211	17.135946	21.273727	
3	26.490401	7.085828	10.804003	23.482255	17.044085	14.880104	
4	27.632466	7.642971	3.670265	16.584843	20.375321	22.174600	
...	...	...	...	...	...	...	
2081	23.758989	15.394219	10.883932	21.691852	20.209391	30.078659	
2082	27.990210	28.998590	8.701462	27.579071	29.770012	15.744797	
2083	14.665414	20.195646	6.703477	19.648529	12.530305	24.321260	
2084	31.116022	14.562066	9.121585	13.831678	15.535040	41.278765	
2085	74.046628	13.003962	9.472274	37.000833	19.295983	13.590392	

	6	7	8	9	...	952	953	\
0	1.109961	20.017821	16.724363	10.494192	...	12.183537	97.219006	
1	2.460624	17.029112	28.346167	17.017284	...	20.268603	1100.969122	
2	1.501203	20.598204	25.855152	12.275738	...	22.455759	1044.965942	
3	1.299056	14.978582	31.214294	10.015235	...	12.250125	1798.786942	
4	1.553541	14.909150	54.435490	13.392213	...	17.028034	1698.772653	
...	...	...	...	...	...	...	...	
2081	0.978559	13.002383	28.629486	6.387968	...	7.454885	51.888541	
2082	3.759037	13.468529	20.548527	5.557939	...	14.881666	63.480940	
2083	2.263398	9.642926	30.248579	22.157856	...	18.221962	75.960478	
2084	1.044817	8.012867	8.701291	4.777847	...	6.753062	55.551521	
2085	0.861233	16.029999	31.036571	7.036183	...	15.167092	63.612161	

Fig 2.5 Dataset representing features from 0 to 953

	954	955	956	957	958	959	\
0	23.011699	15.462851	23.105338	38.696649	16.412698	14.100348	
1	17.763978	18.222554	26.112132	53.158154	24.657516	15.957426	
2	19.992626	23.268570	30.432389	74.463368	17.926723	13.988009	
3	18.344660	13.925511	31.476416	54.208713	21.799192	22.550758	
4	13.155730	10.879576	39.061579	41.811446	20.874913	16.554005	
...	...	...	...	...	...	...	
2081	15.247597	7.091441	25.414310	17.196127	16.471856	27.450341	
2082	14.784845	19.000876	48.333457	24.511178	40.871939	9.596640	
2083	25.268377	5.633854	15.302076	12.837028	23.031890	25.189686	
2084	11.135564	4.009023	22.244020	10.238997	10.014452	6.504573	
2085	18.457928	7.773370	42.215062	25.774330	21.962225	10.795606	

	960	target
0	89.149907	1.0
1	58.026493	1.0
2	26.157333	1.0
3	58.898222	1.0
4	38.204280	1.0
...	...	...
2081	37.445203	5.0
2082	17.868032	5.0
2083	75.656297	5.0
2084	240.639610	5.0
2085	133.199096	5.0

Fig 2.6 Dataset representing features from 954 to 960 and target column

In the next step of pre-processing, cancer types were represented using numbers from 0.0 to 4.0 instead of 1.0 to 5.0 as shown in Fig 2.7.

```
              960  target
0       89.149907     0.0
1       58.026493     0.0
2       26.157333     0.0
3       58.898222     0.0
4       38.204280     0.0
...           ...     ...
2081    37.445203     4.0
2082    17.868032     4.0
2083    75.656297     4.0
2084   240.639610     4.0
2085   133.199096     4.0

[2086 rows x 962 columns]
```

Fig 2.7 Classes of cancer represented using numbers from 0.0 to 4.0

Afterward, the target column has been eliminated from the original dataset and preserved as a separate set of data with distinct features and labels. Implementation of machine learning algorithms requires some basic steps as discussed in the previous chapters as well. These consist of data collection, data exploration, splitting into training testing, implementing the required algorithm, and then evaluating it. The first two steps are done. Next comes the splitting. As a result, we divided our dataset into 30% for testing and 70% for training. Next, we utilize various machine learning classifiers, including logistic regression, gradient boosting, random forest, support vector, and xgb classifiers.

Table 2.4 Performance Measures for Machine Learning Models

	Accuracy	F1 Score	Precision	Recall	Mean Square Error
Random Forest	96.64	96.61	96.63	96.62	9.10

Support Vector	92.65	92.50	92.71	92.70	37.70
Logistic Regression	95.68	95.71	95.70	95.69	19.30
Gradient Boosting	95.84	95.91	95.85	95.86	13.71
XGB Classifier	97.12	97.11	97.13	97.10	6.51

Table 2.4 shows the various performance measures for machine learning models. Various performance measures that are taken into account are accuracy, F1 score, precision, recall, and mean square error. The above table shows that the XGB classifier is performing best among the given classifiers in terms of all the performance measures. Fig 2.8 shows the comparison of machine learning models based on accuracy parameters. XGB classifier has the highest accuracy of 97.12 and the support vector has the lowest accuracy of 92.65. Fig 2.9 shows the comparison of machine learning models based on t h e F1 score.XGB classifier has the highest F1 score of 97.11 and the support vector has the lowest F1score of 92.50.

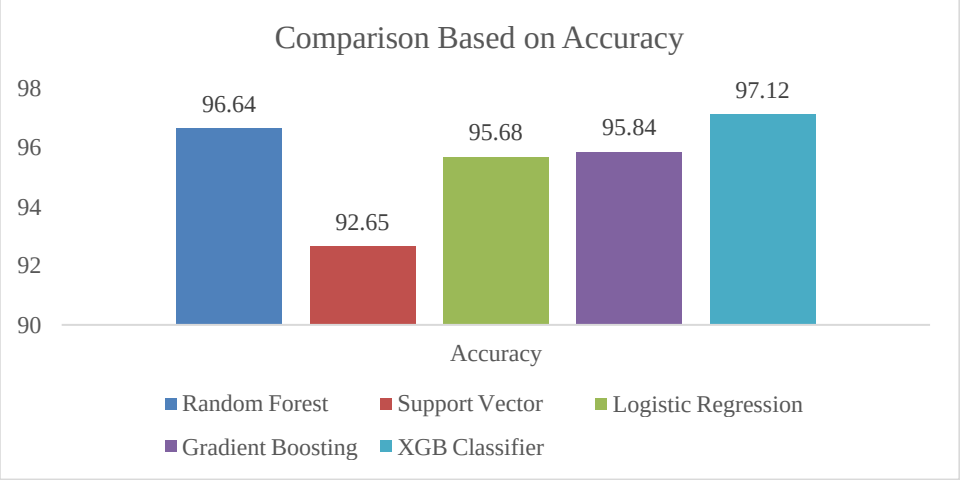

Fig 2.8 Comparison of Machine Learning Models based on accuracy

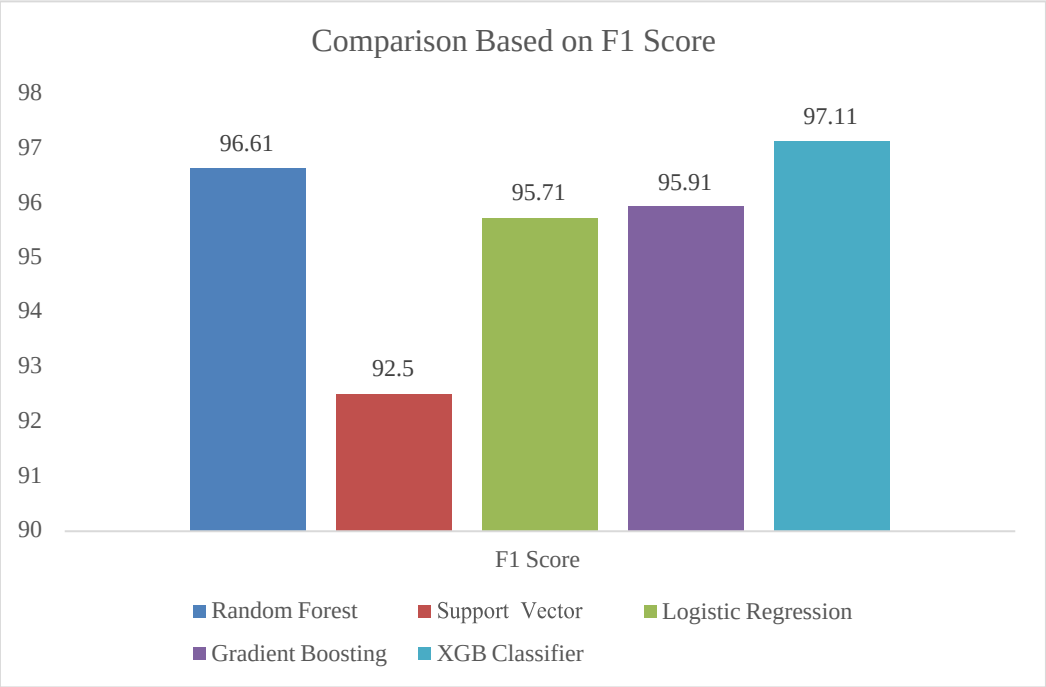

Fig 2.9 Comparison of Machine Learning Models based on F1 Score

Fig 2.10 shows the comparison of machine learning approaches based on precision. XGB classifier has the highest precision of 97.13 and the support vector has the lowest precision of 92.71.

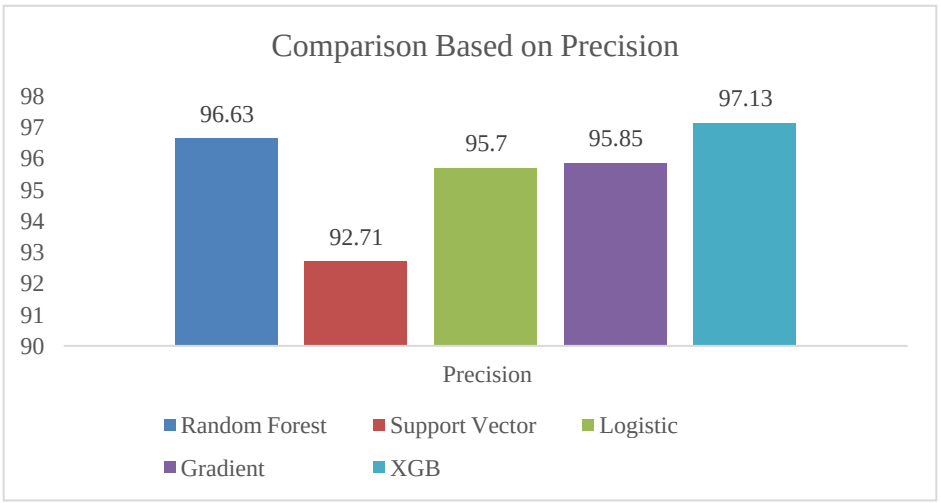

Fig 2.10 Comparison of Machine Learning Models based on Precision

Fig 2.11 shows the comparison of machine learning approaches based on recall. XGB classifier has the highest precision of 97.10 and the support vector has the lowest precision of 92.70.

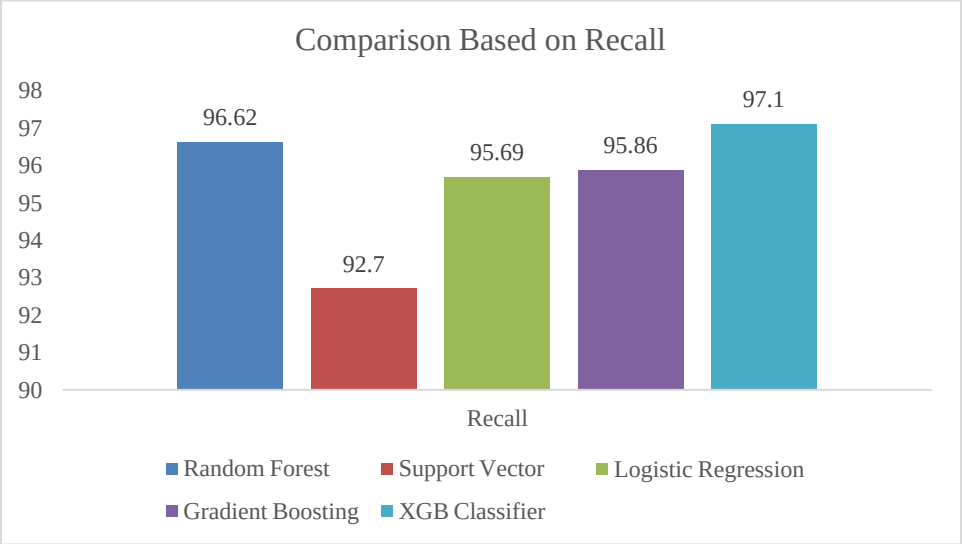

Fig 2.11 Comparison of Machine Learning Models based on Recall

Fig 2.12 shows the comparison of machine learning approaches based on MSE. XGB classifier has the lowest MSE of 6.51 and the support vector has the highest MSE of 37.70.

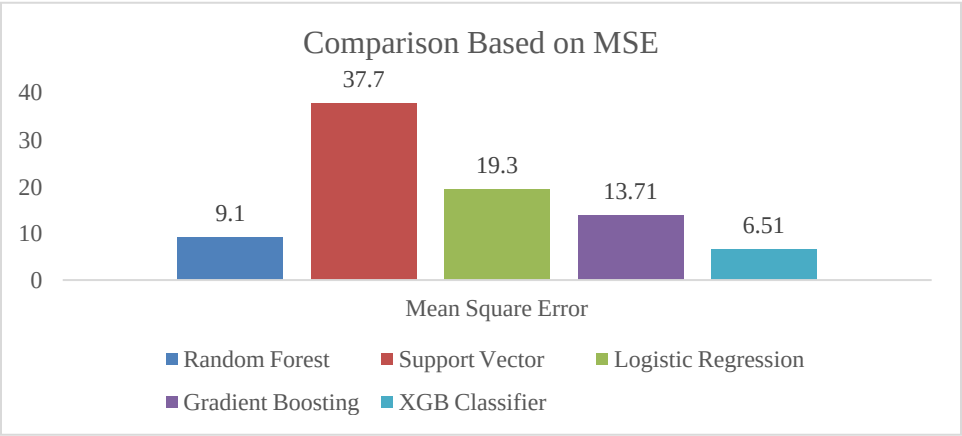

Fig 2.12 Comparison of Machine Learning Models based on MSE

We also compared various CNN models on the same dataset on which we implemented the machine learning models. To implement the CNN models, every sample is converted into an image. Fig 2.13 shows one of those images.

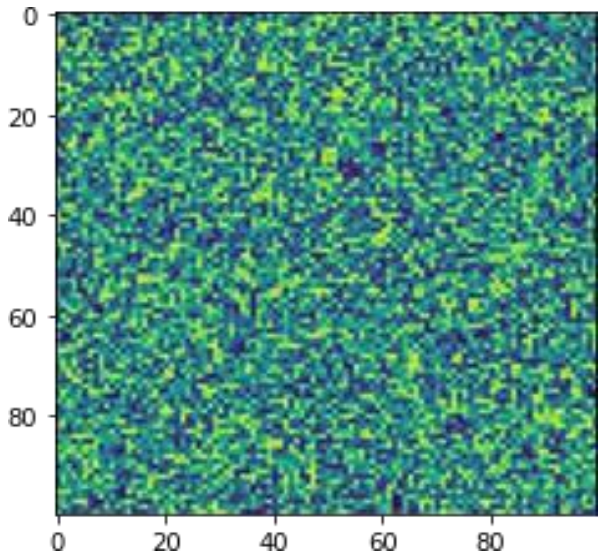

Fig 2.13 A sample of genes converted into an image

Table 2.5 shows the performance measures for various CNN models such as vgg16, vgg19, inceptionv3, and resnet50. VGG16 is performing best among all in terms of all the given parameters.

Table 2.5 Performance Measures for CNN Models

	Accuracy	F1 Score	Precision	Recall	Mean Square Error
VGG16	91.69	91.21	91.72	91.71	39.50
VGG19	43.45	26.34	43.51	43.54	336.7
ResNet50	81.30	82.41	81.32	81.35	95.0

InceptionV3	43.45	28.21	43.53	43.55	324.80

Fig 2.14 shows the bar chart representing the accuracies for various CNN models. VGG16 provides the highest accuracy of 91.69 and VGG19 and InceptionV3perform poorly, providing an accuracy of 43.45.

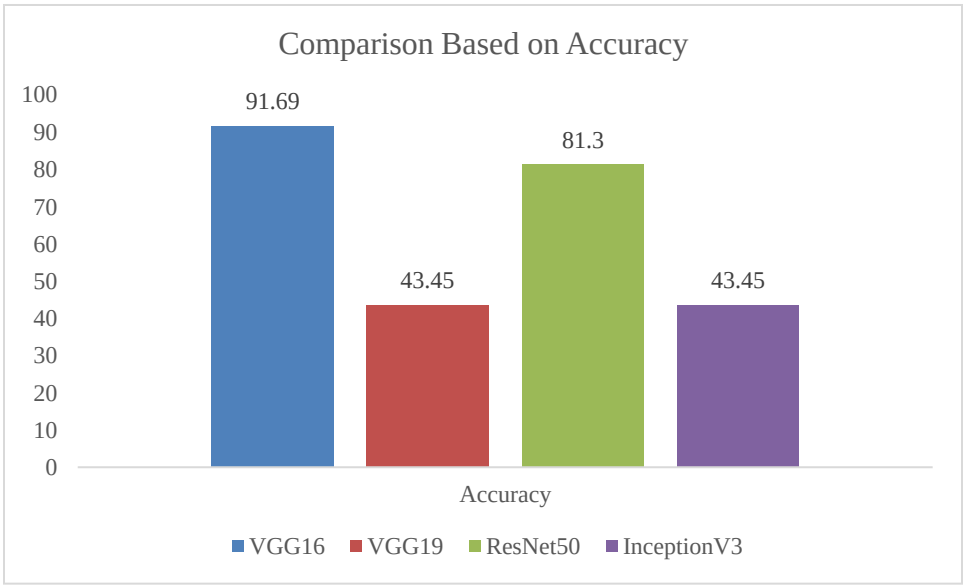

Fig 2.14 Comparison of CNN Models Based on Accuracy

Fig 2.15 shows the F1 score comparison of various CNN models for 70% training and 30% testing data. VGG16 provides the highest F1 score of 91.21 and VGG19 provides the lowest F1 score of 26.34.

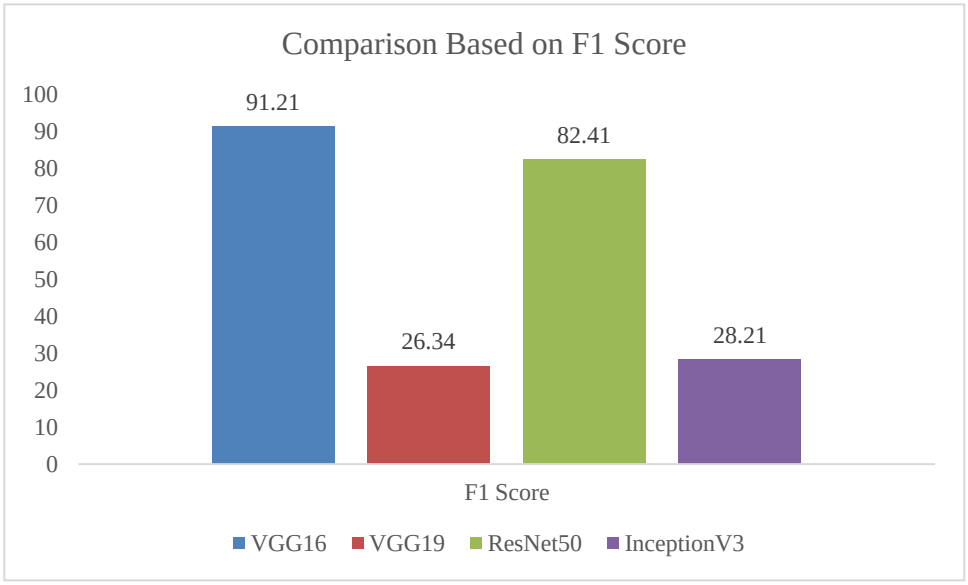

Fig 2.15 Comparison of CNN Models Based on F1 Score

VGG16 provided the highest precision of 91.72 and VGG19 provided the lowest precision of 43.51 as shown in Fig 2.16.

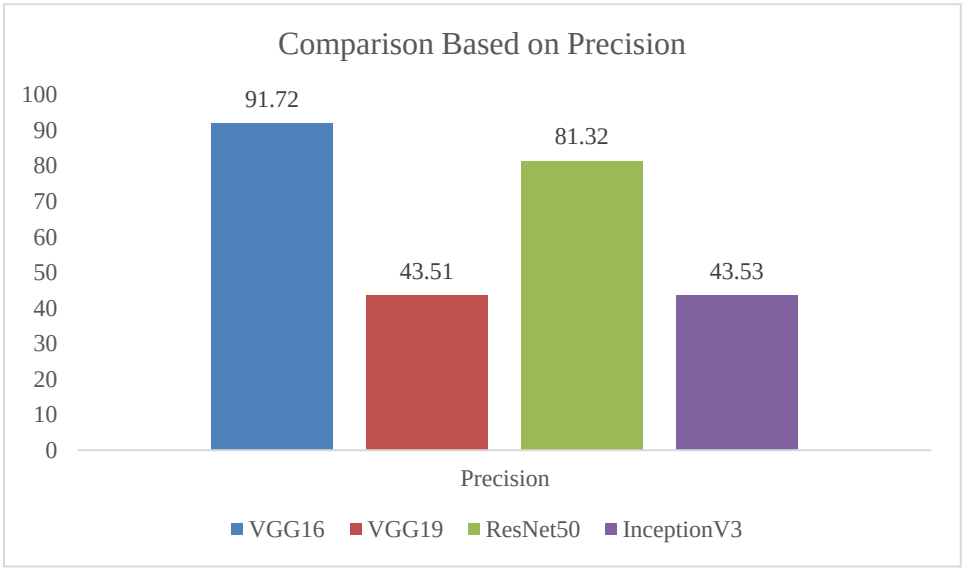

Fig 2.16 Comparison of CNN Models Based on Precision

Fig 2.17 shows the comparison of various machine learning models based on their recall value. VGG16 gives the highest recall value of 91.71 and VGG19 provides

the lowest recall of 43.54. As illustrated in Fig. 2.18, VGG16 yields the lowest MSE of 39.5 and VGG19 yields the greatest MSE of 336.7.

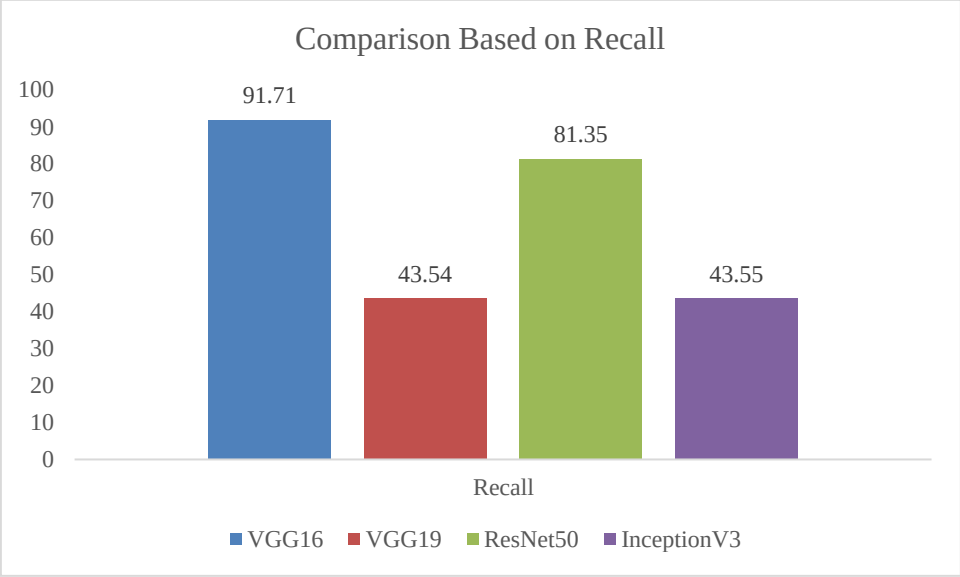

Fig 2.17 Comparison of CNN Models Based on Recall

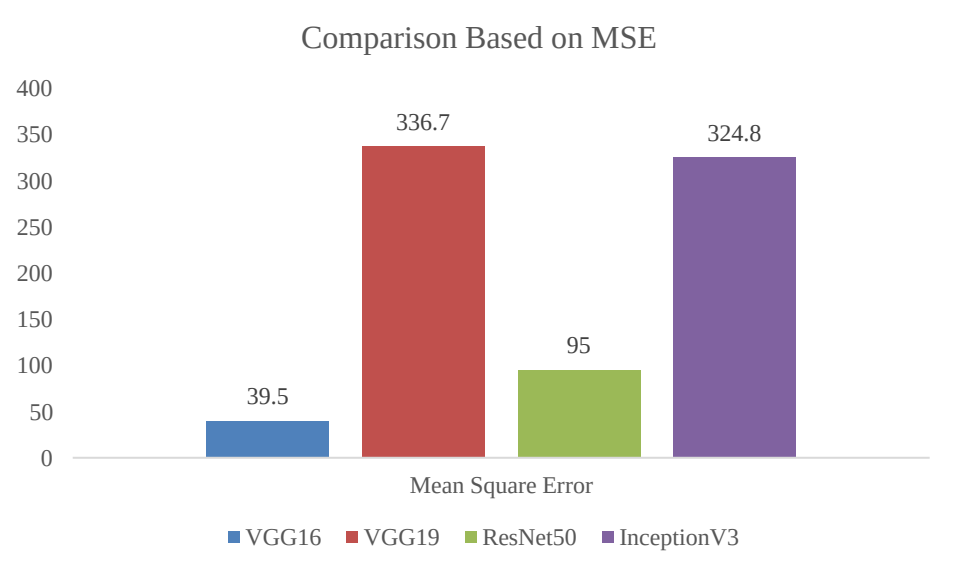

Fig 2.18 Comparison of CNN Models Based on MSE

After comparing various machine learning and CNN models, various RNN models such as simple RNN, GRU, and LSTM are compared. Table 2.6 shows various RNN models as per the performance metric values.

Table 2.6 Performance Measures for RNN Models

	Accuracy	F1 Score	Precision	Recall	Mean Square Error
Simple RNN	92.49	92.2	92.5	92.5	33.4
LSTM	94.56	94.5	94.6	94.6	31.9
GRU	95.04	94.9	95	95	21.7

As shown in Fig 2.19, GRU provides the highest accuracy of 95.04 and simple RNN provides the lowest accuracy of 92.49.

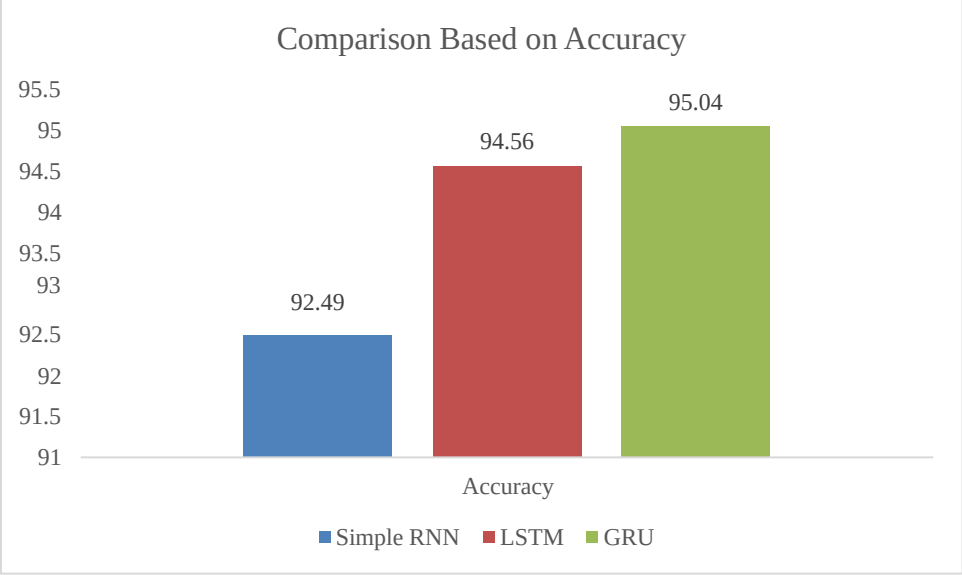

Fig 2.19 Comparison of RNN Models Based on Accuracy

GRU provides the highest F1 Score among all the RNN models which are compared based on various performance metrics. It gives the F1 Score of 94.9. Simple RNN, on the other hand, has the lowest F1 Score (92.2) as shown in Fig 2.20. Similarly, GRU provides the highest precision of 95 and simple RNN provides a 92.5 precision value which is the lowest among all as shown in Fig 2.21.

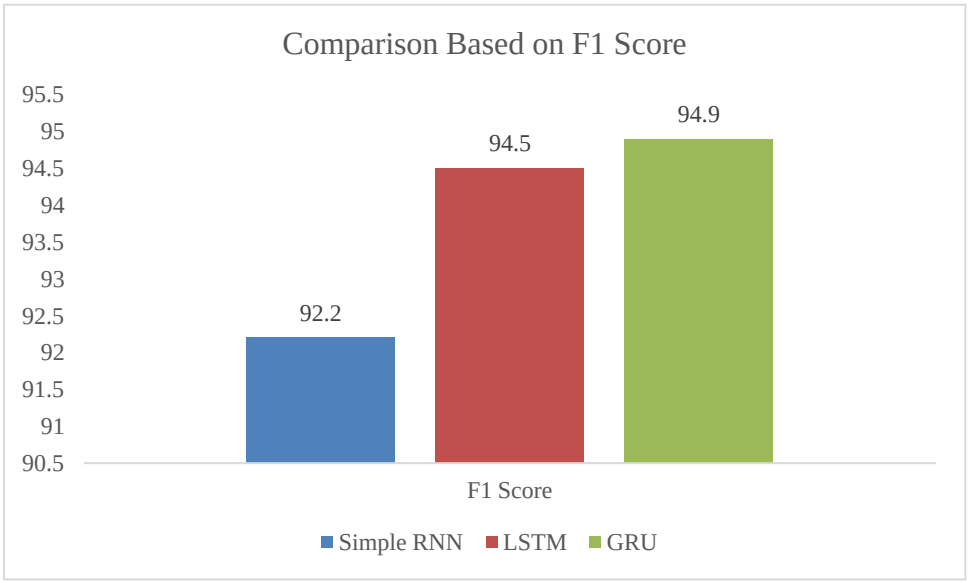

Fig 2.20 Comparison of RNN Models Based on F1 Score

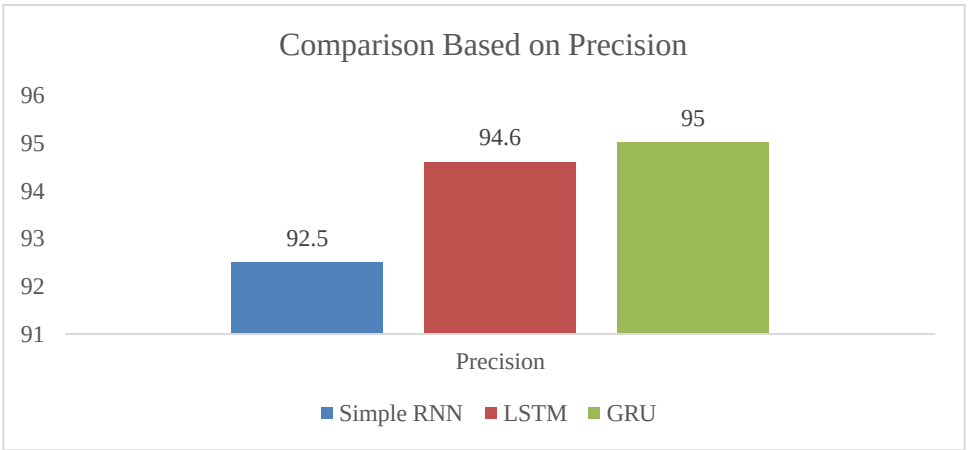

Fig 2.21 Comparison of RNN Models Based on Precision

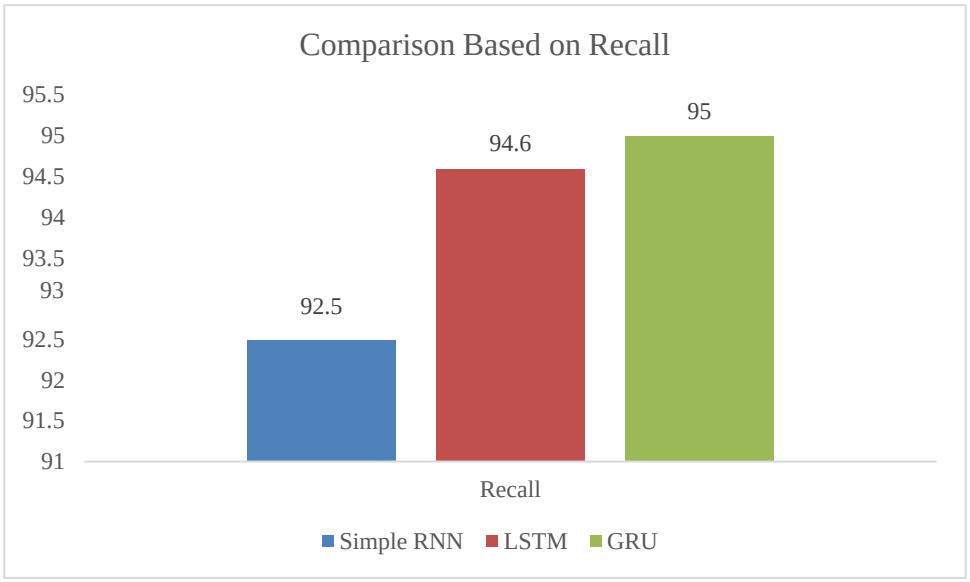

Fig 2.22 Comparison of RNN Models Based on Recall

Figures 2.22 and 2.23 demonstrate how GRU outperforms other models in terms of recall and MSE as well. GRU provides the highest recall of 95 and the lowest MSE of 21.7. Simple RNN, on the other hand, has the greatest MSE of 33.4 and the lowest recall of 92.5.

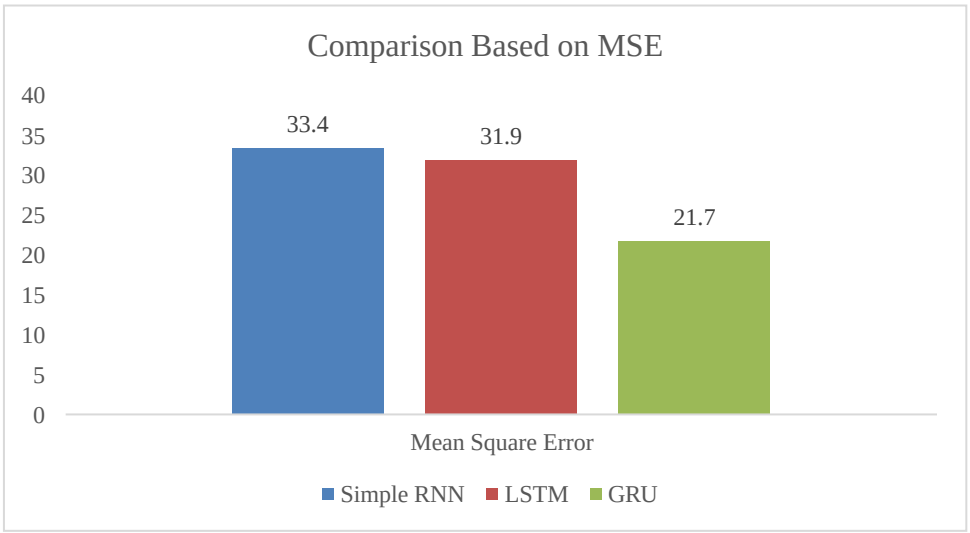

Fig 2.23 Comparison of RNN Models Based on MSE

2.6 Summary

The beginning of this chapter gives insights into different deep learning and machine learning methods for gene expression data. It has been discovered from the literature that high dimensionality and limited sample sizes are issues with gene expression data. That is why in the second section, various feature extraction techniques have been discussed. Then for the classification of cancer, various CNN and RNN models are discussed in the last section. Various methods of machine learning, CNN, and RNN have been compared. It has been discovered that deep learning models outperform machine learning models in terms of effectiveness.

CHAPTER 3
RESEARCH OBJECTIVES

This chapter highlights the different research gaps that we identified after reviewing several machine learning and deep learning techniques for the analysis of gene expression data, feature extraction techniques, and cancer classification approaches. Also, this chapter presents research objectives that we framed on the basis of gaps that we found in our literature.

3.1 Introduction

One of the scariest illnesses is cancer. Its timely detection and treatment is very important. From the literature, it is concluded that there are so many techniques depending on artificial intelligence to predict and classify cancer. It has been found from the literature that classifying cancer on the basis of gene expression is very useful as it provides additional information as well. There are so many methods that are using gene expression data to classify cancer. Some of these techniques include computational analysis of gene expression, ANN, compound variate prediction method, PCA, FA, Support vector machine, Convolutional Neural networks, and many more. All these methods have their advantages and disadvantages. However, it is found that CNN is playing a good role in cancer class classification and discovery.

3.2 Research Gaps

From the whole literature survey that has been done in the previous chapter, the following are the research gaps that have been found:

- It has been observed that the gene expression additionally contains some extra information that enhances the cancer diagnostic and classification process. Thus, it indicates that we can use the gene expression data to improve our outcomes.

- Based on the tumor RNA sequence gene expression dataset, the author of that article [38] classified several types of cancer using CNN and BPSO-DT. They implemented a 5-layered neural network architecture with a 2086 sample dataset. However, a multilayer CNN architecture is also available [37] and [49] for even more accuracy.
- Since the dataset is smaller in [46]–[52], It has been noted that machine learning algorithms are being used. However, these methods would lead to an overfitting issue on the training dataset if there is big data. Therefore, we can achieve better results by using better algorithms.
- Additionally, by utilizing the extensive dataset and suggested techniques, we can enhance performance by selecting relevant features [38].
- It has also been found in [38] that the performance can be evaluated using precision, recall, and F1 score. But we can consider the other parameters also such as accuracy, sensitivity, and specificity [79].
- It has also been found that gene expression datasets suffer from a problem of fewer samples and high dimensions. To overcome this problem, dimensionality reduction is required.

3.3 Research Objectives

These are the objectives of the research to be carried out:

- To study and analyze the existing deep learning approaches to address the challenges.
- To develop suitable Feature Extraction Method for Cancer Prediction
- To design the hybrid model for cancer prediction using RNN and CNN for gene expression.
- To compare and evaluate the proposed hybrid model with existing approaches using standard metrics.

3.4 Summary

This chapter presents the research gaps and objectives based on the given research gaps. It has been observed that deep learning methods yield favorable results when applied to gene expression data. Therefore, we decided to develop a hybrid model based on deep learning models for the categorization of cancer. Dimensionality reduction is necessary since gene expression data also suffers from short sample sizes and large dimensions. In support of this, we also decided to make another hybrid model based on deep learning for feature extraction.

CHAPTER 4

PROPOSED METHODOLOGY

This chapter describes the research methodology that has been used to achieve the objectives of our research work. The methodology consists of the basic steps such as loading of data, data exploration, splitting into training and testing, implementing the desired models, and evaluating the model using different performance measures.

4.1 Introduction

Based on the literature survey, we found that when working with gene expression data, it helps to provide more accurate results while diagnosing cancer. And deep learning models work well with this data. So in lieu of this, we framed 4 objectives. For the first objective, we reviewed various machine learning and deep learning models, dimensionality reduction methods, and RNN-CNN-based classification models. For achieving the other three objectives, in the next section, a methodology has been given.

4.2 Methodology

This research proposes two models. One is for extracting features from gene expression data, and another is for the classification of cancer using gene expression datasets. It has been found from the literature that there are so many techniques available for the purpose such as FA, PSO, SVM, RF, CNN, and RNN. However, we have chosen CNN and RNN because of their advantages over other existing methods. The proposed method will be compared with other existing methods to show its performance.

The proposed methodology consists of the following steps which are implemented on three different gene expression datasets of cancer:

Step I. Data Collection: Three different Gene expression datasets have been collected from the online sources which are available at https://data.mendeley.com/datasets/sf5n64hydt/1,

https://archive.ics.uci.edu/ml/datasets/gene+expression+cancer+RNA-Seq# and https://www.kaggle.com/datasets/brunogrisci/breast-cancer-gene-expression-cumida.

Step II. Data Preprocessing: As per the requirements based on datasets, several preprocessing steps have been performed such as removal of unwanted columns, naming the columns, labeling of cancer classes, etc.

Step III. Split Data into train and test data: The dataset has been divided into training and testing subsets for several ratios, including 70:30, 80:20, and 60:40.

Step IV. Image Scaling: Gene expression data has been converted into image (2D Data) and image scaling has been performed.

Step V. Bottleneck Feature Extraction Hybrid Model: Pretrained models such as VGG16 and Vgg19 have been used to create a hybrid model for extracting the bottleneck features

Step VI. Classification: XGB Classifier has been used to classify the data after bottleneck feature extraction.

Step VI. Hybrid RNN-CNN: A hybrid model has been created using RNN and CNN to perform the final classification.

Step VII. Model Evaluation: Several performance metrics, including accuracy, F1 Score, recall, precision, and MSE, have been used to assess the model.

The given fig 4.1 shows the flow of the whole research work. The dataset has been loaded into Google Drive and then it is being mounted with the Google colab to directly get the data from the drive. The whole work of implementation has been done on Google Colab using Python. We also used Tableau Desktop and Microsoft Excel to create some of the visualizations supporting our work.

After mounting Google Drive into the colab, all the required libraries are loaded into colab.

The next stage is data preprocessing, where certain tasks are now done, like labeling the samples, deleting unnecessary columns, and handling any missing values. The dataset is transformed into images after completing all necessary pre-processing, and it is then divided into different training and testing samples. CNN model works on the image data which is why the dataset has been converted into images. Also, CNN can learn the spatial patterns from the images well which helps for better feature extraction. Secondly, by creating a pretrained model-based bottleneck feature extraction technique using VGG16 and VGG19, the primary second goal has been fulfilled. After comparing our suggested approach to other models including ResNet 50, Inception V3, VGG-16, and VGG-19, the XGB classifier has been used for classification following feature extraction. It has been determined via comparison that the proposed feature extractor performs the best out of all of those. Next, the RNN-CNN-based hybrid classifier has been developed. But to apply the RNN, image data needs to be converted to sequences because RNN learns patterns over time from sequential data. The output from CNN has been converted into sequential data to learn the temporal abilities of the RNN. Our suggested model performs better than existing models in terms of accuracy, F1-score, recall, precision, and mse. It has been assessed again against different existing models, includingVGG16, VGG19, ResNet50, InceptionV3, and MobileNet.

The literature done so far shows that RNN and CNN provide better results for classification. Also, before the classification using RNN-CNN, features are extracted using a bottleneck feature extractor which has been made using VGG16 and VGG19. VGG16 extracts mid-level and spatial features. On the other hand, VGG19 extracts more abstract and high-level features, which makes the feature extraction more robust and detailed as compared to other general CNN models. The bottleneck layers of the proposed feature extractor help in dimensionality reduction by reducing features to pass to the classifier for classification by avoiding overfitting. Normal CNN is used to handle spatial patterns and RNN is used to handle sequences. However, the proposed RNN-CNN model combines both spatial and sequential learning. This will help the model to classify cancers by understanding gene’s static and dynamic relationships. It might help to understand how genes evolve.

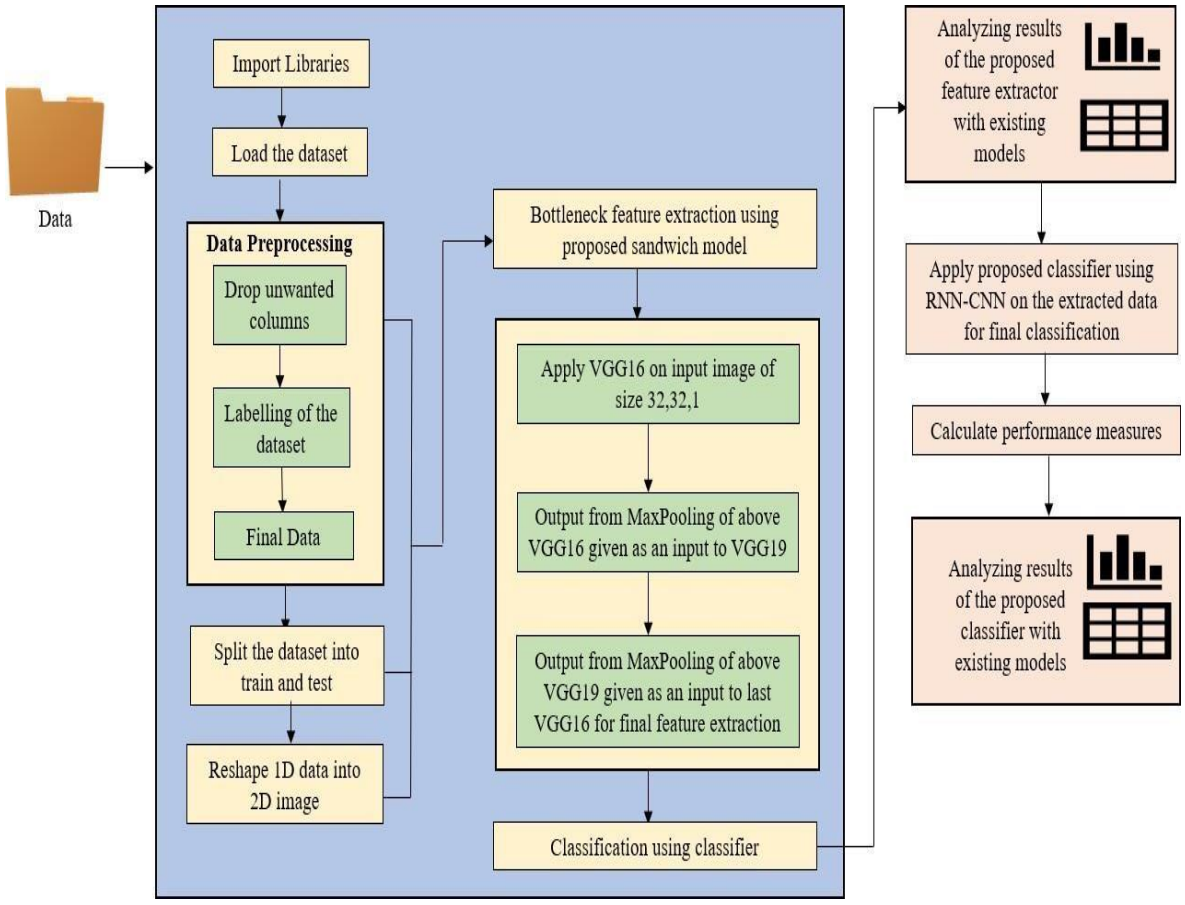

Fig 4.1 Methodology

4.3 Performance Metrics

As discussed above, the proposed models are evaluated using various performance metrics such as accuracy, F1-score, recall, precision, and mse. These measures help to find how correctly the model has been trained and evaluated on the given data. These measures use the terms true positive (TP), true negative (TN), false positive (FP), and false negative (FN). TP and TN are the correctly classified samples whereas FP and FN are the incorrectly classified samples.

4.3.1. Accuracy

It represents the total number of samples correctly classified which is represented with the help of the given equation:

$$\boldsymbol{accuracy} = \frac{\boldsymbol{TP} + \boldsymbol{TN}}{\boldsymbol{TP} + \boldsymbol{FP} + \boldsymbol{FN} + \boldsymbol{TN}}$$

Accuracy is useful to find the performance of the model. It provides a general idea about the performance, but other parameters are also required to validate this.

4.3.2. Precision

The proportion of true positive predictions to all positive predictions is known as precision. The equation is as follows:

$$\boldsymbol{Precision} = \frac{\boldsymbol{TP}}{\boldsymbol{TP} + \boldsymbol{FP}}$$

When false positives have severe consequences, precision is especially helpful. High precision, for example, indicates that the model is generating fewer incorrect diagnoses of illness present, which is important for medical applications.

4.3.3. Recall

The ratio of true positive predictions with actual positive instances in the dataset is known as recall. The equation to represent this is as follows:

$$\boldsymbol{Recall} = \frac{\boldsymbol{TP}}{\boldsymbol{TP} + \boldsymbol{FN}}$$

Recall is essential in medical applications as it shows how well the model identifies every necessary example. A strong recall is important to reduce false negatives, which means that fewer diseases go unnoticed.

4.3.4. F1 Score

When the samples are split over various categories, the F1-score is a metric that examines a classifier's accuracy by taking into consideration both the precision (PR) and recall (RC) of the prediction [152].

$$\boldsymbol{F1\ Score} = \frac{\boldsymbol{2 * Precision * Recall}}{\boldsymbol{Precision + Recall}}$$

In an imbalanced dataset like gene expression, the F1 score acts as a single measure

that captures true positives and prevents false positives.

4.3.5. MSE

MSE is the full form for mean squared error. The difference between the actual and the predicted value is known as an error [158]. MSE is represented with the given equation:

$$\boldsymbol{MSE} = \frac{\mathbf{1}}{\boldsymbol{n}} \sum(\boldsymbol{y} - \hat{y})^{\mathbf{2}}$$

The lower value of MSE shows that the predicted value is near the expected value. It is helpful to find an error.

4.4 Summary

This chapter presents the methodology that has been used to achieve the research objectives. It consists of various steps such as loading the data, data exploration, splitting data into training and testing, implementing the desired model, and then evaluating the model using various performance measures. Implementation is done in Python on Google Colab. For some visualizations, Tableau Desktop and Microsoft Excel have been used.

CHAPTER 5
DEVELOPING SUITABLE FEATURE EXTRACTION METHOD FOR DIMENSIONALITY REDUCTION

This chapter presents a concise summary of the research conducted on dimensionality reduction for gene expression datasets using various methods. Our second objective of research is to develop a feature extraction method for gene expression data. Thus, this chapter describes the proposed sandwich stacked ensemble model for bottleneck feature extraction using VGG16 and VGG19 and the results.

5.1 Introduction

Cancer is one of those diseases that are dangerous for humans as its rate of fatality ishigh. However, early-stage detection (diagnosis) and treatment increase the probability ofpatient recovery [95]. To detect cancer several tests can be performed in the lab. It requiresplenty of money, time, and resources to predict cancer in a laboratory. Computational methods are becoming complementary techniques to assess human physicians in the diagnoses of many diseases. Genetic testing has made it possible to diagnose cancer, enabling early illness identification and treatment. When it comes to gene expression, thereare typically fewer samples but higher dimensionality [67], [68]. There are thousands of genes, where every gene is a feature in gene expression data. Some genes are more responsible for cancer than others, which is why feature selection (gene selection) is required. Feature selection has proved its worth in the classification problems particularlywhen the dimensionality is high. This process is known as dimensionality reduction [69], [70].

Real-time data such as MRI scans, images, or voice recordings contain data of high dimensionality. So, to work with such type of data, dimensionality should be handled properly to get effective results. Dimensionality reduction is the process of reducing the dimensions or the features of the data. The reduced dimensions should match the

complexity level of the original data to avoid any loss of important information and to get the required information with the reduced set of features. Classification tasks become easier with dimensionality reduction as it reduces high dimensionality to low [141].

To optimize the dataset, the dimensions of the dataset have been reduced by removing the redundant and inappropriate features from the dataset. This process is known as feature extraction or feature selection. The reduced dimensionality helps in improving accuracy, reducing time and space complexity, handling noisy data, and overcoming overfitting [142]. It is not feasible to study all the gene sequences in the gene expression data because the data is highly dimensional, redundant, and sometimes noisy. So, these dimensions are required to be reduced before the classification to get the most dominating features which helps in achieving high accuracy. Dimensionality reduction has been done in two days: feature selection and feature extraction. This process removes the irrelevant features, thus reducing the training time and improving accuracy [143].

Biomedical research is getting more attention these days in fields such as cancer. Gene expression data consisting of thousands of genes, is generated through DNA microarray methods. This data carried crucial information about genes which also helps in early detection and prediction of cancer-like diseases. Despite this, gene expression data contains a higher number of features and a lesser number of observations because of redundant and unrelated genes, and noise present in the data. Due to this, these datasets suffer from overfitting and high variance. Dimensionality reduction helps to handle such type of data by reducing the number of features with the help of two methods:

- Feature Selection – Although the human body contains thousands of genes, only a smaller number of genes are related to cancer and any health issue. Finding those genes may help in diagnosing cancer well. If the dataset contains n features, most probably there will be a 2^n subset of genes. Here, gene or feature selection methods will help to choose that minimal set of genes which will provide better results such as improved accuracy and many more. The act of picking an ideal number of genes to produce the greatest amount of generalization or the lowest level of danger is known as feature selection. There are various types of feature selection algorithms

such as filter, ensemble, wrapper, hybrid, and embedded which are used for dimensionality reduction.

- Feature Extraction – The input data containing multiple features has been represented with a reduced set of features using a procedure called feature extraction. Feature extraction methods are of two types namely linear and non-linear. In linear feature extraction, matrix factorization is used to convert high-dimensional space to low-dimensional feature space to cope with linear features. In non-linear feature extraction, non-linear correlation has been found among the features to represent them in reduced feature space [144].

Cancer is one of the major causes of death around the world. The human body contains biological molecules and the relationship between those molecules is determined by omics data analysis. The types of cancer and tumor and non-tumor samples are found using omics datasets which are based on genome analysis. As the genomics dataset contains a lesser number of samples and a larger number of genes, analyzing these datasets is quite a difficult task. Data augmentation is one of the ways which generate synthetic samples to solve this issue. The other two feasible ways to handle high-dimensional data are feature selection and feature extraction. The advantage of using these two methods is to reduce the computational power and to improve the accuracy of classification. In feature selection, features are selected from the actual dataset based on some conditions. It acts as a pre-processing step in which redundant and irrelevant features are removed from the dataset. On the other hand, feature extraction is also a pre-processing step in which new features are made using the existing ones based on some conditions. New features having low dimensional space are extracted from the existing feature space [145].

The focus of this research work is to extract features with high discriminative power using a bottleneck feature extraction method based on sandwich sandwich-stacked convolutional neural network. We have stacked VGG16 and VGG19 in such a way that VGG19 is sandwiched between two VGG16 networks. The resultant network is ensembled in nature. For the classification, we have used the XGBoost model. Our model has outperformed its competitors in terms of recall, accuracy, f1 measure, and precision.

5.2 Bottleneck Feature Extraction

Even with the growth of deep learning, building a model for the classification of an image remains challenging. Deep learning requires a lot of training data to avoid the problem of overfitting. ImageNet is one of those models that has been used for the classification of images and has been trained and tested on a very large dataset. So, building such a model requires a large amount of data for training and validation, which is not an easy task. To support this, transfer learning is among the features of deep learning with the help of which a pre-trained deep learning model can be modified according to the requirements of the current problem. A model that has to be modified is pre-trained, so it is going to use the information that this model has already gained during its initial training on the large dataset, for the new classification problem. This is known as transfer- learning because the pre-trained model is going to transfer the information as per the new problem of classification.

During transfer learning, the last fully connected layer is removed from the pre-trained model, and the remaining portion of that model is used for extracting features from the dataset. The extracted features are the features till the activation layer of the pre-trained model (that is before the fully connected layer), so the features are known as bottleneck features. After this, the extracted features can then be given to any classifier for the final classification of the data as per requirements [96].

Fig 5.1 shows bottleneck feature extraction for CNN based model. There are 5 convolutional blocks and 1 fully connected block. Every convolutional block has a max pooling layer at the end in addition to the convolution layer. As per the given figure, the last block is a fully connected block that consists of flatten and dense layers. Bottleneck features are the features till the max-pooling of the convolutional block before the fully connected layer. That is why the features till the max-pooling of the fifth convolution block are known as bottleneck features. A different model can be used as the final classifier, which works on the bottleneck features, instead of the fully connected layer of the pre-trained model.

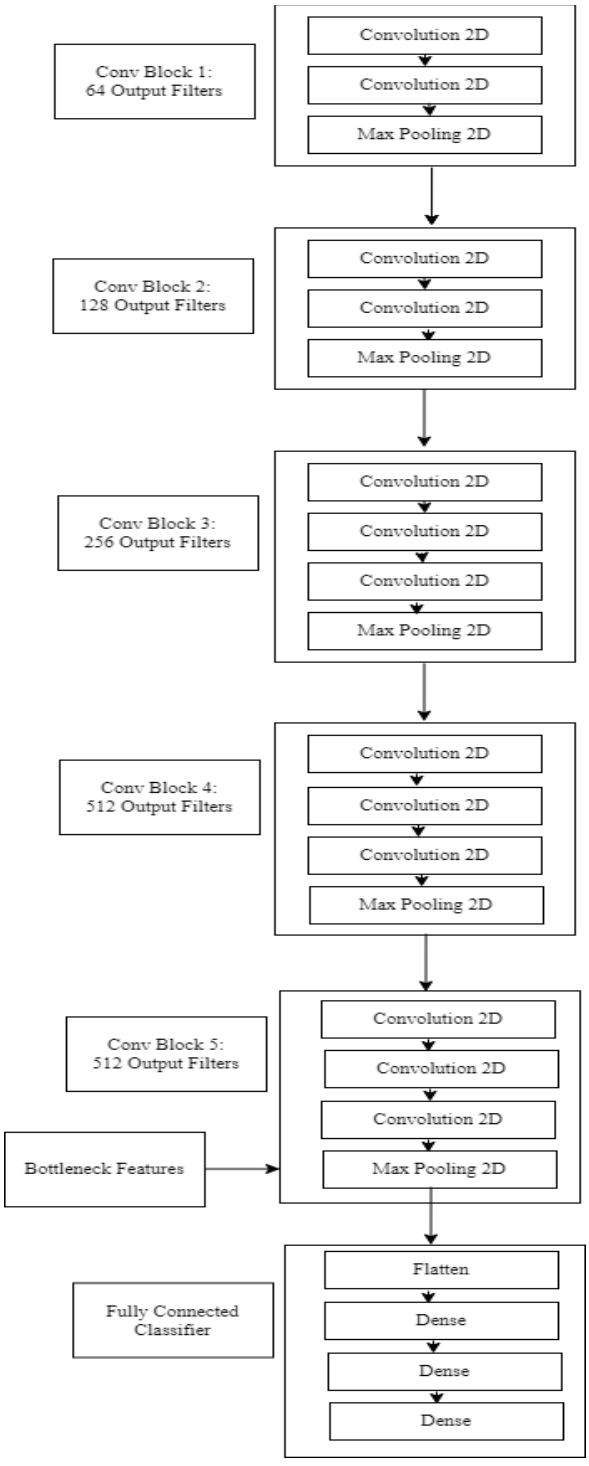

Fig 5.1 Bottleneck Feature Extraction

Deep learning models with bottleneck concepts have applications in various areas such as diagnosis of retinal disease, visual question-answering, content-based image retrieval, and learning concept-based models [97]. Bottleneck features (BN) are based on deep structures and achieved a good amount of success in Automatic Speech Recognition (ASR). However, applying BN features to small or medium-scale problems is not the right decision because of two main reasons. One is the need for a large amount of data for training the deep structure models and the other is the high inter-dimension correlation among the bottleneck features.

The bottleneck layer of the multilayer perceptron produces bottleneck features. The input layer of the MLP defines the input features and the output layer provides the corresponding labels as the output for the desired problem. The bottleneck layer in between learns from the input features after getting trained on a large amount of training data. The basic architecture of bottleneck MLP is given in Fig 5.2. It consists of 5 layers. The input layer is the bottommost layer, whereas the output layer is the uppermost one. Out of 5 layers, the remaining 3 are hidden layers. The middle layer is the bottleneck layer that generates bottleneck features. The given architecture is deep as it consists of multiple hidden layers [98].

Bottleneck features that are produced by multilayer perceptron are one of the techniques that can be used for dimensionality reduction. The dimensionality reduction technique can extract the most prominent features that can be used for different operations such as the classification of any disease. As bottleneck features are generated by MLP, one of the hidden layers in MLP is smaller as compared to other layers which helps to represent the input features in the lower dimensional space. The activation function in the bottleneck layer is a low-dimensional nonlinear function, so autoencoder can also be used as a method of dimensionality reduction. Using bottleneck features, instead of conventional features, provide more accurate results. As MLP is a conventional method, pre-trained deepneural networks act as improved MLPs. There are two main advantages of using pre- trained DNN. One is improvement in performance for all types of NN. And second is deepnetworks with multiple hidden layers provide better performance than a deep network that

uses random initialization [99].

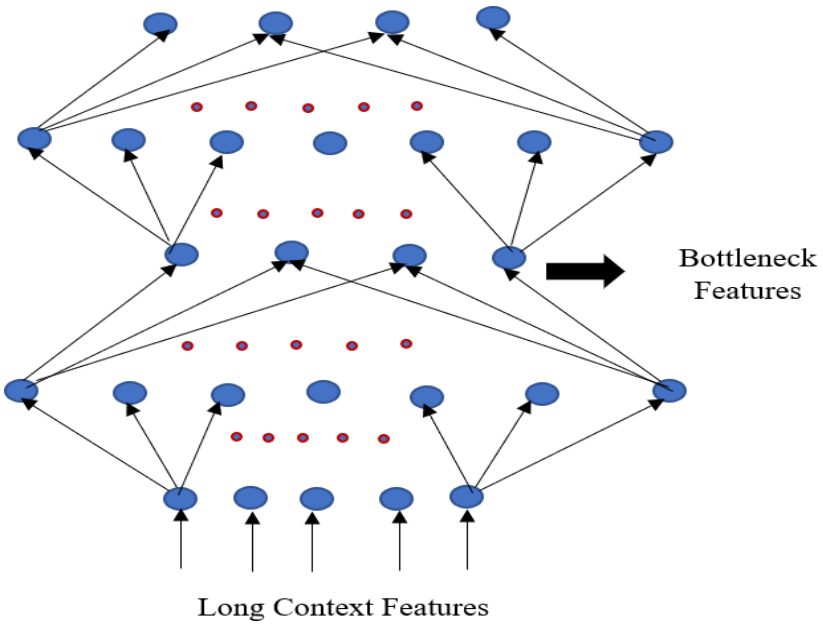

Fig 5.2 Basic MLP Based Bottleneck Feature Extraction

For the gene expression data, when a variable i.e., a gene is compressed and at the same time it carries all the information about another variable such as the cancer stage is known as the information bottleneck. This helps to compress a variable and also to find which is the most important relevant variable [100]. One of the hidden layers in DNN used to have a smaller size as compared to other layers, that layer is known as the bottleneck layer. The smaller size of this layer is used to compress the features received from the network and represent the features received from the previous layers [101].

In [102], two different feature extractors are compared: one based on CNN and the other on DNN. Images are given as input to the bottleneck network, and it provides an image as the output that is further passed to the classifier for the final classification. The bottleneck layer performs dimensionality reduction as the dimensions of the images are reduced from 4096 to 128. CNN-based feature extractor outperforms DNN-based feature extractor.

5.3 Various Pre-Trained Models for Feature Extraction

Stacked bottleneck (SBN) feature extraction is an important part of various areas such as diagnosis of a disease, ASR system, etc. In conventional SBN, there exists a hidden layer between the BN layer and the output layers. But this doesn't provide good results. So, a hidden layer has been removed between the two. There is another advantage of using a BN layer in the NN. The BN layer provides some sort of regularization that removes the need for other regularization techniques [97]. A wide range of deep learning and machine learning methods have been applied to extract features from gene expression data. Some of those methods are already discussed in the literature survey. In this section, we are going to explain the methods that we have used to get bottleneck features from gene expression.

5.3.1 VGG16

Amongst the most widely used CNN models for image classification is VGG16. [104]. It is a good method to be used for feature extraction. Convolutional layers with a predefined kernel size of 3X3 are stacked in the VGG 16 network that has been used for counting and detection of objects [105]. In terms of parameters, memory, and evaluation, it is costly. It consists of almost 138 million parameters [106]. VGG16 model has been trained on the ImageNet database which makes it a pre-trained model. Since the model has been trained on a huge set of data, it can perform well even when working with smaller datasets. It has a total of 16 convolution layers along with 5 max pooling and 3 fully connected layers as shown in Fig 5.3. While using VGG16, the default image input size is(224,224,3) [112]. But for our method, we took an image of size (32,32,1).

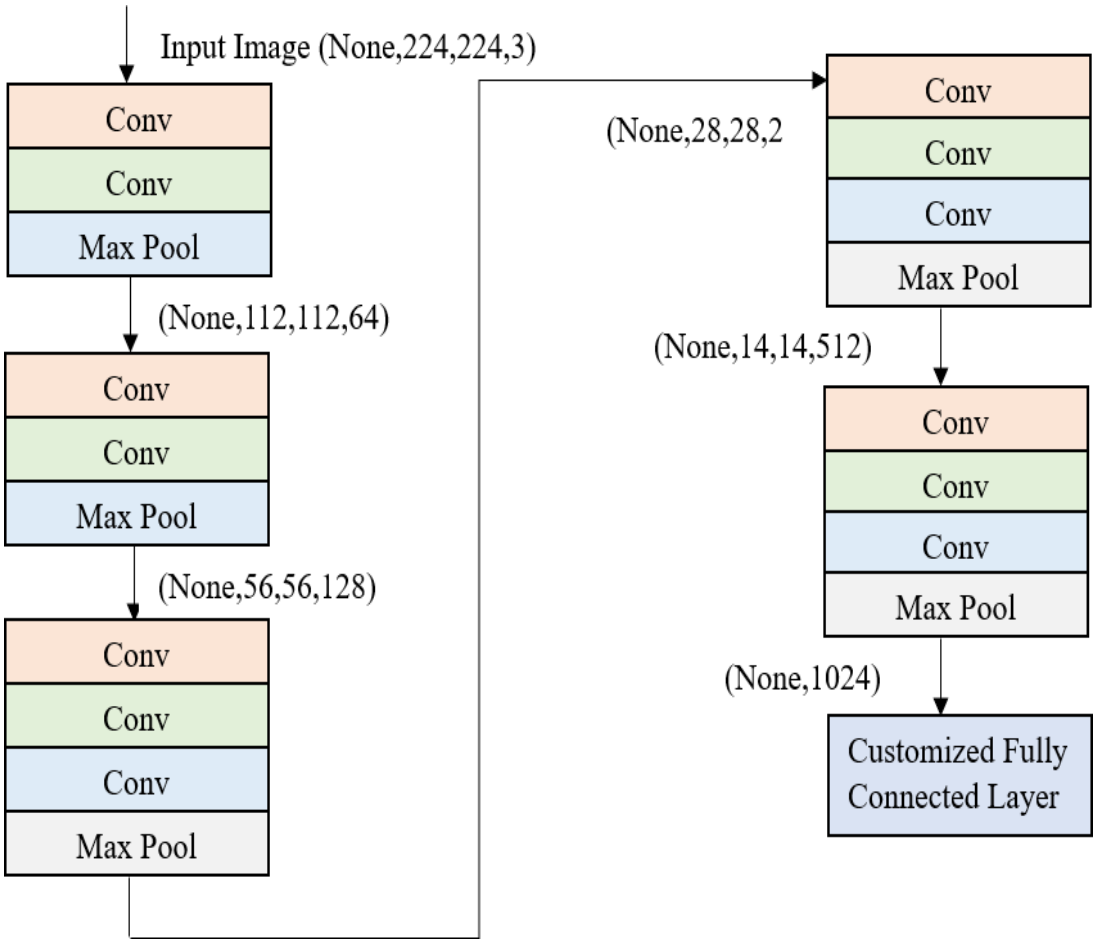

Fig 5.3 VGG-16 Architecture

5.3.2 VGG19

VGG19 is among the popular deep-learning models that are developed by "Simonyan and Zisserman". It is a popular model that takes into account the depth of relevant layers while keeping the overall number of parameters low. It contains 16 convolution layers and 3 FC layers [112]. VGG19 model has 19 layers and 144 million trainable parameters [108]. It has 16 convolutions and three fully connected layers. Additionally, it is one of the pre-trained models that has been trained on the ImageNet database of millions of images that are categorized into one thousand different classes. Fig 5.4 shows a basic architecture of the VGG19 model in which convolution layers that are 16 in number are used for feature extraction and the rest three fully connected layers are utilized for classification [113].

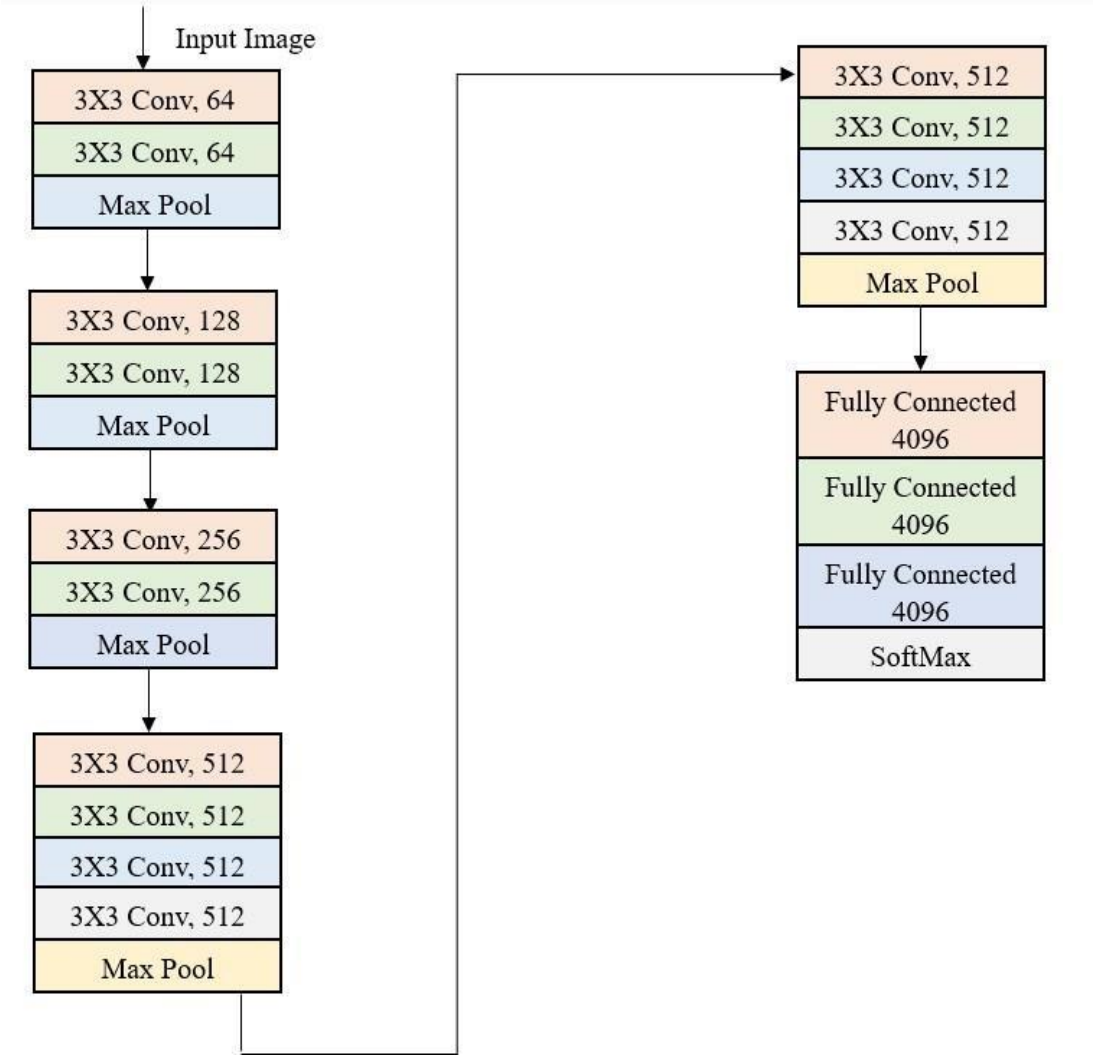

Fig 5.4 Basic Architecture of VGG-19

5.3.3 Inception V3

To increase the deepness and wideness of the NN, InceptionNet has been proposed. The main purpose of increasing depth and width was to improve the performance of the NN. One of the majorly used InceptionNet for the image classification is Inception V3. It allows the representation of high dimensions by introducing factorization into small convolutions [107].

5.3.4 ResNet 50

ResNet50 is a popular DL model for object recognition and classification with a large number of data, and it effectively sorts out the deterioration generated by the network's growing number of layers. ResNet50 provided better results for the image classification on the ImageNet dataset than other CNN Models [107]. It has a depth of 50 which makes it a deeper model than AlexNet and VGG16. It takes an image as an input having a size of 224 X 224 [110]. It is a short form of residual network which has 50 layers. The problem of vanishing gradient and overfitting has also been reduced by this model [112].

5.3.5 XGBoost Classifier

One of the ensemble learning algorithms is XGBoost. XGBoost is a boosting algorithm. Boosting consists of various models to perform the classification. The next immediate model is provided with the misclassified data along with weights as input from the current model. XGBoost provides good performance while performing binary as well as multi-classification due to its features such as tree pruning, parallel computing, and regulation [111].

5.4 Proposed Sandwich Stacked Ensemble Model

As shown in Fig. 5.5, we developed the Sandwich Stacked Convolutional Method, a feature extraction approach that extracts the bottleneck features from the dataset. We have given it the name sandwich because our method consists of 2 models arranged in a sandwich

structure.

First and last, the pre-trained model VGG16 is there, and in between two VGG16 models, the pre-trained model VGG19 is there. An image's default input size for VGG16 is (224,224,3). But for our method, we took an image of size (32,32,1). Similarly,for VGG19, we also took an image of size (32,32,1) as an input.

In Fig 5.5 (a), VGG16 model is represented. It takes an input image and then applies the convolution to it. The last layer present is Softmax which provides the output of VGG16 to the CNN1 of VGG19. Fig 5.5 (b) represents the internal structure of VGG19. It takes the output of the VGG16 as an input to its CNN1. Then at last same as VGG16, the Softmax Layer of the VGG19 provides the output of VGG19 to the CNN1 of VGG16. Fig 5.5 (c) represents the internal structure of VGG16. It takes the output from the Softmax of VGG19 as its input and then the softmax layer of this VGG16 provides the extracted bottleneck features.

In the VGG16 model, the first layer present is the input layer where we have given an image of size of (32,32,1). After the input layer, the first block consisting of 2 convolutional layers and 1 max pooling layer is there. Then a second block consisting of the same two convolutional layers and one max pooling layer is there to which input of size (16,16,128) has been given from the previous block. After this, 3 more blocks are there consisting of 3 convolutional and 1 max pooling layer. The total number of parameters received from VGG16 is 14,713,536 with 0 non-trainable parameters. Finally, to extract the bottleneck features, flatten and dense layers with an input size of 1024 are added to the model.

The extracted features from the VGG16 model are given to the VGG19. VGG19 consists of an input layer in which we provided the input image of size (32,32,1) after reshaping. This model consists of two blocks containing 2 convolutional and 1 max pooling layer just like VGG16. However, the next three VGG19 blocks are made up of one max pooling layer and four convolutional layers. The total number of parameters received from VGG19 is 20,023,232 with 0 non-trainable parameters. Finally, to extract the bottleneck features once more, flatten and dense layers with an input size of 1024 are added to the model.

Then VGG16 has been added again in the last which takes the input from the previous model i.e., VGG19. The model is made up of five blocks that include the max pooling layer and convolutional layers. It provides the extracted bottleneck features as the output. After this, these features are given as input to the classifier namely XGBoost. The dataset has been partitioned into training and testing data in the ratio of 70,30. The classifier after training and testing classifies our dataset into 5 different classes of cancer.

.

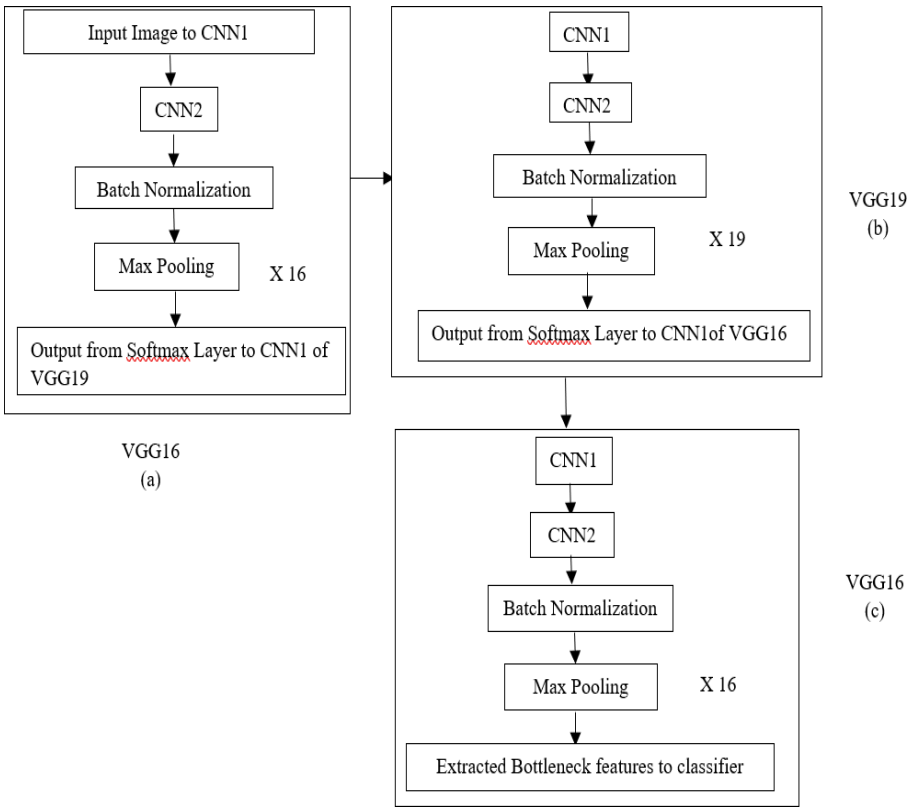

Fig 5.5 Proposed Sandwich Stacked Ensemble Method

The following equations represent the proposed model for feature extraction in which the input is given to VGG16 which extracts the features first and which is given to VGG19 for deeper extraction and then in the end to VGG16 for refinement of extracted features. After all the preprocessing steps, VGG16 is initialized first with the input for the feature extraction as follows:

$$F_{VGG16One} = VGG16(I)$$

Here I is the input from which the initial convolution layer and pooling layer will extract the

features.

$$F_{VGG16Dense} = Dense_{D1}(Flatten(F_{VGG16one}))$$

The output F_{VGG16} has been flattened into 1D and passed to the Dense layer with D_1 units. The output from this step has been passed to the convolution and pooling layer of VGG19 as per the given equation:

$$F_{VGG19} = VGG19(F_{VGG16Dense})$$

The output from the convolution layer of VGG19 is passed to the flatten layer and then to the dense layer with D_2 units as given in the equation:

$$F_{VGG19Dense} = Dense_{D2}(Flatten(F_{VGG19}))$$

The above output from the VGG19, $F_{VGG19Dense}$ has been given to the second VGG16 for further refinement of extracted features.

$$F_{VGG16Two} = VGG16(F_{VGG19Dense})$$

Then finally the output of the convolution layer of the second VGG16 has been flattened and passed to the Dense layer with D_3 units which makes the final vector of extracted features as per the given equation:

$$F_{sandiwch} = Dense_{D3}(Flatten(F_{VGG16Two}))$$

The above-extracted features are then passed to the classifier for the final classification. The whole process of working on the proposed sandwich stacked bottleneck feature extraction has been given in the form of the algorithm below as Algorithm 1.

ALGORITHM 1: BOTTLENECK FEATURE EXTRACTION FROM GENE EXPRESSION DATA

Input: image of size (32,32,1)

Output: Extracted bottleneck features

Step 1: input image of size (32,32,1) given to the input layer of VGG16

Step 2: input is then passed to the first block consisting of 2 convolutions and 1 max pooling layer

Step 3: output of the first block with size (16,16,128) passed to the second block consisting of 2 convolutions and 1 max pooling layer

Step 4: output from the previous block passed to further three blocks consisting of 3 convolutions and 1 max pooling layer

Step 5: 14,713,536 trainable parameters received from VGG16

Step 6: flatten and denser layer added at the last with size 1024 toextract bottleneck features

Step 7: Extracted features reshaped to (32,32,1) and passed to the input layer of VGG19. Then step 2 and step 3 is repeated for VGG19

Step 8: for step 4 in VGG19, there are four blocks

Step 9: 20,023,232 trainable parameters received from VGG19

Step 10: repeat step 5 for VGG19

Step 11: Extracted features reshaped to (32,32,1) and passed to input layer of VGG16 and repeat steps from 2 to 5 for final bottleneck feature extraction

5.5 Results and Discussion

Classifying different cancer kinds from gene expression data is our key research goal. But for promising classification results, we proposed a feature extractor known as a sandwich stacked bottleneck feature extractor based on VGG19 and VGG16, pre-trained deep learning models to extract promising features as discussed in Chapter 5, later on, which are passed to the classifier for final classification. We analyzed our proposed feature extractor on three different datasets available at https://data.mendeley.com/datasets/sf5n64hydt/1, https://archive.ics.uci.edu/ml/datasets/gene +expression+cancer+RNA-Seq# and https://www.kaggle.com/datasets/brunogrisci/breast-cancer-gene-expression-cumida.

As discussed above in the chapter, feature extraction has been done to reduce the

dimensionality of the dataset. There are several reasons based on which features are extracted. As per the proposed model, it contains a sandwich of VGG16 and VGG19, so the convolutional layers of these models detect patterns by generating feature maps. The upper layers detect simple patterns but as the model goes deeper, the layers detect complex patterns. These are the abstract and high-level features that are relevant for classification. Also, the features that reduce the error are kept, and the rest are ignored. During convolution when feature maps are created, only the patterns that are recognized repeatedly are considered as the most dominating features to be extracted. These features are extracted because activated neurons during the forward pass of the network as not all the neurons get activated for every input given to the network. Only the most dominating features activate the neurons. Also, the features that contribute maximum to the classification of the target class are extracted.

5.5.1 Model evaluation on Dataset 1

Dataset 1 is the dataset on which various machine learning, RNN, and CNN models are applied as shown in the previous section. For feature extraction, the dataset has been divided into several training and testing ratios, such as 60:40, 70:30, and 80:20. VGG19 has been stacked in between two VGG16 models. At first VGG16 model was applied on the input of shape (32,32,1). The extracted features from the VGG16 are again reshaped into (32,32,1) and passed to the VGG19 and then the extracted features from VGG19 are reshaped again and passed to VGG16 for final feature extraction. To evaluate the performance of our proposed model, we applied the XGB classifier to the extracted features to classify the samples as per their class.

The dataset has been split into 60%, 70%, and 80% training, and 40%, 30%, and 20% testing data respectively for evaluating the proposed model along with other deep learning-based feature extractors namely ResNet50, Inception V3, VGG16, and VGG19. The fig **5.6** and 5.7 show the extracted features for training and testing data from gene expression which are further passed to the classifier.

```
Shape of extracted training features
(625, 1024)
Extracted Training Features are
[[   6.916149    -83.96799      52.638454   ...  -34.497314
    92.09084       0.53600883]
 [ 113.412224    -79.39043      54.714806   ... -175.06496
   157.15123     -66.7282    ]
 [  50.731693    -13.030761     49.848034   ... -159.86058
   141.99644    -167.31902   ]
 ...
 [  31.03481      10.6082      116.67645    ... -155.46315
    87.79326      31.455109  ]
 [  33.184753     24.487808     62.997105   ... -112.44797
    37.298447     51.4568    ]
 [ 154.41864      38.320946     34.07643    ... -184.44772
   190.56828     -20.130013  ]]
```

Fig 5.6 Extracted features from training data

```
Shape of extracted testing features
(1461, 1024)
Extracted Testing Features are
[[  80.938484  -41.710945 -112.02898  ...  -98.63495    85.36141
    16.669548]
 [  72.520546  -34.884842   48.41642  ... -106.54831    71.45312
    68.32278 ]
 [ 115.427376  -31.28768  -130.11894  ... -254.22871   190.16025
  -160.82454 ]
 ...
 [  43.278137   21.186913  -66.29389  ... -107.78134    51.12511
    58.250492]
 [ 143.41785   -38.369022 -216.18256  ... -163.28175   253.6097
  -209.80576 ]
 [ 113.86406   101.65314    11.094513 ... -505.5117     79.020874
  -266.46527 ]]
```

Fig 5.7 Extracted features from testing data

Fig 5.8 illustrates the comparison of accuracies of various existing models in comparison to our proposed feature extractor. It has been found that the bottleneck feature extractor provides the best result among all in terms of accuracy. It provides the highest accuracy of 0.954 with 80% train data. However, with 70% of the training data, VGG19 offers the lowest accuracy of 0.77.

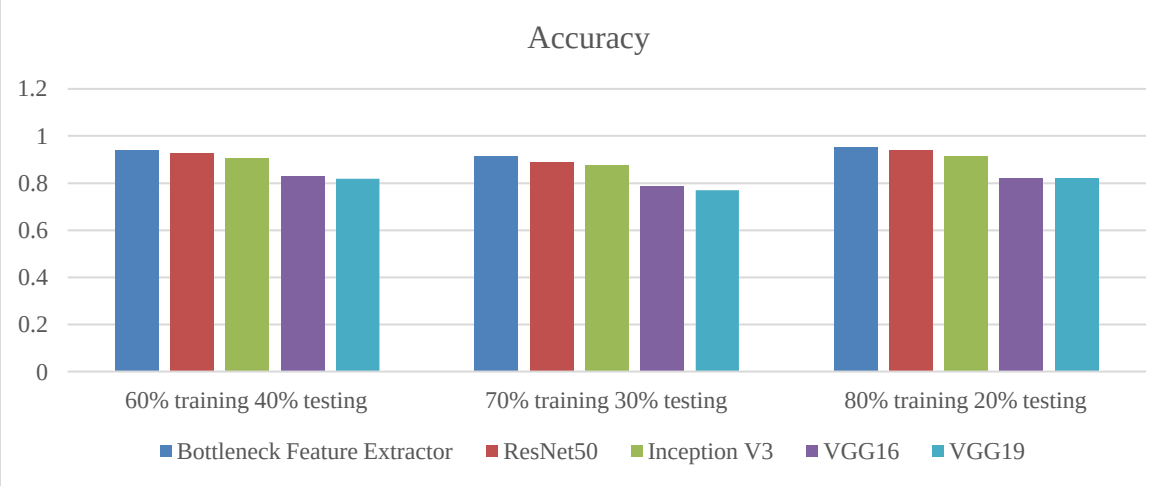

Fig 5.8 Accuracy comparison at different training and testing ratios

As seen in fig. 5.9, the bottleneck feature extractor offers the highest precision of 0.955 with 80% training data and the lowest precision of 0.915 with 70% training data. Among all the given models, VGG19 gives the lowest precision of 0.77. In terms of precision as well, bottleneck feature extractor performance is best.

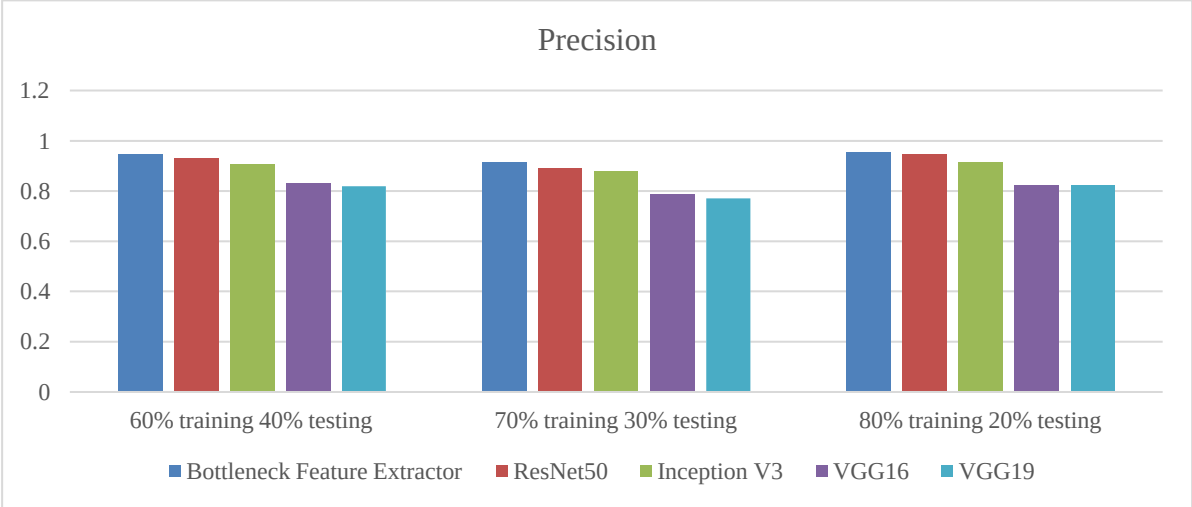

Fig 5.9 Precision comparison at different training and testing ratios

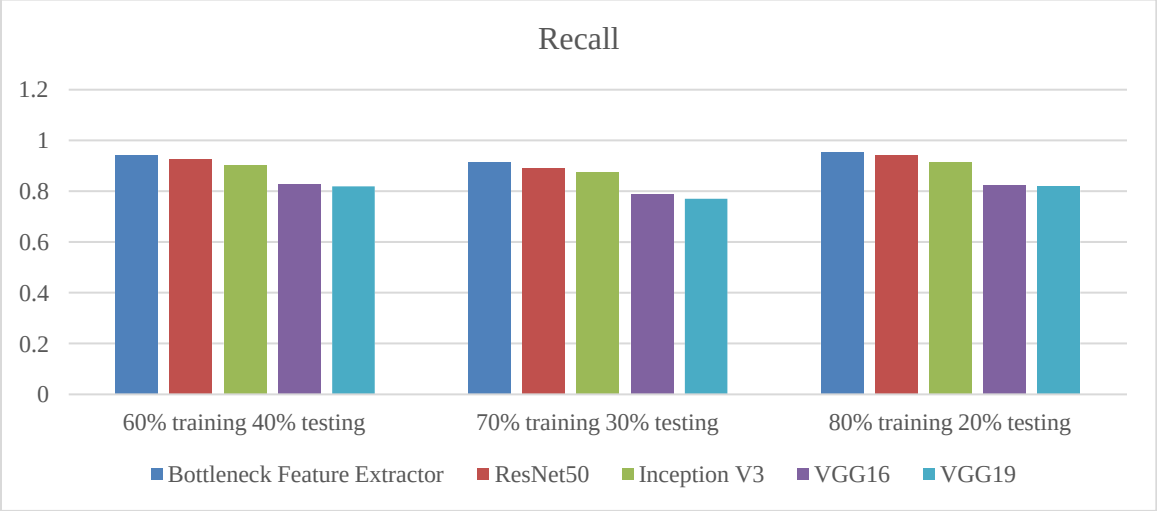

Fig 5.10 Recall comparison at different training and testing ratios

Recall and F1 score are likewise best achieved by the proposed bottleneck feature extractor, as Figs. 5.10 and 5.11 demonstrate. With 80% training data, the bottleneck feature extractor yields the maximum recall (0.955) and F1 score (0.955). The F1 score and recall performance of VGG19 are the lowest of all. It provides the lowest recall of 0.77 and the lowest F1 score of 0.757.

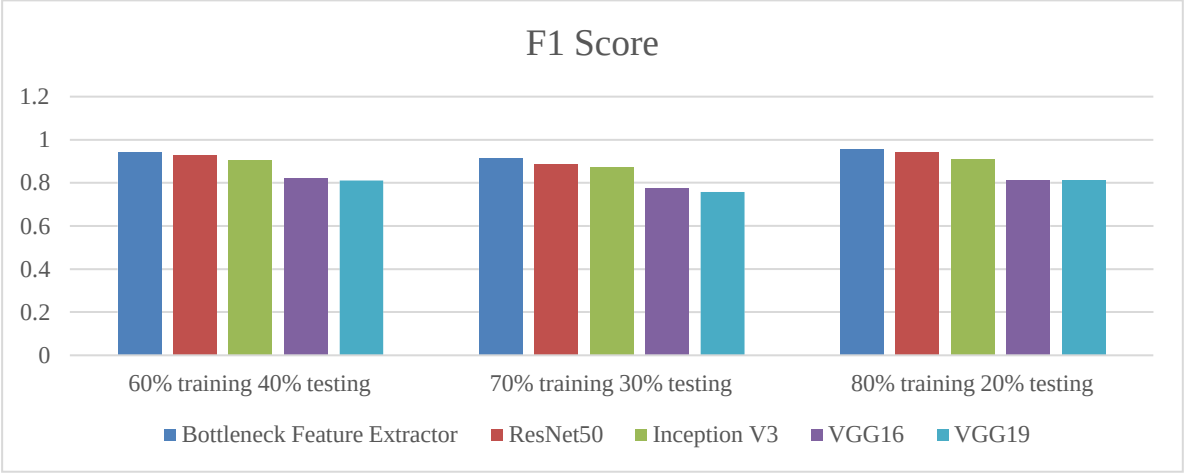

Fig 5.11 F1 Score comparison at different training and testing ratios

The suggested feature extractor provides the lowest MSE which makes our proposed model the best model among various existing models as shown in Fig 5.12. It offers the lowest MSE of 0.187 when training on 80% of the data and VGG19 comes out to be the worst performer among all the given models with an MSE of 1.127

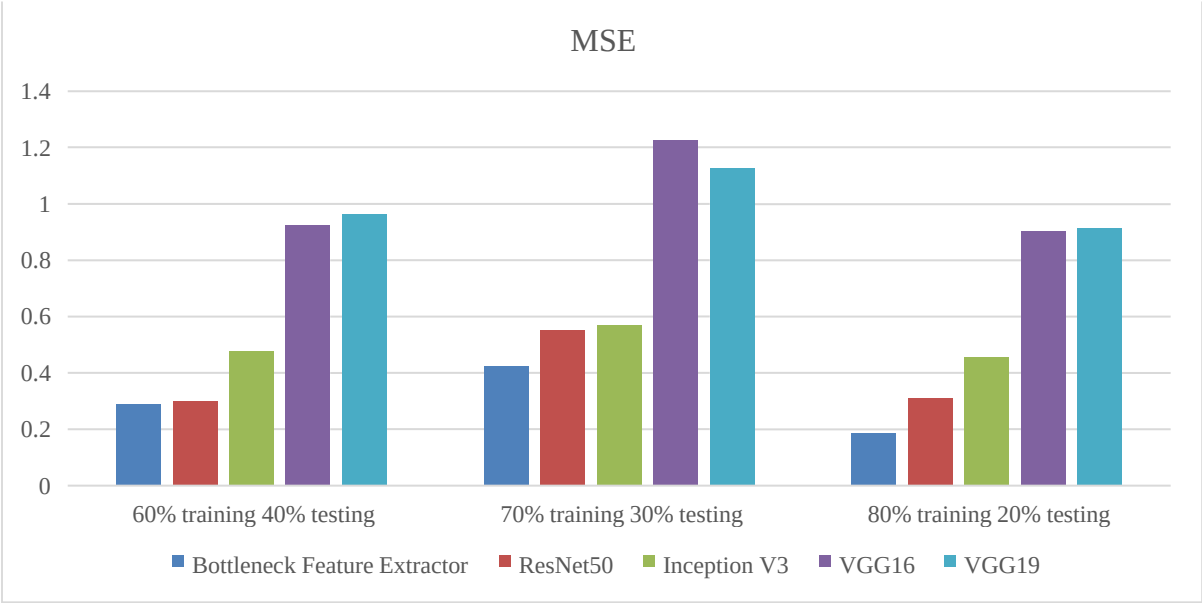

Fig 5.12 MSE comparison at different training and testing ratios

5.5.2 Model evaluation on Dataset 2

We analyzed our proposed feature extractor on another dataset which is available at https://archive.ics.uci.edu/ml/datasets/gene+expression+cancer+RNA-Seq#. The dataset consists of gene sequences of 801 samples. Each sample contains 20532 gene sequences. There are five different types of cancer: PRAD (prostate adenocarcinoma), KIRC, LUAD, COAD (colon adenocarcinoma), and BRCA. Every row has a unique sample for every patient. Similarly, the values of each gene's RNA sequence are listed in each column. 300 BRCA samples, 146 KIRC samples, 141 LUAD samples, 78 COAD samples, and 136 PRAD samples are included. There were two CSV files for this dataset, one containing features and the other containing labels. Fig 5.13 shows the first five rows for the features and Fig 5.14 shows the labels for the first five samples.

	Unnamed: 0	gene_0	gene_1	gene_2	gene_3	gene_4	gene_5	gene_6	gene_7	gene_8	...	gene_20521	gene_20522	gene_20523	gene_20524
0	sample_0	0.0	2.017209	3.265527	5.478487	10.431999	0.0	7.175175	0.591871	0.0	...	4.926711	8.210257	9.723516	7.220030
1	sample_1	0.0	0.592732	1.588421	7.586157	9.623011	0.0	6.816049	0.000000	0.0	...	4.593372	7.323865	9.740931	6.256586
2	sample_2	0.0	3.511759	4.327199	6.881787	9.870730	0.0	6.972130	0.452595	0.0	...	5.125213	8.127123	10.908640	5.401607
3	sample_3	0.0	3.663618	4.507649	6.659068	10.196184	0.0	7.843375	0.434882	0.0	...	6.076566	8.792959	10.141520	8.942805
4	sample_4	0.0	2.655741	2.821547	6.539454	9.738265	0.0	6.566967	0.360982	0.0	...	5.996032	8.891425	10.373790	7.181162

5 rows × 20532 columns

gene_20525	gene_20526	gene_20527	gene_20528	gene_20529	gene_20530
9.119813	12.003135	9.650743	8.921326	5.286759	0.0
8.381612	12.674552	10.517059	9.397854	2.094168	0.0
9.911597	9.045255	9.788359	10.090470	1.683023	0.0
9.601208	11.392682	9.694814	9.684365	3.292001	0.0
9.846910	11.922439	9.217749	9.461191	5.110372	0.0

Fig 5.13 Sample data containing features for first five samples

	Unnamed: 0	Class
0	sample_0	PRAD
1	sample_1	LUAD
2	sample_2	PRAD
3	sample_3	PRAD
4	sample_4	BRCA

Fig 5.14 Sample data containing labels for first five samples

There is an unnamed:0 column in the labels file which represents the sample number for each sample that has been dropped later from the file and then the labels file containing only the class column has been merged with the features file for further processing as shown in fig5.15.

	Unnamed: 0	gene_0	gene_1	gene_2	gene_3	gene_4	gene_5	gene_6	gene_7	gene_8	...	gene_20522	gene_20523	gene_20524	gene_20525
0	sample_0	0.0	2.017209	3.265527	5.478487	10.431999	0.0	7.175175	0.591871	0.0	...	8.210257	9.723516	7.220030	9.119813
1	sample_1	0.0	0.592732	1.588421	7.586157	9.623011	0.0	6.816049	0.000000	0.0	...	7.323865	9.740931	6.256586	8.381612
2	sample_2	0.0	3.511759	4.327199	6.881787	9.870730	0.0	6.972130	0.452595	0.0	...	8.127123	10.908640	5.401607	9.911597
3	sample_3	0.0	3.663618	4.507649	6.659068	10.196184	0.0	7.843375	0.434882	0.0	...	8.792959	10.141520	8.942805	9.601208
4	sample_4	0.0	2.655741	2.821547	6.539454	9.738265	0.0	6.566967	0.360982	0.0	...	8.891425	10.373790	7.181162	9.846910

5 rows × 20533 columns

gene_20526	gene_20527	gene_20528	gene_20529	gene_20530	Class
12.003135	9.650743	8.921326	5.286759	0.0	PRAD
12.674552	10.517059	9.397854	2.094168	0.0	LUAD
9.045255	9.788359	10.090470	1.683023	0.0	PRAD
11.392682	9.694814	9.684365	3.292001	0.0	PRAD
11.922439	9.217749	9.461191	5.110372	0.0	BRCA

Fig 5.15 Sample data with first five rows after merging features and labels file

The dataset's final column includes cancer categories, which are classified as 0, 1, 2, 3, and 4 for PRAD, LUAD, BRCA, KIRC, and COAD, in this sequence and also the unnamed:0 column has been dropped as shown in fig 5.16.

	gene_0	gene_1	gene_2	gene_3	gene_4	gene_5	gene_6	gene_7	gene_8	gene_9	...	gene_20522	gene_20523
0	0.0	2.017209	3.265527	5.478487	10.431999	0.0	7.175175	0.591871	0.0	0.0	...	8.210257	9.723516
1	0.0	0.592732	1.588421	7.586157	9.623011	0.0	6.816049	0.000000	0.0	0.0	...	7.323865	9.740931
2	0.0	3.511759	4.327199	6.881787	9.870730	0.0	6.972130	0.452595	0.0	0.0	...	8.127123	10.908640
3	0.0	3.663618	4.507649	6.659068	10.196184	0.0	7.843375	0.434882	0.0	0.0	...	8.792959	10.141520
4	0.0	2.655741	2.821547	6.539454	9.738265	0.0	6.566967	0.360982	0.0	0.0	...	8.891425	10.373790

5 rows × 20532 columns

gene_20524	gene_20525	gene_20526	gene_20527	gene_20528	gene_20529	gene_20530	Class
7.220030	9.119813	12.003135	9.650743	8.921326	5.286759	0.0	0
6.256586	8.381612	12.674552	10.517059	9.397854	2.094168	0.0	1
5.401607	9.911597	9.045255	9.788359	10.090470	1.683023	0.0	0
8.942805	9.601208	11.392682	9.694814	9.684365	3.292001	0.0	0
7.181162	9.846910	11.922439	9.217749	9.461191	5.110372	0.0	2

Fig 5.16 Sample data containing first five rows of the final dataset

Following the completion of the aforementioned procedures, the dataset is split into three training and testing ratios: 60:40, 70:30, and 80:20 but before that, the class column has been stored separately again for easy implementation. The suggested feature extractor has been executed on different splits of the dataset to extract the most prominent features. Just like our proposed method provides promising results on dataset 1, it also provides good results on dataset 2 as well. With 80% training data, the suggested bottleneck feature extractor achieves the best accuracy of 0.931 and VGG19 provided the lowest accuracy of 0.832 with the same training data as shown in Fig 5.17. The proposed model is performing best among all the given models with all the given training and testing ratios.

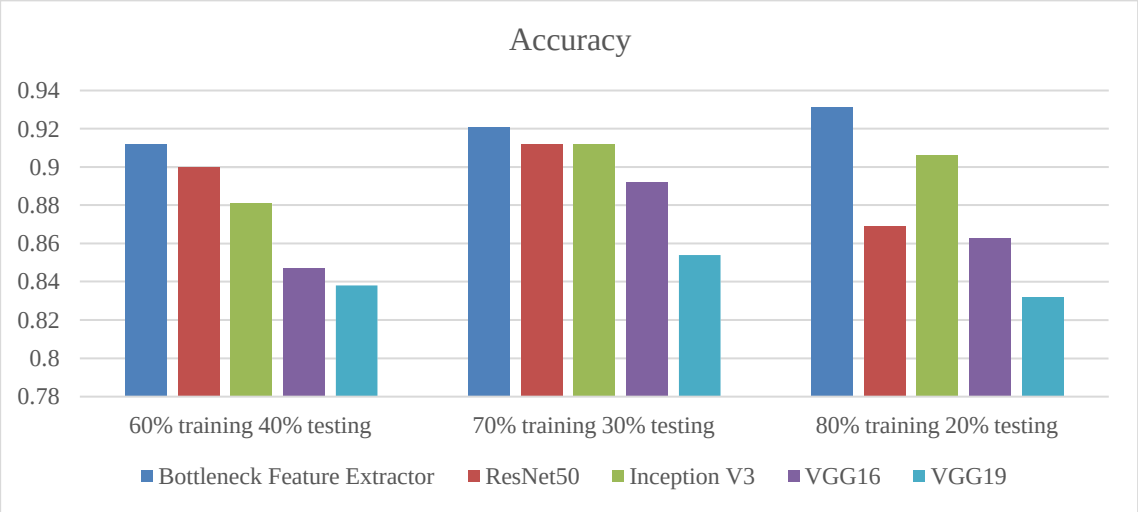

Fig 5.17 Accuracy comparison at different training and testing ratios

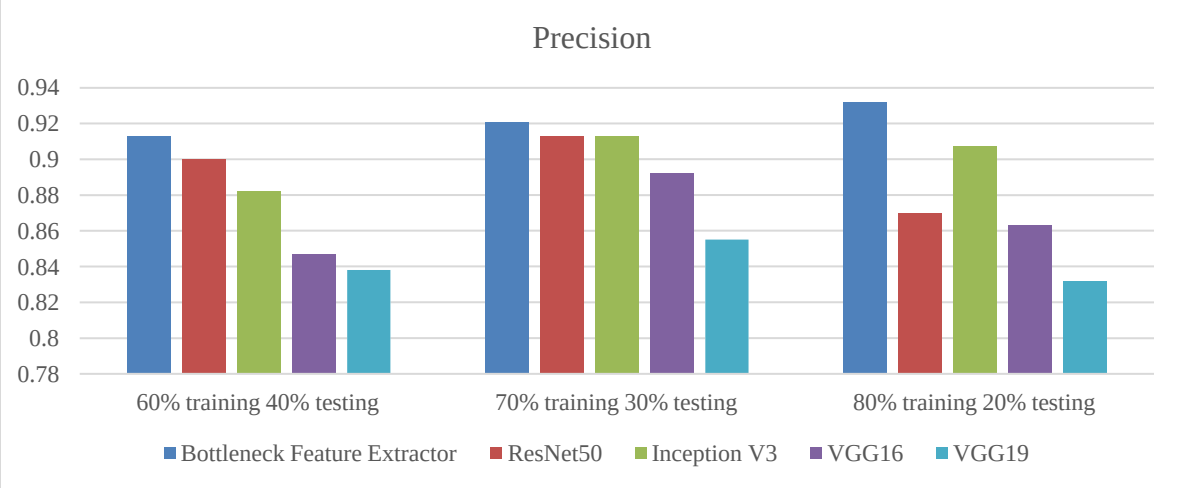

Fig 5.18 Precision comparison at different training and testing ratios

As shown in Fig 5.18, the bottleneck feature extractor performs best in terms of precision among all the given models with different training and testing data. It gives the highest precision of all the models. At 80% training data, it yields the greatest precision of 0.932; at 70% and 60% training data, it yields 0.921 and 0.913, respectively. VGG19 comes out to be the poor performer in terms of precision with a 0.832 precision value with

80% training data. Similarly, the proposed feature extractor gives the best performance interms of recall too. It gives the highest recall value of 0.932 which is the best among all the given models at various ratios of training and testing. VGG19 gives the lowest recall of 0.832 as shown in fig 5.19.

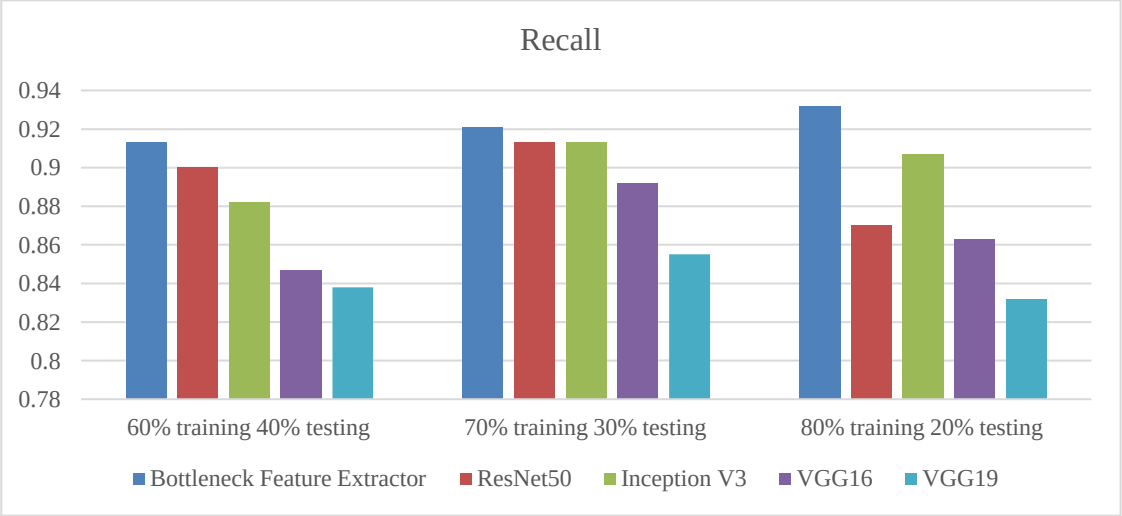

Fig 5.19 Recall comparison at different training and testing ratios

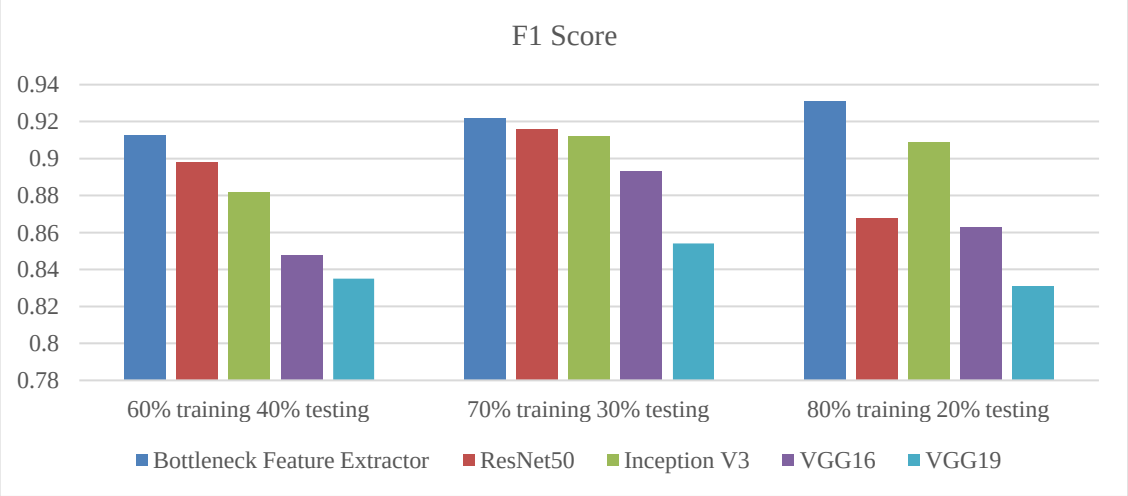

Fig 5.20 F1 Score comparison at different training and testing ratios

The F1 score and MSE of the suggested model have also been compared to those of other current models, and the results are displayed in Figures 5.20 and 5.21. Bottleneck feature

extractor comes out to be the best model for extracting features than other existing deep learning models. With 80% training data, it provides the greatest F1 score of 0.931, whereas VGG19 yields the lowest, 0.831. At the same time, VGG19 yields the maximumMSE of 0.789 using 80% training and 20% testing data. But the bottleneck feature extractorgives the minimum MSE of all the given models which is 0.186.

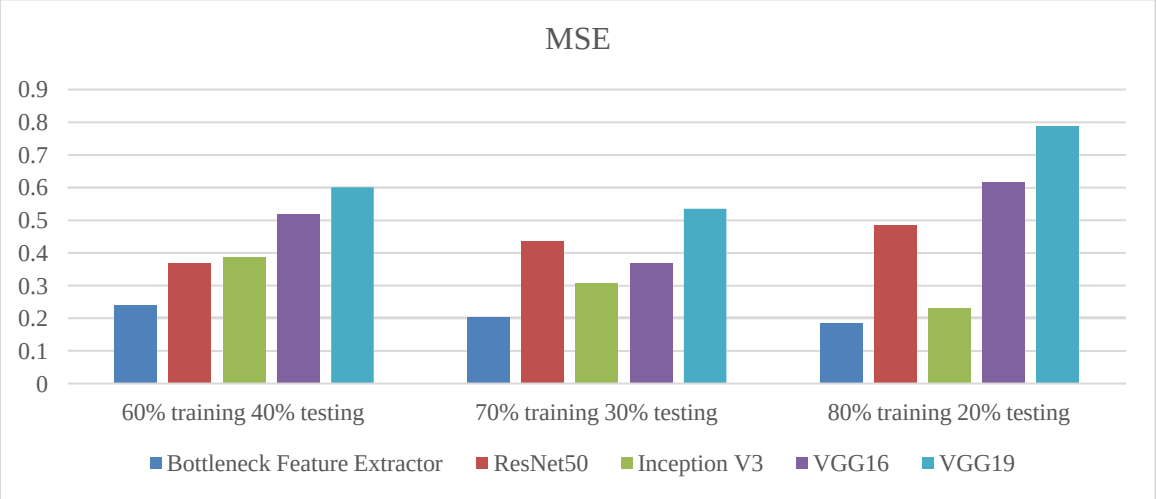

Fig 5.21 MSE comparison at different training and testing ratios

5.5.3 Model evaluation on Dataset 3

The third dataset used in our implementation is Breast_GSE45827 which is available online at https://www.kaggle.com/datasets/brunogrisci/breast-cancer-gene-expression- cumida. It consists of 151 samples and 54676 gene sequences. Basal, HER, Luminal_A, Luminal_B, Cell_Line, and Normal are the six classes in total. Every row has a unique sample for every patient. Similarly, the values of each gene's RNA sequence are listed in each column. There are 41 samples of basal, 30 of HER, 30 of luminal_B, 29of luminal_A,14 of cell_line, and 7 samples of normal. Fig 5.22 shows the screenshot of our dataset which consists of 151 rows and 54677 columns. The samples column in the dataset represents the sample number. The class of cancer is mentioned in the type column ofthe dataset. The rest of the columns represent the features i.e., gene sequences for each

sample. Late on the second column in the dataset which consists of classes has been coded with 0, 1, 2, 3, 4, and 5 for basal, HER, cell_line, normal, luminal_A, and luminal_B respectively, and then the samples column which represents the sample number has also been dropped from the dataset as shown in fig 5.23.

	samples	type	1007_s_at	1053_at	117_at	121_at	1255_g_at	1294_at	1316_at	1320_at
0	84	basal	9.850040	8.097927	6.424728	7.353027	3.029122	6.880079	4.963740	4.408328
1	85	basal	9.861357	8.212222	7.062593	7.685578	3.149468	7.542283	5.129607	4.584418
2	87	basal	10.103478	8.936137	5.735970	7.687822	3.125931	6.562369	4.813449	4.425195
3	90	basal	9.756875	7.357148	6.479183	6.986624	3.181638	7.802344	5.490982	4.567956
4	91	basal	9.408330	7.746404	6.693980	7.333426	3.169923	7.610457	5.372469	4.424426
...	...	...	...	...	...	...	...	...	...	...
146	230	luminal_B	10.392638	7.334408	6.848586	7.020486	3.228065	7.815439	5.448470	4.496955
147	233	luminal_B	10.930875	8.415294	5.906827	7.753572	3.270557	7.367931	5.906849	5.194349
148	236	luminal_B	11.027098	7.180876	6.304736	7.641197	3.206950	8.569296	5.823146	4.617309
149	237	luminal_B	10.444395	7.525153	5.964460	7.825939	3.384147	7.268454	5.245072	5.088004
150	238	luminal_B	11.345817	7.379299	5.891172	7.394586	3.183420	7.792885	5.355978	4.457914

151 rows × 54677 columns

...	AFFX-r2-Ec-bioD-3_at	AFFX-r2-Ec-bioD-5_at	AFFX-r2-P1-cre-3_at	AFFX-r2-P1-cre-5_at	AFFX-ThrX-3_at	AFFX-ThrX-5_at	AFFX-ThrX-M_at	AFFX-TrpnX-3_at	AFFX-TrpnX-5_at	AFFX-TrpnX-M_at
...	12.229711	11.852955	13.658701	13.477698	6.265781	5.016196	4.901594	2.966657	3.508495	3.301999
...	12.178531	11.809408	13.750086	13.470146	6.771853	5.291005	5.405839	2.934763	3.687666	3.064299
...	12.125108	11.725766	13.621732	13.295080	6.346952	5.171403	5.184286	2.847684	3.550597	3.158535
...	12.111235	11.719215	13.743108	13.508861	6.610284	5.193356	5.086569	3.031602	3.524981	3.272665
...	12.173642	11.861296	13.797774	13.542206	6.414354	5.040202	5.235318	2.956232	3.445501	3.193947
...	...	...	...	...	...	...	...	...	...	...
...	12.638556	12.122129	14.331152	14.133976	8.495888	4.971727	6.209136	2.852852	3.697448	3.333289
...	12.523507	11.977970	14.285405	14.070989	8.193182	6.528948	7.108210	2.929800	3.833289	3.213893
...	12.256767	11.661126	14.149586	13.977076	8.351331	6.882504	7.329545	3.085127	3.628848	3.215807
...	12.321900	11.727694	14.186277	13.943521	7.927210	6.839086	7.089259	3.018525	3.770597	3.102298
...	12.126110	11.478893	14.070188	13.857547	7.992141	5.661898	6.460331	3.061585	3.857525	3.129827

Fig 5.22 Sample dataset representing feature and labels for GSE45827

	type	1007_s_at	1053_at	117_at	121_at	1255_g_at	1294_at	1316_at	1320_at	1405_i_at
0	0	9.850040	8.097927	6.424728	7.353027	3.029122	6.880079	4.963740	4.408328	8.870780
1	0	9.861357	8.212222	7.062593	7.685578	3.149468	7.542283	5.129607	4.584418	7.767646
2	0	10.103478	8.936137	5.735970	7.687822	3.125931	6.562369	4.813449	4.425195	9.417956
3	0	9.756875	7.357148	6.479183	6.986624	3.181638	7.802344	5.490982	4.567956	9.022345
4	0	9.408330	7.746404	6.693980	7.333426	3.169923	7.610457	5.372469	4.424426	9.400056

5 rows × 54676 columns

...	AFFX-r2-Ec-bioD-3_at	AFFX-r2-Ec-bioD-5_at	AFFX-r2-P1-cre-3_at	AFFX-r2-P1-cre-5_at	AFFX-ThrX-3_at	AFFX-ThrX-5_at	AFFX-ThrX-M_at	AFFX-TrpnX-3_at	AFFX-TrpnX-5_at	AFFX-TrpnX-M_at
...	12.229711	11.852955	13.658701	13.477698	6.265781	5.016196	4.901594	2.966657	3.508495	3.301999
...	12.178531	11.809408	13.750086	13.470146	6.771853	5.291005	5.405839	2.934763	3.687666	3.064299
...	12.125108	11.725766	13.621732	13.295080	6.346952	5.171403	5.184286	2.847684	3.550597	3.158535
...	12.111235	11.719215	13.743108	13.508861	6.610284	5.193356	5.086569	3.031602	3.524981	3.272665
...	12.173642	11.861296	13.797774	13.542206	6.414354	5.040202	5.235318	2.956232	3.445501	3.193947

Fig 5.23 First 5 samples after removing samples column and coding class column with numbers

For further processing, labels from the dataset are stored separately and both the features and the labels are divided into three different ratios: 60:40, 70:30, and 80:20 for training and testing. The suggested bottleneck feature extractor has been applied to the dataset and has been evaluated using various performance metrics along with various existing models used for feature extraction. As shown in fig 5.24 and 5.25, the proposed feature extractor has performed well in terms of accuracy and precision. It gives the highest accuracy of 0.967 and the highest precision of 0.968 with 80% training and 20% testing data. On the other hand, VGG19 gives the lowest accuracy of 0.589 and the lowest precision of 0.587 with 70% training data.

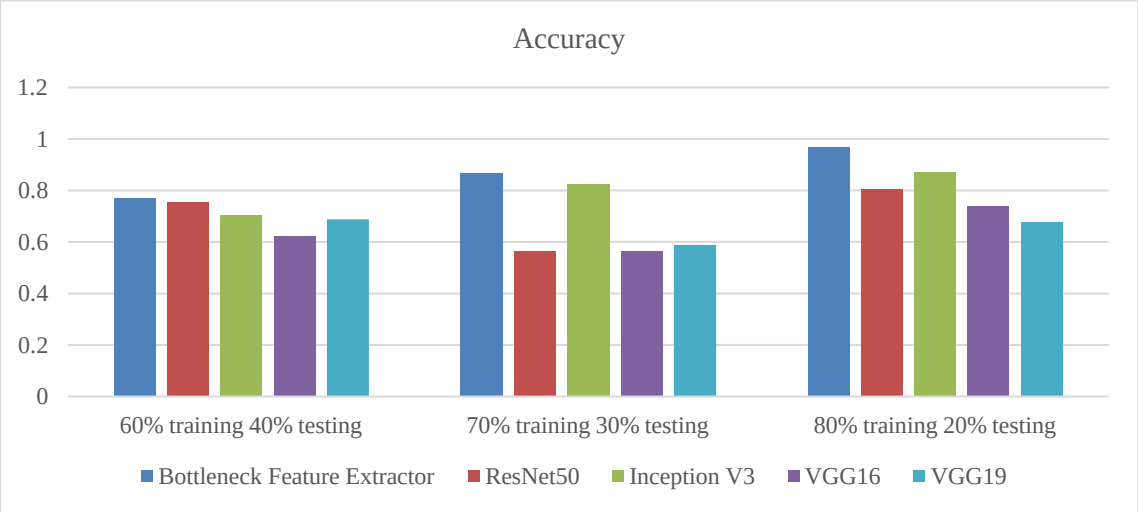

Fig 5.24 Accuracy comparison at different training and testing ratios

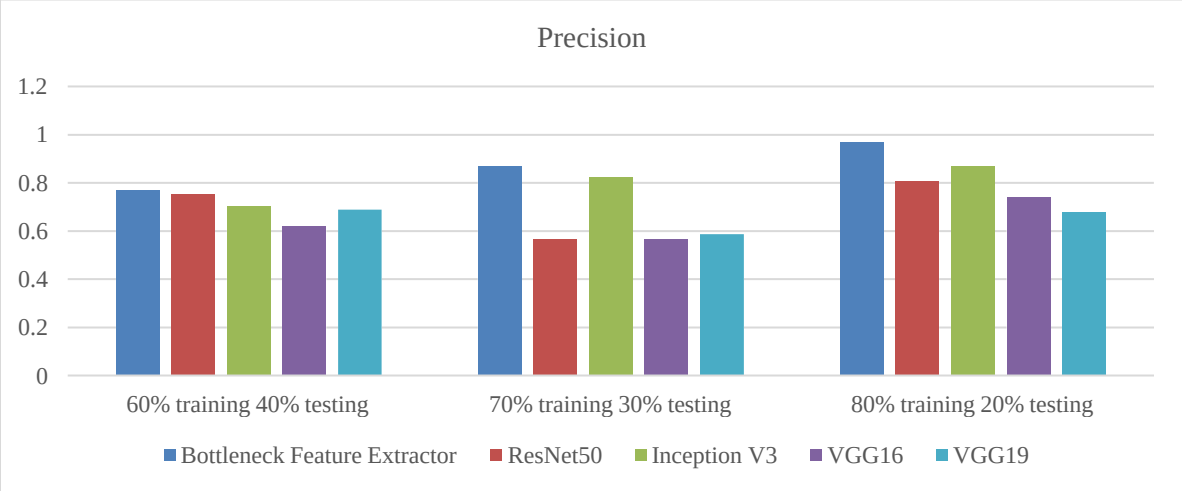

Fig 5.25 Precision comparison at different training and testing ratios

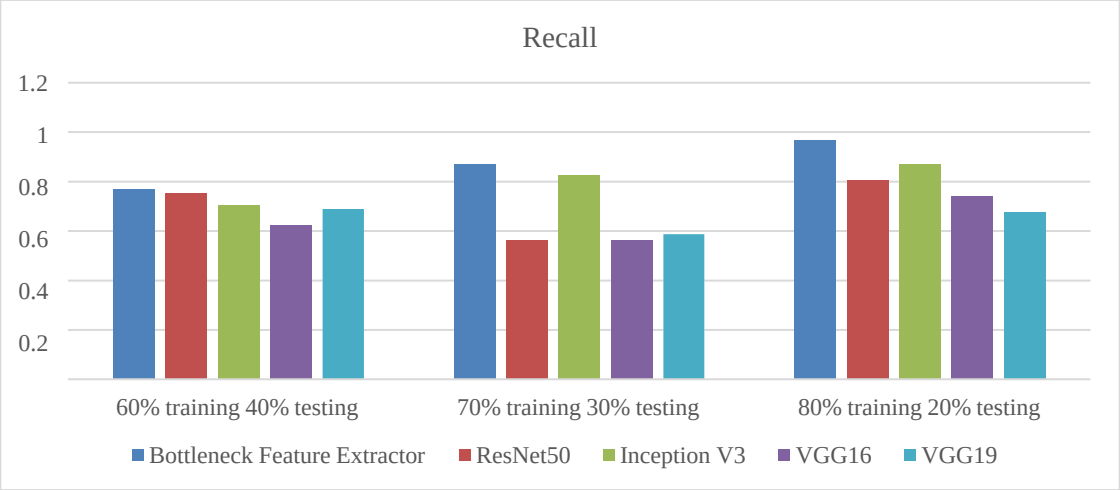

Fig 5.26 Recall comparison at different training and testing ratios

When using 80% and 60% of the training data, respectively, the suggested model yields the highest recall value of 0.968 and the lowest recall of 0.77. VGG19 gives the lowest recall among all the given models which is 0.587 with 70% training data as shown in Fig 5.26. VGG19 also comes out to be the poor performer based on the F1 score as well. It gives the lowest F1 score of all the given models which is 0.551 with 70% training data. But our proposed model again gives the highest F1 score of all, which is 0.968, and the lowest F1 score value of 0.768 at 80% and 60% of training data respectively as given in fig 5.27.

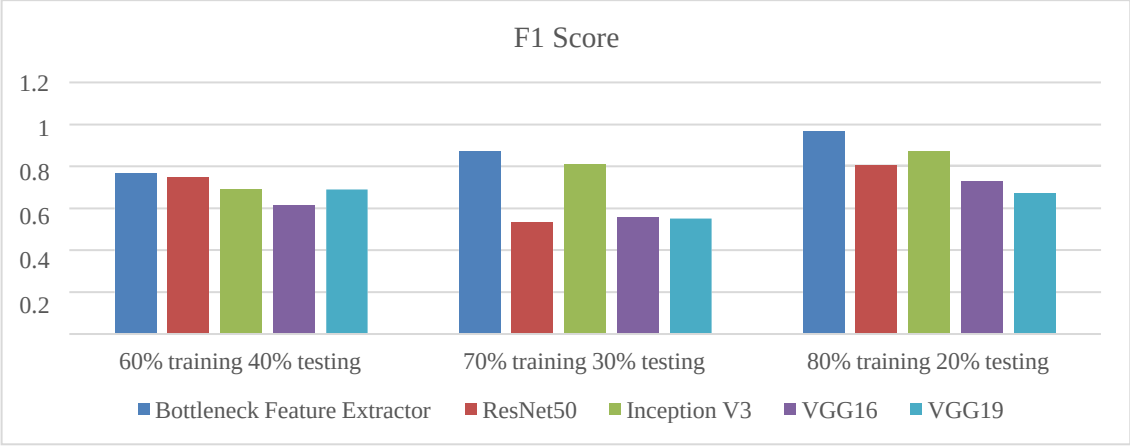

Fig 5.27 F1 Score comparison at different training and testing ratios

MSE is also one of the important performance measures which is used to evaluate a model. Out of all the models provided, the suggested bottleneck feature yields the lowest MSE, which is 0.032 for 80:20 training and testing data. VGG19 gives the highest MSE of 3.419 of all the models as shown in fig 5.28.

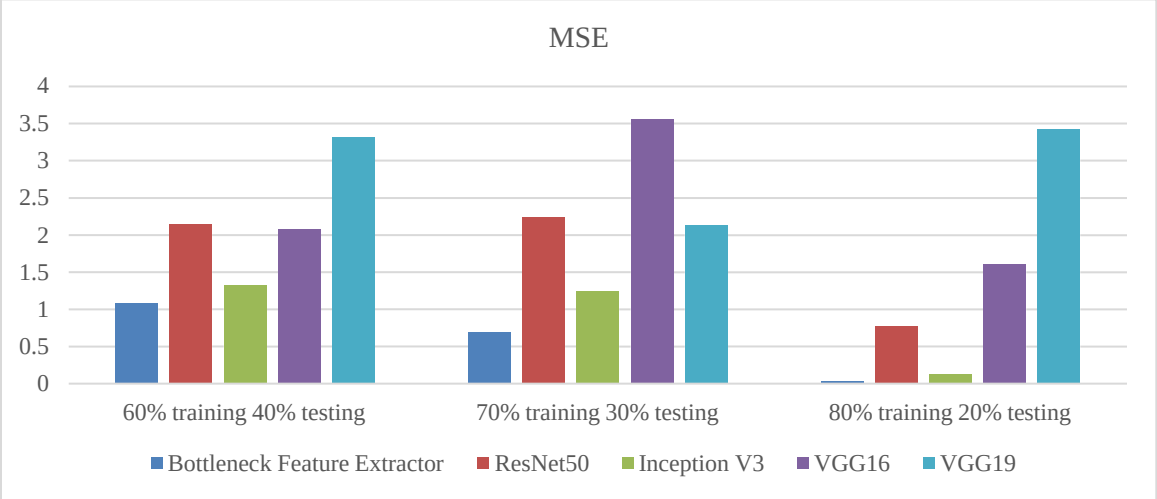

Fig 5.28 MSE comparison at different training and testing ratios

It has been found that our proposed bottleneck feature extractor has performed well among all the given deep learning-based feature extractor terms of various performance measures. We evaluated our proposed model on three different datasets to check its effectiveness and on all these three datasets it provides promising results. Even though the datasets consist of multiple classes but still our proposed method comes outto be the best.

5.6 Summary

This chapter provides various insights of our proposed model. To extract the most notable features from our dataset and increase classification accuracy, we presented a sandwich-layered bottleneck feature extraction method. We used pre-trained VGG-16 and VGG-19 models for creating a sandwich feature extractor. VGG-19 is stacked between two VGG-16 models. XGB-Classifier is then supplied with the retrieved features to perform the

classification.

CHAPTER 6
DESIGNING HYBRID MODEL FOR CANCER PREDICTION USING RNN AND CNN

This chapter provides a concise summary of the research conducted on the classification of gene expression datasets using different methods. Our third objective of research is to design a hybrid model for cancer prediction using RNN and CNN for gene expression. Consequently, the proposed hybrid approach based on RNN and CNN is explained in this chapter. Our proposed method has been compared with other classifiers to show the performance of our model.

6.1 Introduction

Cancer, an umbrella term for malignant proliferative diseases with abnormal cell proliferation, invasion, and metastasis, has been shown to represent one of the greatest threats to human health [114,115]. Epidemiologic data have shown that cancer is one of the leading causes of death in humans, second only to infectious, cardiovascular, and cerebrovascular disorders. It presents a significant risk to human health [116]. Gene expression is one of the most widely used factors in the categorization of cancer. Gene expression, or the transcriptome reflecting the actively expressed genes at any given time, can be used to determine the physiological condition and gene activity of biological systems [117]. The transcriptome is any RNA, including messenger RNA. These molecules carry genetic information from DNA, which is all the information needed to describe the properties and functions of each cell, to the ribosome [118]. RNA-Seq analyzes a gene's transcription by converting long RNAs into a library of complementary DNA (cDNA) segments that produce an expression profile. Determine which genes are essential for phenotypic specification by comparing gene expression profiles across different tissues. For example, comparing tissues from healthy and ill individuals can provide fresh insight into the genetic variables linked to pathology. Aspects of gene expression data that can be analyzed computationally can be used by researchers to find gene regularity targets,

diagnose illnesses, and develop novel medications. Studies have shown that these data can provide highly important information on the characteristics of the tumor, providing options for the patient's care, management, and therapy [119–122].

Finding genes that are highly expressed in tumor cells but not in normal ones is believed to be a difficulty that calls for the application of computational methods based on gene expression data. The high dimensionality and very small sample size of gene expression data added to the challenges associated with applying computational methods to the data. Several supervised and unsupervised learning methods have been developed for the classification of cancer-based gene expression data [122,123]. Deep learning methods address the limitations of traditional machine learning techniques when analyzing gene expression data for cancer [124].

6.2 Cancer Classification

In a literature survey, we reviewed various cancer classification techniques based on machine learning and deep learning. Classification is a supervised learning algorithm where a machine is provided with both features and labels. It is a process in which a machine is provided with training data consisting of features and labels and then unseen data for testing purposes. The association between input and output data is being targeted by classification [127]. It is a supervised method that is used to predict a sample’s class [130]. In this section, a brief review of classification techniques is given that we have used in our work to compare with our proposed classifier. Our proposed classifier is CNN and RNN-based. It has been contrasted with other deep learning and machine learning classifiers, including support vector classifiers, VGG16, VGG19, ResNet50, Inception V3, decision trees, gradient boosting, gaussian naïve Bayes, K nearest neighbor, and MobileNet. Our suggested approach performs better than other cutting-edge techniques.

6.2.1 Decision Tree

The decision tree is one of the classifiers that uses decision functions to classify the unknown samples into a class. The decision tree consists of terminal and non-terminal nodes. Non-terminal nodes are root nodes and interior nodes, and terminal nodes are the

nodes that provide the final classification as shown in Fig 6.1 [125]. DT has various roles in various areas such as character recognition, speech recognition, expert systems, medical diagnosis, and many more [126]. DT converts complex problems into simpler ones and provides easier and more understandable solutions. An attribute's test condition is represented by the interior node, the attribute's output is represented by the branch, and a class label is represented by the leaf node [127].

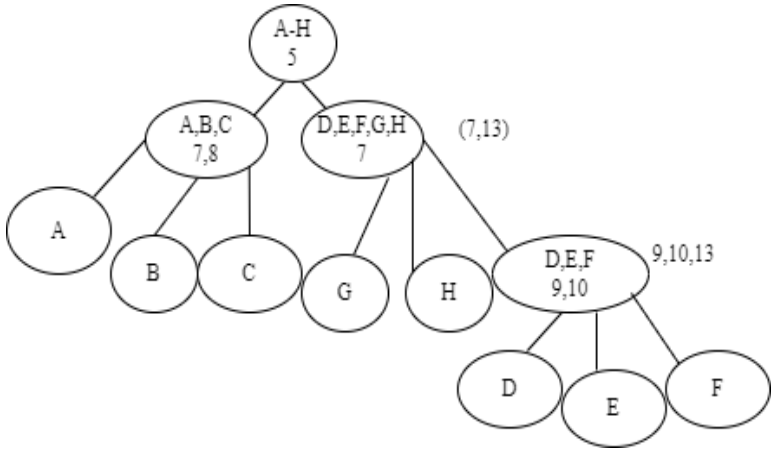

Fig 6.1 Decision Tree

6.2.2 Gradient Boosting

Ensemble classifiers are the type of classifiers in which multiple classifiers are combined to get more accuracy for predictions. Gradient boosting (GB) is an ensemble classifier that has provided outstanding performance when the count of features is more than the samples i.e., the high dimensional data [128]. GB is one of the useful models of ML which is used when there is less training data, and less training time [129].

6.2.3 Gaussian Naïve Bayes

The Bayes theorem of probability forms the basis of the naïve Bayes classifier. Each variable is considered as an independent variable in NB. It can work well even with a small amount of data [130]. It is an efficient and useful classifier that has its use in different areas such as categorization of text, spam-ham filtering, and data stream classification. This model has having fast training and testing process. Gaussian NB is the case of NB in which

the attributes that are given class labels are assumed to have a Gaussian distribution. Gaussian NB works well for estimating the distribution of continuous data [131]. It assigns a label of a class which increases the posterior probability of each sample [132].

6.2.4 K Nearest Neighbor

The Nearest Neighbor is a classifier that assigns a class to the unlabeled sample based on the class of the nearest neighbor. The nearest neighbor is calculated using a distance function such as the Euclidean distance formula. K-Nearest Neighbor is a type of nearest neighbor where k is several nearest neighbors [133]. It stores the whole set of training data. The majority class among the K-nearest samples in the training data is used for the classification of unlabeled data [134]. It is a non-parametric classification method. The performance of KNN classification depends on how fine the nearest neighbors are determined and determining the nearest neighbors depends on how fine the dataset has been pruned [135]. It can even be useful when there is little or no information about data distribution. KNN is a simple, robust, and effective model [137].

6.2.5 Support Vector Classifier

The samples are classified into two or more classes using the support vector classifier (SVC). However, it cannot detect the samples or outliers that do not belong to any of the given classes. Different types of kernels such as linear and non-linear are there for SVC. The samples near the boundary that help in classification are known as the support vectors [136]. For the same amount of data, SVC has a higher convergence rate and produces good results with fewer support vectors than other state-of-the-art techniques. SVC,which is based on SVM, is used in the fields of text categorization, face identification, numerical pattern recognition, and protein fold recognition [138].

6.2.6 Mobile Net

Mobile Net is used for object detection and classification. It is one of the pre-trained models on the ImageNet dataset with tensorflow. Pretrained models have the advantage of giving high accuracy with a small amount of data as compared to traditional neural networks

[139]. The model's accuracy is maintained despite the use of a lightweight, simplified neural network that lowers the number of detection parameters. The depth multiplier and resolution multiplier are its two hyperparameters. Deeply separable convolution is used by MobileNet, and there isn't a pooling layer between the layers of depth-wise separable convolution [140].

6.2.7 Concurrent Neural Network

A kind of neural network that has been utilized for image and video recognition is the convolution neural network. CNN is also used in other domains, like natural language processing and drug development, among others [16]. As seen in Fig. 6.2, CNN is made up of several layers, including an activation function, multiple convolution layers, sub-sampling layers, and a fully connected layer.

This given model is known as LeNet-5 and has been coined by LeCun et. al [59]. Features are extracted from the data using convolution layers that are present through the input-to-output layer in the network. Then a fully connected layer is there that has been used for classification. Between each convolution layer, sub-sampling or the pooling layer has been inserted. CNN model takes the 2D image as an input. In CNN, not all the neurons ineach layer are connected to every other neuron in the next layer. It depends on the featuremap that has been received from the previous layer. As the number of connections is less in CNN unlike other neural networks, it reduces the training time as well as overfitting. The weights and bias values of each neuron in the filter are retained, and it is connected tothe same number of neurons as in the preceding layer. This boosts the learning process and minimizes the requirements of memory. This suggests that each filter's neurons are trying to detect the same pattern from various regions of the input image. On the other hand, the size of the network has been controlled by sub-sampling or pooling layers of theCNN [60].

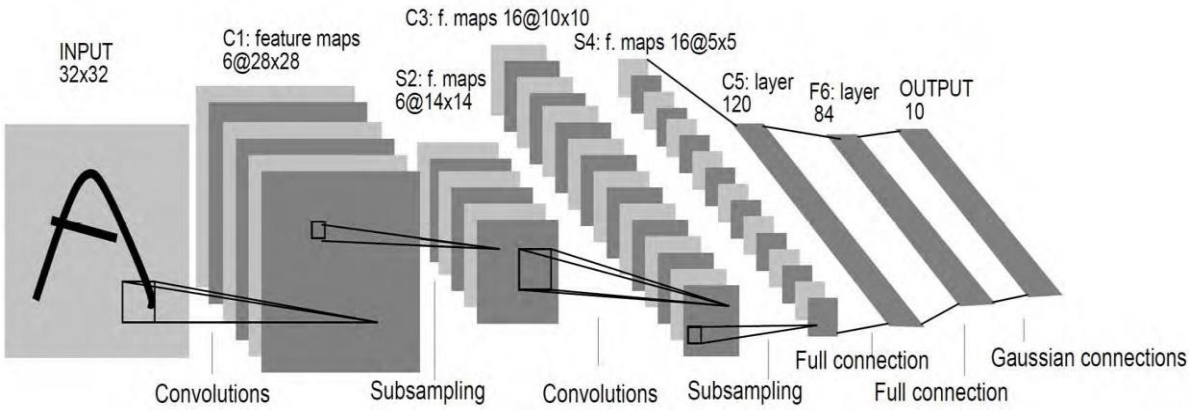

Fig 6.2 7-Layer CNN for Character Recognition [59]

In fully connected layers, all neurons are connected to every other neuron and this layer performs the final classification. Deep CNN can be implemented using many series of weight-sharing convolution layers and sub-sampling layers. These provide better representations along with controlling the other parameters such as locality, and invariance to the input image [61]. There are various versions and implementations of the CNN namely:

- AlexNet
- Inception
- ResNet
- VGG
- DCGAN [16]

6.2.8 Recurrent Neural Network

Recurrent Neural Network was a primary topic of research and development in the 1990s. Sequential and time-varying patterns are learned by RNN. RNNs are neural networks that have closed-loop feedback. There are various areas where RNN has been used such as predicting head tracking using virtual reality systems, estimating the power of wind turbines, financial prediction, forecasting of electrical load, etc. [62]. The human brain is one of the strongest recurrent structures that helps humans to learn, act, and perceive in any situation. RNN is a neural network that has computing power like a brain. There is feedback in RNN that guides RNN to perform accordingly. The size of the RNN is smaller

as compared to the feed-forward network that provides the same accuracy. RNNs can be categorized into two types: globally recurrent networks and locally recurrent networks. Both the categories can be used for dynamic systems. Globally recurrent network suffers from a problem of large training time and has complicated structures. On the other hand, a locally recurrent network has a less complicated system and less training time.

RNNs can also be divided into two categories: simultaneous recurrent networks and time-delayed recurrent networks. Over time, the time-delayed recurrent network offers accurate prediction. Nevertheless, a strong function approximator is provided by simultaneous recurrent networks [61]. The RNN has two different types of architectures such as fully interconnected and partially connected. In a fully connected recurrent network, each node takes input from every other node, and each node can also have feedback from itself as shown in Fig 6.3.

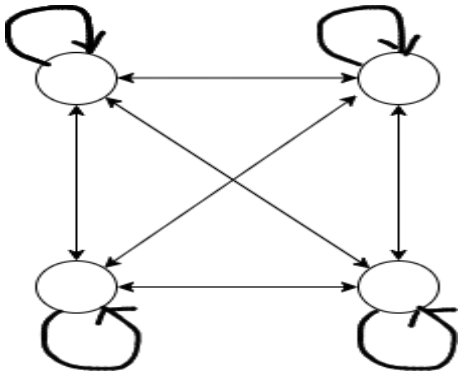

Fig 6.3 Fully Connected Recurrent Neural Network

Fig 6.4 shows the structure of a simple partially connected recurrent network. In this architecture, some nodes act as a feedforward network and other nodes have sequential access and get feedback from various other nodes of the network. The units of the second layer provide time-delayed feedback to the weights from C1 and C2, which are processed via backpropagation for the inputs [62].

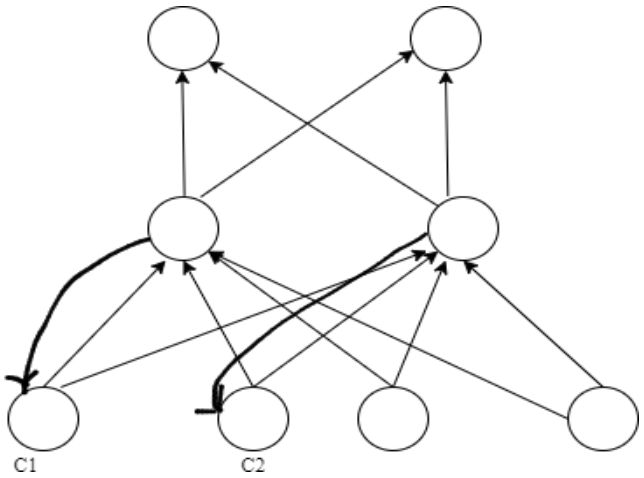

Fig 6.4 Simple Recurrent Neural Network

6.3 Proposed RNN-CNN based Classifier for Gene Expression

We proposed a classifier that is based on CNN and RNN to classify various types of cancer for gene expression data. Gene expression dataset has been taken and it has been preprocessed as well. After preprocessing, features were extracted using the proposed feature extractor and to perform the final classification, the proposed classifier was then applied to the extracted features.

The basic structure of the proposed classifier is shown in Fig 6.5. The proposed technique, feature extractor is used to extract the bottleneck features, as demonstrated in the previous chapter. The feature extractor has been made using pre-trained VGG19 and VGG16 joined in the form of a sandwich. VGG19 is placed as a sandwich layer between two VGG16 models. The combination of VGG16 and VGG19 is used for multi-scale feature extraction. VGG16 has fewer layers as compared to VGG19, which in turn performs low to mid-level feature extraction. On the other hand, VGG19 is deeper so it performs high-level feature extraction. This combination acts as a pre-preprocessing step for the hybrid RNN-CNN which performs final classification. The feature extractor performs multi-scale feature extraction, assuming none of the gene patterns are going to be missed. The proposed feature extractor performs dimensionality reduction which passes the most dominating features to reduce the problem of overfitting during classification. This extracts the spatial features

which are further passed to a hybrid of CNN and RNN.

There are three 2D convolution layers in the proposed model. An activation layer comprised of a relu activation function, a batch normalization layer, and a dropout layer connects to each 2D convolution layer. Each dropout layer has a value of 0.25. The dropout layer is added to ignore the contribution of some of the neurons in the model for the next layer to prevent our model from overfitting during training. An image of size (32,2,2) has been provided as an input to the first 2D convolution layer. Then the output from the 1st convolution block consisting of activation, normalization, and dropout is provided to the input layer of the second 2D convolution layer with a size of (64,2,2). The activation function has been used to add non-linearity to the output. To normalize the data, batch normalization is there in the proposed model. It is also used to increase the training speed. The outcome from this is then sent as an input of size (128,2,2) to the third convolution block from the dropout layer of the second convolution block.

After this, the RNN and LSTM layer has been added to the network to work on the sequences. The output of these layers was then passed to the dense layer with SoftMax acting as the activation function, followed by the flatten layer. To send the output from the preceding levels to the dense layer, the flatten layers turn it into a one-dimensional format. Because every neuron in this layer is connected to every other neuron in the layer before it, it is known as the dense layer. It uses the softmax function as our data has multiple classes i.e., 5 classes of cancer. This layer performs the final classification for our given data. The proposed model is trained with an Adam optimizer. The role of the optimizer is to change the weights and learning rate to decrease the loss during the training process. The model has been tested on the extracted features for 59 epochs to get the desired performance measures.

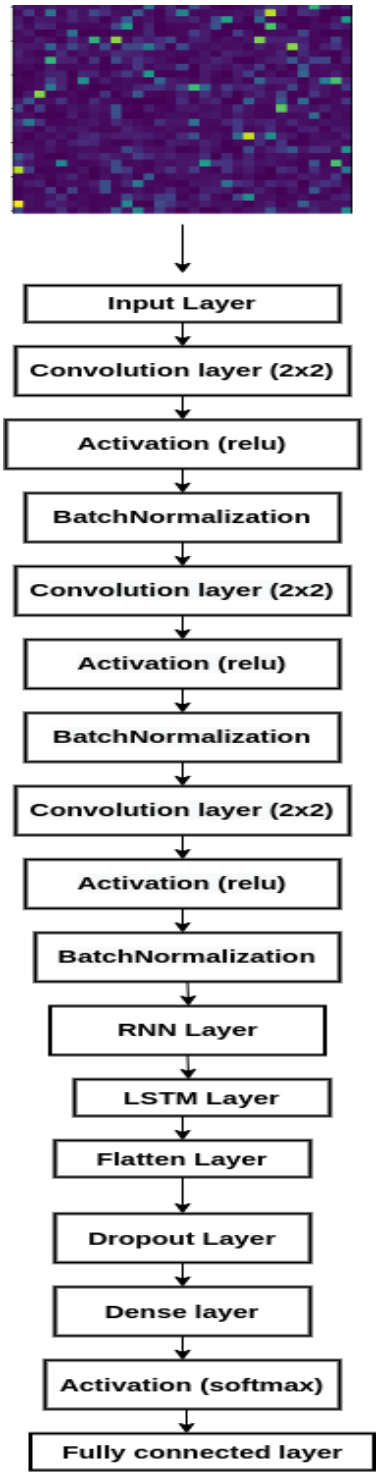

Fig 6.5 Proposed RNN-CNN Classifier for Gene Expression

The whole process of working is given below in the form of the algorithm as Algorithm 2:

ALGORITHM 2: CANCER PREDICTION USING RNN AND CNN FROM GENE EXPRESSION

Input: Extracted features

Output: Cancer classes

Step 1: extracted training and testing features from the proposed feature extractor reshaped into the size of (224,224,3)

Step 2: input supplied to the first block included a dropout layer of 0.25, a layer of batch normalization, a relu activation, a convolution layer with 32 filters, and a kernel size of (2,2)

Step 3: moved to the second block, which had a dropout layer of 0.25,a layer of batch normalization, a relu activation, a convolution layer with 64 filters, and a kernel size of (2,2)

Step 4: then to the third block, which has a dropout layer of 0.25, a layer of batch normalization, a relu activation, a convolution layer with 128 filters, and a kernel size of (2,2)

Step 5: Output from the third block reshaped into 3D which represents the batch size, time step for every sequence, and features

Step 6: reshaped 3D input is given to a simple RNN containing 128 units with return sequences as true

Step 7: output from simple RNN given to LSTM containing 128 units and return sequences as true

Step 8: output from LSTM is flattened and a dropout of 0.25 added

Step 9: for final cancer classification, the flattened output is sent to the dense layer with softmax activation

The above-proposed model consists of various components as discussed, such as the input layer, convolutional layer, reshaping, recurrent layers, flatten layer, and dense layer. All these layers have been also shown mathematically with the help of the given equations:
The extracted features from the feature extractor developed in the previous chapter, are given to the model in the form of an image of size 224 X 224 X 3 which means height, width, and RGB channel as shown in the equation,

$$\mathrm{I} \in R^{224*224*3}$$

After that CNN model was applied which consists of several layers of convolution, ReLU, batch normalization, and dropout layer as shown in the equation,

$$O^i = F^i * Ip^{(i-1)} + bi^i$$

Where,

O^i is the output of the ith convolution layer

F^i is the filter or kernel of convolution

$*$ is the convolution operation

$Ip^{(i-1)}$ represents the input to the layer

bi^i represents biasness

The ReLU activation has been applied to add non-linearity as follows,

$$\mathrm{RA}=\max(0, O^i)$$

This represents the output with 32 filters and 2 stride

The output received from the ReLU activation has been normalized using batch normalization as per the given equation,

$$\mathrm{BN(RA)}=\frac{\mathrm{RA}-\mu}{\sigma}$$

Where,

μ and σ represents the mean and variance of RA

The dropout layer has been added after batch normalization which drops a few neurons having some probability as given below,

$$Ó=\text{Dropout}(BN(RA), p)$$

Where,

Ó is the final output which will be given to the next convolution layer

$BN(RA)$ is the batch normalized output which has been derived from the ReLU function

p is the probability such as 0.25

The above-mentioned output Ó has been given to the next convolution block which contains convolution and ReLU which produces output containing 64 filters having 2 strides which has been then passed to the batch normalization and dropout layer which generates the output the same way as discussed earlier. The output for this block is represented as Ú. This output is further given to the third convolution block which produces Ć as the final output containing 128 filters. This output has been reshaped from a 3D tensor to a 2D matrix for the RNN block as given,

$$Í=Reshape(Ć) \in R^{n3*(n3*128)}$$

The above input has been given to RNN and has been represented with the help of an equation as follows,

$$HS_1 = \sigma(Wt_r Í + WtH_r HS_0 + bi_r)$$

In the above equation, where values are used which are represented as:

Wt_r and WtH_r is the weight of the input state and hidden state

HS_0 is the initial hidden state

bi_r is the biasness

σ is the activation function

The output from the RNN block has been given to the LSTM layer or block which mainly consists of forget gate, input gate, and output gate as follows:

$$fg_t = \sigma(Wt_f Í + WtH_f HS_f + bi_f) \quad \text{(forget gate)}$$

$$ig_t = \sigma(Wt_i Í + WtH_i HS_i + bi_i) \quad \text{(input gate)}$$

$$Cd_t = \tanh(Wt_c Í + WtH_c HS_c + bi_c) \quad \text{(candidate memory)}$$

$$Ċ_t = fg_t \odot Ċ_{t-1} + ig_t \odot Cd_t \quad \text{(cell state)}$$

$$og_t = \sigma(Wt_o\acute{I} + WtH_oHS_o + bi_o) \quad \text{(output state)}$$

$$HS_t = og_t \odot \tanh(\dot{C}_t) \quad \text{(hidden state)}$$

The output received from the LSTM block has been represented as $HS_t \in R^{n3*128}$ which has been given to the flatten layer to convert it into a 1D vector as follows:

$$O_{flat} = Flatten(HS_t) \in R^{n3*128}$$

This output has been passed to the fully connected layer for final classification depending on the number of classes.

6.4 Hyperparameters

Hyperparameters are the configurations that are set before the model training to improve the performance of the model by reducing overfitting and underfitting, optimizing the learning process, efficient training process, and making the model adaptable to new datasets. As per our implementation, below are given the hyperparameters that have been set manually to increase the performance of our proposed model.

Table 6.1 List of Hyperparameters

CNN Layers	
Hyperparameter	Value
No. of Filters	32, 64 and 128 filters
Size of filter	2X2
Activation Function	ReLU
Dropout Rate	0.25
Normalization	Batch Normalization
RNN Layers	
Simple RNN	One Simple RNN having 128 units and return_sequences = TRUE

LSTM Layer	LSTM having 128 units and return_sequences = TRUE
Dropout Rate	0.25
Fully Connected Layer	
Dense Layer	No of neurons equal to number of classes
Activation Function	Softmax
Training Parameters	
Epochs	20
Optimizer	Adam Optimizer
Loss Function	Poisson Loss
Callbacks	ReduceLROnPlateau (reduce learning rate by 0.5 if validation loss has not been increased till 7 epochs) EarlyStopping (stop training if the loss doesn't improve for 20 epochs)

As the number of filters kept on increasing, it contributed to capturing more and more complex features from the dataset. The filter size of 2X2 helps to capture every localized pattern which helps in higher performance. ReLU adds non-linearity to the model so that it can learn non-linear relationships among the data. The dropout layer prevents overfitting and batch normalization helps to make the model generalizable to the new data. On the other hand, simple RNN and LSTM handle sequential and long-term sequential modeling respectively. The softmax activation is useful for performing multi-class classification. In the code, increasing the number of epochs helps to increase the accuracy. Adam optimizer is used to set the learning rates properly.

6.5 Results and Discussion

After extracting prominent features from the data of gene expression, the next and final step is to classify cancer into different classes as per the datasets. We proposed a novel deep learning-based classifier using RNN and CNN. The suggested model includes layers for activation, batch normalization, and dropout in addition to three 2D convolutional blocks and a single layer of Simple RNN and LSTM along with dropout, flatten, and dense layer as discussed above. The proposed classifier is more impactful as compared to the existing machine learning models such as DT, NB, gradient boosting, KNN, and SVC in terms of handling gene expression data which is highly dimensional. These methods perform manual feature engineering and fail to identify sequential or spatial information from the data. However, our proposed method uses CNN models to extract the features and the RNN model to learn the temporal information from the data. Similarly proposed classifier outperforms existing deep learning models such as VGG16, VGG19, Inception V3, ResNet50, and MobileNet. These deep learning models can work with spatial data, but these models fail to handle sequential modeling. However, the proposed RNN-CNN can handle both spatial and sequential data very well.

We trained and evaluated both proposed models at different training and testing sizes such as 60:40, 70:30, and 80:20. It helps to make a model more generalized towards unseen data and to check for model overfitting and underfitting. It makes the model robust and not only dependent on the trained data. For different training and testing sizes, there will be different performance metrics that will make the model more effective. It also makes hyperparameter tuning effective. Our proposed classifier has been evaluated on three different datasets as discussed in the previous section. Fig 6.6 shows the predicted classes using the proposed classifier for the testing data.

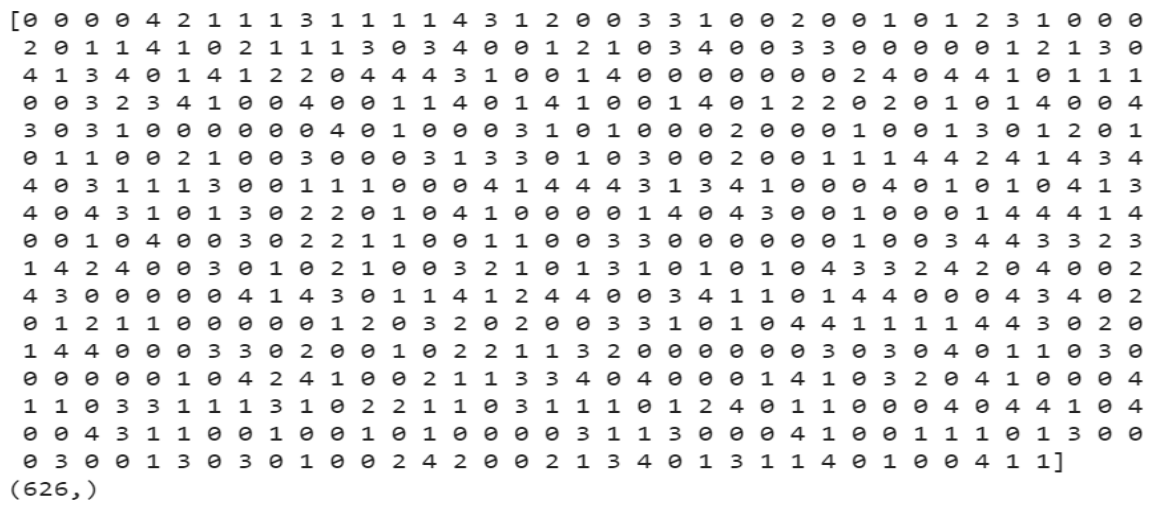

Fig 6.6 Predicted classes of cancer after classification

6.5.1 Model evaluation on Dataset 1

To demonstrate the efficacy of our suggested classifier, comparisons with both machine learning and deep learning models have been made. The dataset has been divided into different training and testing ratios of 60:40, 70:30, and 80:20 following the feature extraction techniques discussed, similarly, the same extracted data has been used for classification purposes. The various machine learning models that we have used for comparison are Decision Tree, Support Vector Classifier, Gaussian Naïve Bayes, Gradient Boosting, and K Nearest Neighbor. On the contrary, we have compared the deep learning models VGG16, VGG19, ResNet50, Inception V3, and MobileNet. Figure 6.7 illustrates that out of all the provided machine learning models, the RNN-CNN classifier that has been proposed has the best accuracy (0.995), while the Gaussian naïve Bayes model has the lowest accuracy (0.772). This clearly shows that our model is the best based on accuracy as compared to existing ML models such as decision tree, gaussian naïve Bayes,gradient boosting, K neighbor, and support vector classifier.

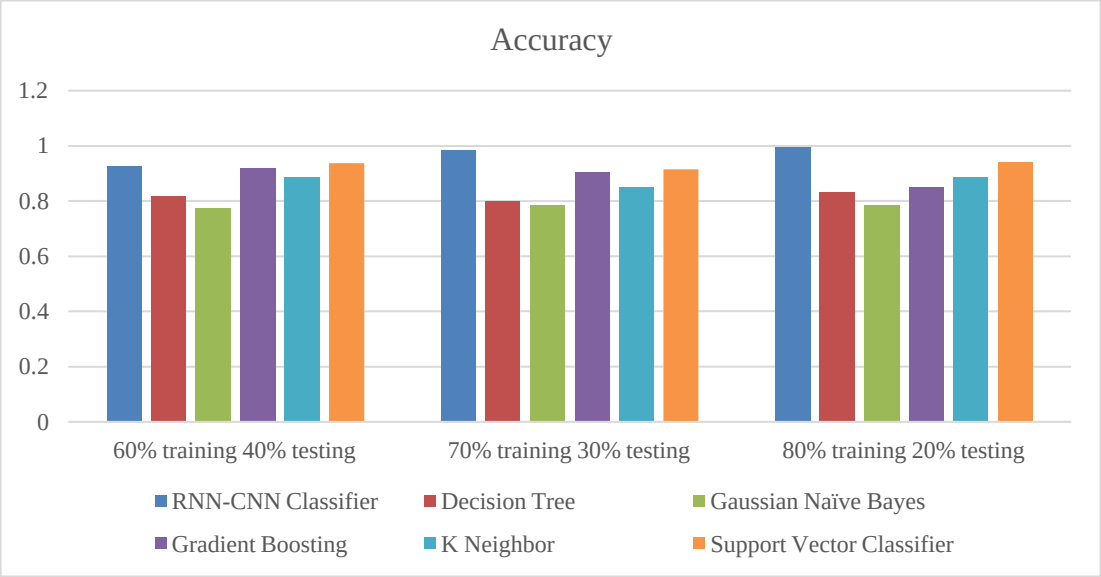

Fig 6.7 Accuracy comparison with ML Models at different training and testing ratios

The suggested model has also been compared with various DL models as mentioned above and comes out to be the best again. Both VGG16 and VGG19 do not perform well in the case of classification with an accuracy of 0.414 and 0.105 respectively which is the bottom two accuracies of the given models as given in fig 6.8.

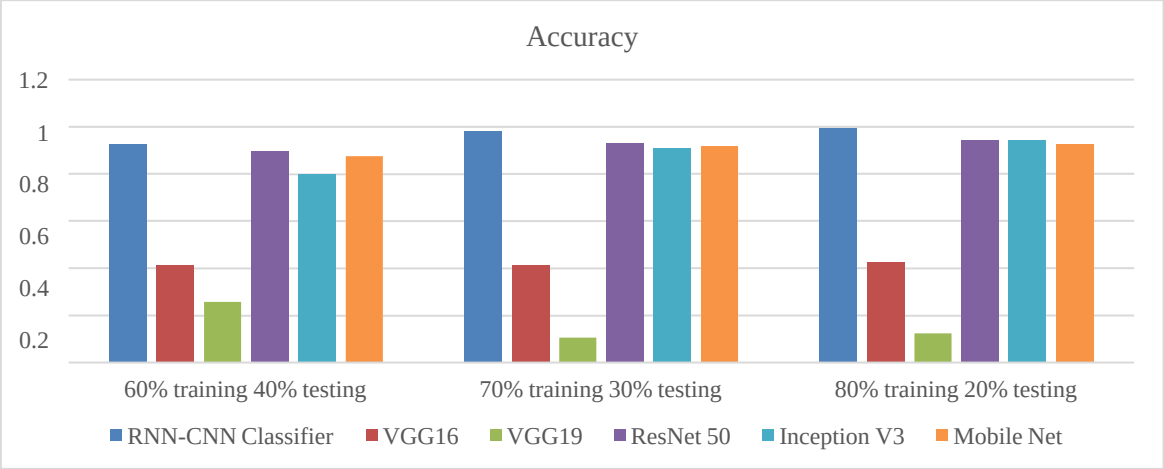

Fig 6.8 Accuracy comparison with DL Models at different training and testing ratios

The RNN-CNN has been compared with various ML and DL models in terms of precision as well, as shown in fig 6.9 and 6.10. It provides the highest precision of 0.995. In

ML models, gaussian naïve Bayes does not perform well and provides the lowest precision of 0.772. Both VGG16 and VGG19 have not performed well in the case of DL models. VGG16 provides 0.414 precision value and VGG19 provides 0.105 as the lowest precision.

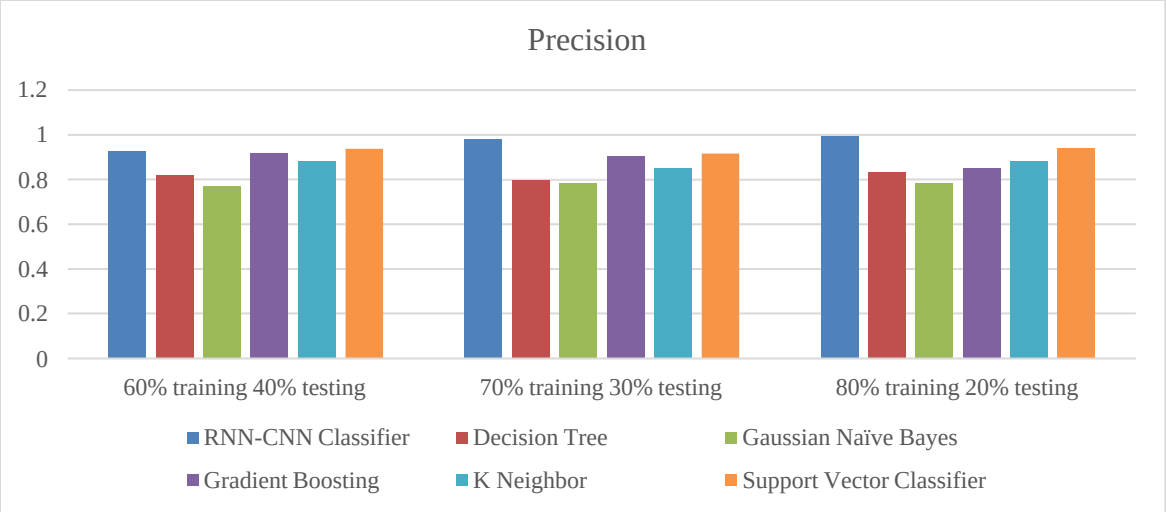

Fig 6.9 Precision comparison with ML Models at different training and testing ratios

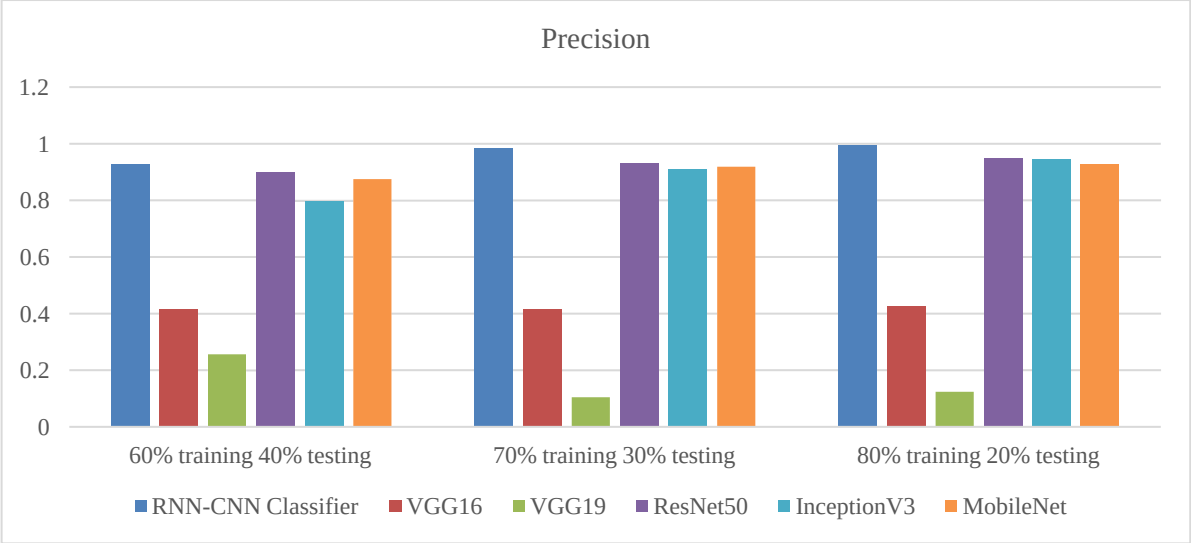

Fig 6.10 Precision comparison with DL Models at different training and testing ratios

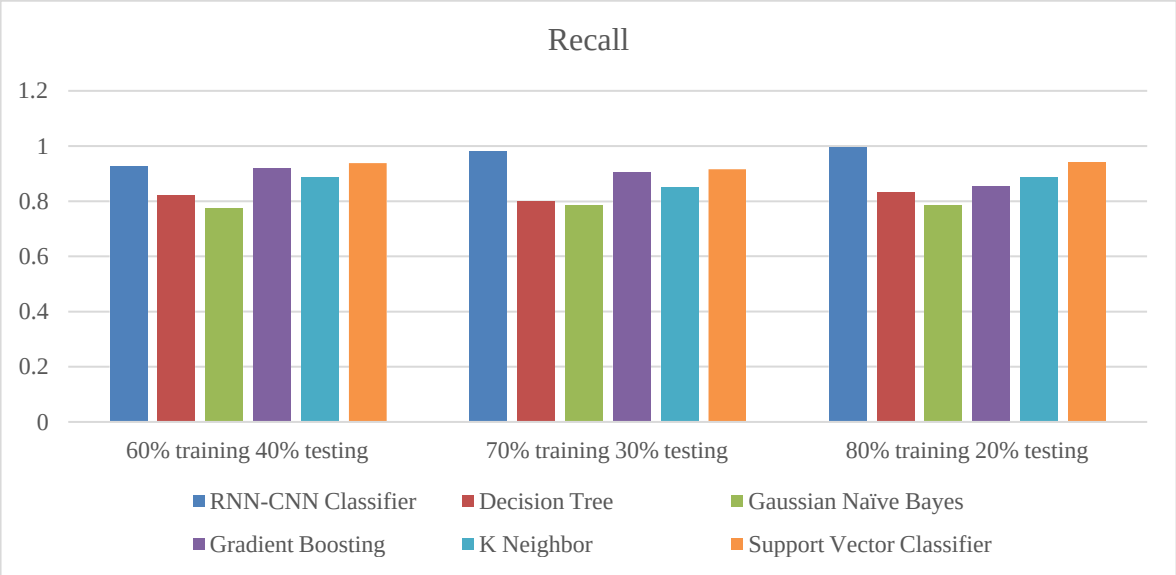

Fig 6.11 Recall comparison with ML Models at different training and testing ratios

As shown in Fig 6.11 RNN-CNN classifier yields the highest recall of 0.995 amongst all the given ML models and Gaussian naïve Bayes provides the lowest recall of 0.772. All the given ML models give the lowest recall value as compared to our proposed classifier. On the other hand, the RNN-CNN classifier again comes out to be the best among various DL models. VGG19 provides the lowest recall of 0.105 among given DL models as shown in fig 6.12.

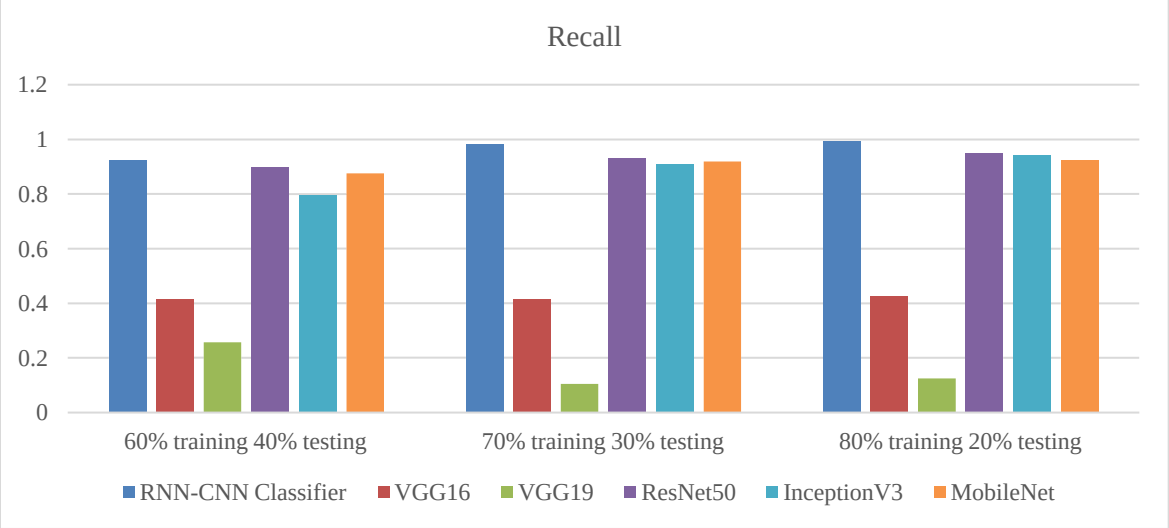

Fig 6.12 Recall comparison with DL Models at different training and testing ratios

As seen in Figures 6.13 and 6.14, the RNN-CNN classifier likewise fared well in terms of F1 score with both ML and DL models. The RNN-CNN classifier yields the greatest F1 score of 0.995 at 80% training data, similar to other performance measurements. The F1 scores of all other models, including decision trees, naïve Bayes, gradient boosting, KNN, and support vector classifiers, are smaller than those of RNN-CNN, ranging from 0.829 to 0.94 at 80% training and 20% testing data.

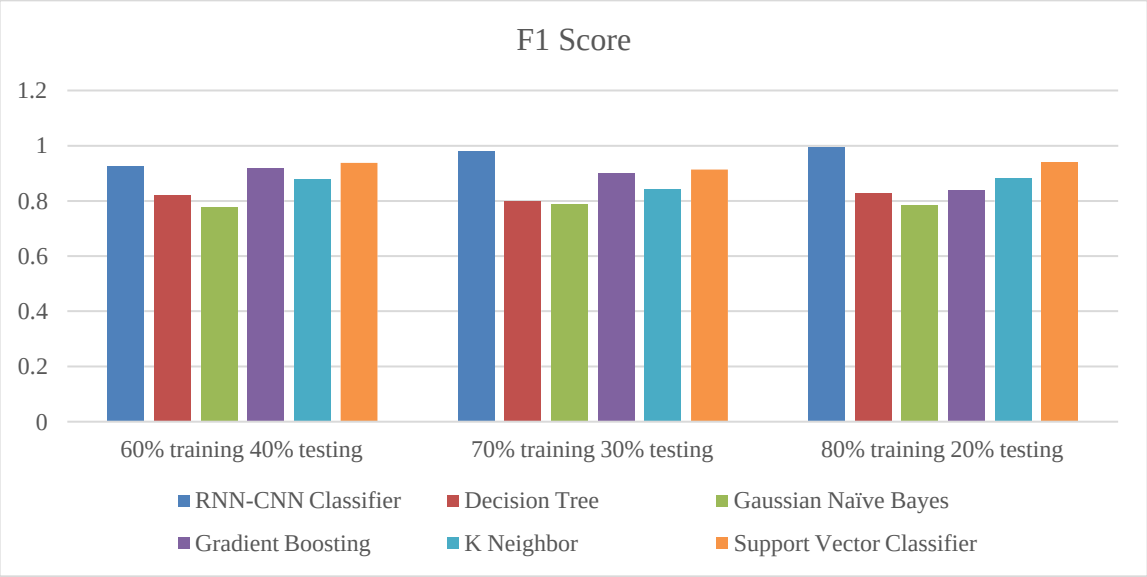

Fig 6.13 F1 Score comparison with ML Models at different training and testing ratios

At 60% training data, the Gaussian naïve Bayes algorithm yields the lowest F1 score value of all, 0.778. VGG19 is the least successful DL model out of all of them, with the lowest F1 score of 0.02 at 70% of training data and the maximum F1 score of 0.105 at 60%of training data.

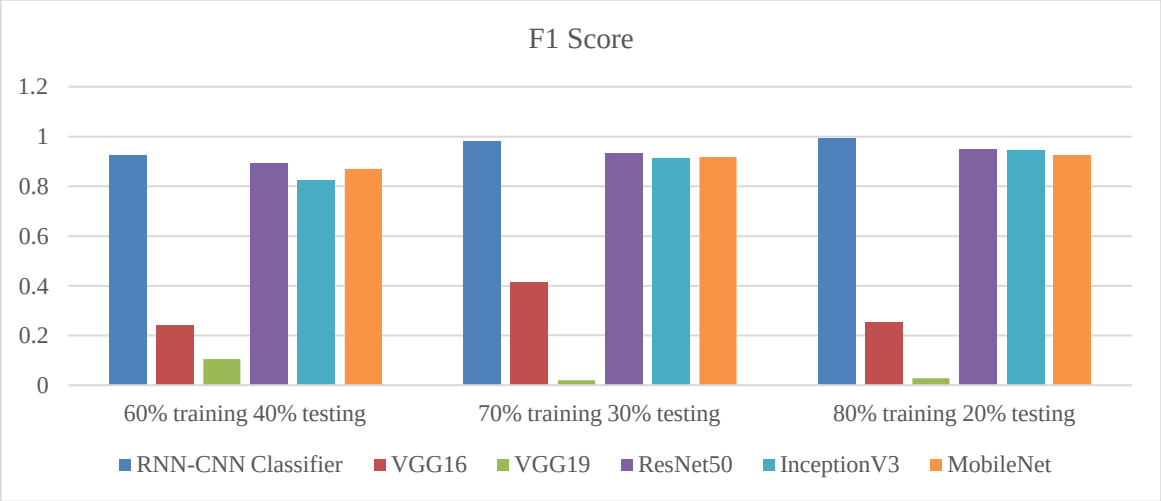

Fig 6.14 F1 Score comparison with DL Models at different training and testing ratios

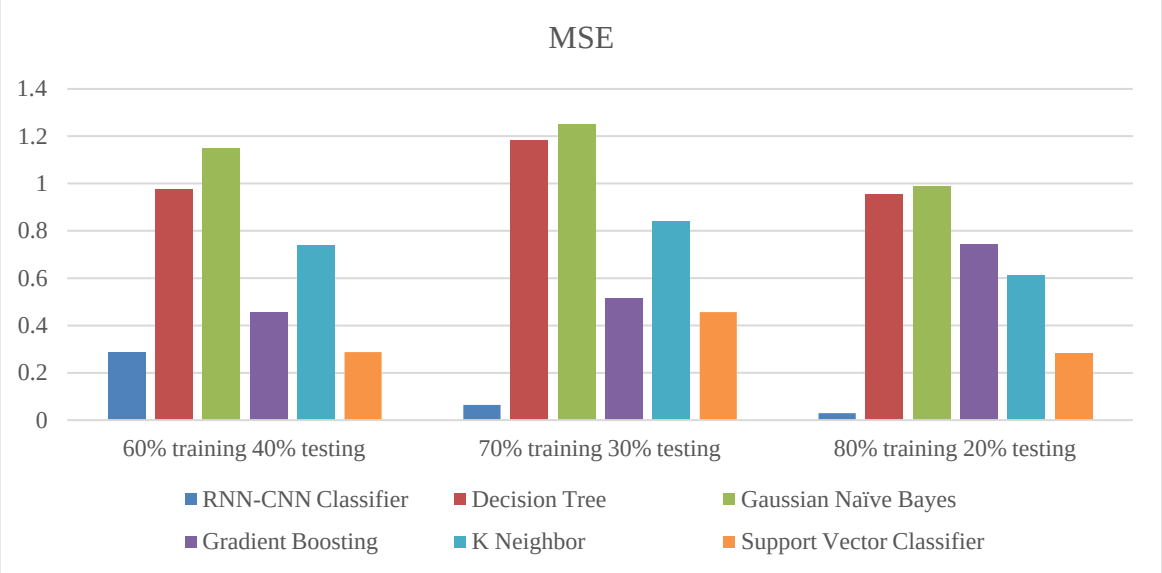

Fig 6.15 MSE comparison with ML Models at different training and testing ratios

Our proposed RNN-CNN classifier gives the lowest error i.e., MSE among all ML and DL models as shown in Fig 6.15 and 6.16. The lowest MSE of the proposed model is 0.029 with 80% training and 20% testing data. Gaussian naïve Bayes gives the highest MSE among all the ML models which is 1.25 at 70% training data. VGG19 gives the highest MSE among all the DL models which is 5.079 at 80% training data.

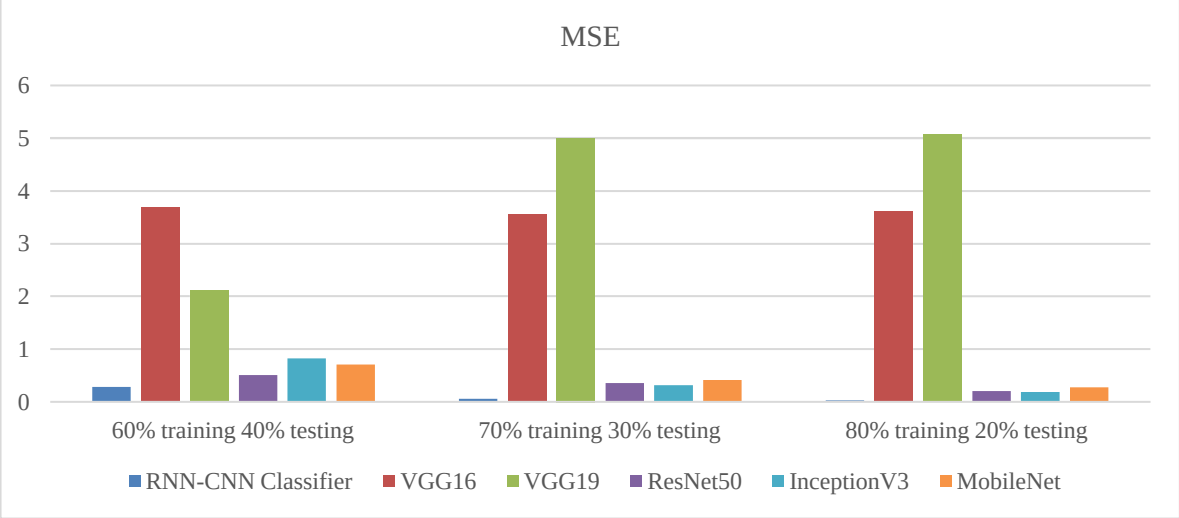

Fig 6.16 MSE comparison with DL Models at different training and testing ratios

6.5.2 Model evaluation on Dataset 2

The suggested RNN-CNN classifier has also been evaluated on dataset 2 as well which is available at https://archive.ics.uci.edu/ml/datasets/gene+expression+cancer+RNA-Seq#. After the successful feature extraction using the bottleneck feature extractor, the most prominent features are passed to the suggested classifier for further classification of cancer. To check the effectiveness of our classifier, the model has been checked for various performance measures as discussed in the previous sections. With 80% and 60% of the training data, respectively, the suggested RNN-CNN classifier produced the greatestaccuracy of 0.994 and the lowest accuracy of 0.946. However, as seen in fig. 6.17, VGG19 has produced the lowest accuracy of 0.149 using 80% training and 20% testing data. The proposed classifier also provided the highest precision value of 0.994 and VGG19 has provided the lowest precision which is 0.149. From Fig 6.18, Out of all the provided DL models, the suggested classifier is determined to be the most effective.

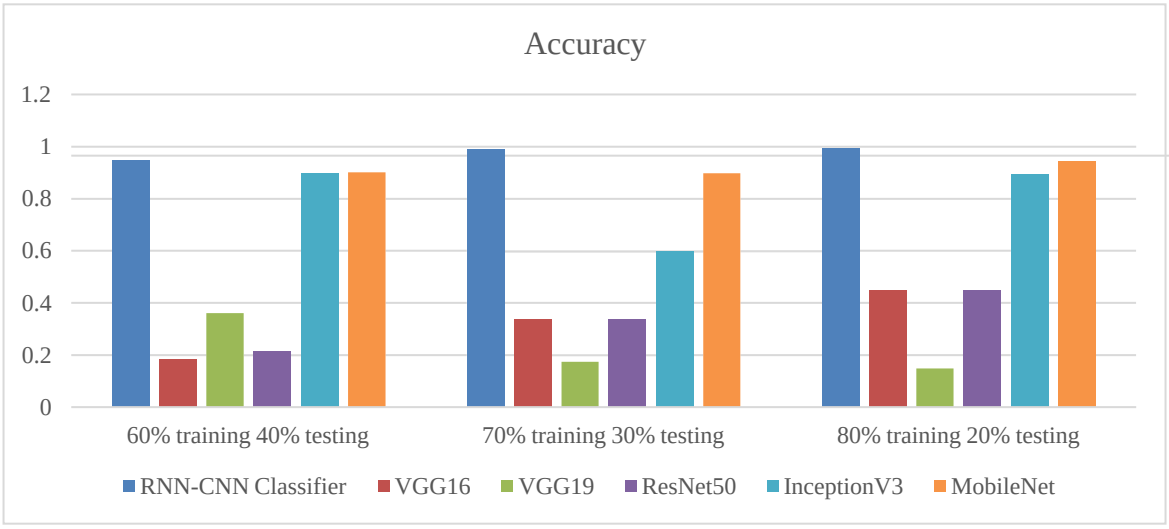

Fig 6.17 Accuracy comparison with DL Models at different training and testing ratios

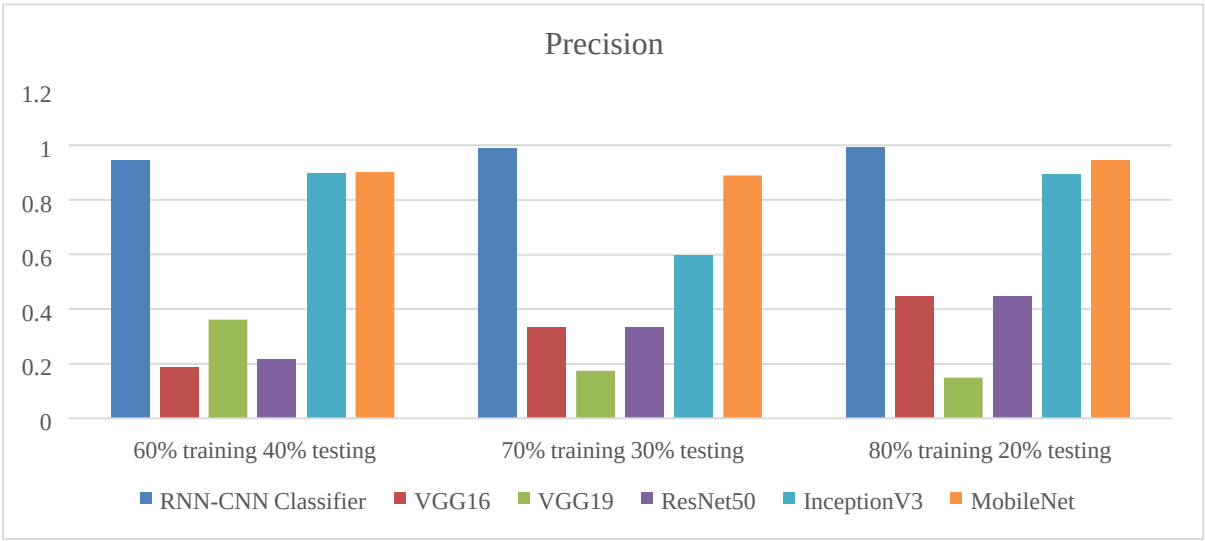

Fig 6.18 Precision comparison with DL Models at different training and testing ratios

The suggested RNN-CNN classifier has been assessed and contrasted with existing deep learning classifiers using recall, F1 score, and MSE, among other significant performance measures. At 80% of the training data, the suggested classifier yields the highest recall and F1 score (0.994 and 0.994, respectively). However, with 80% training data, VGG19 once more yields the lowest recall and F1 score, at 0.149 and 0.039, respectively. This again proved that the RNN-CNN classifier is the best among all the given models as shown in

Fig 6.19 and 6.20.

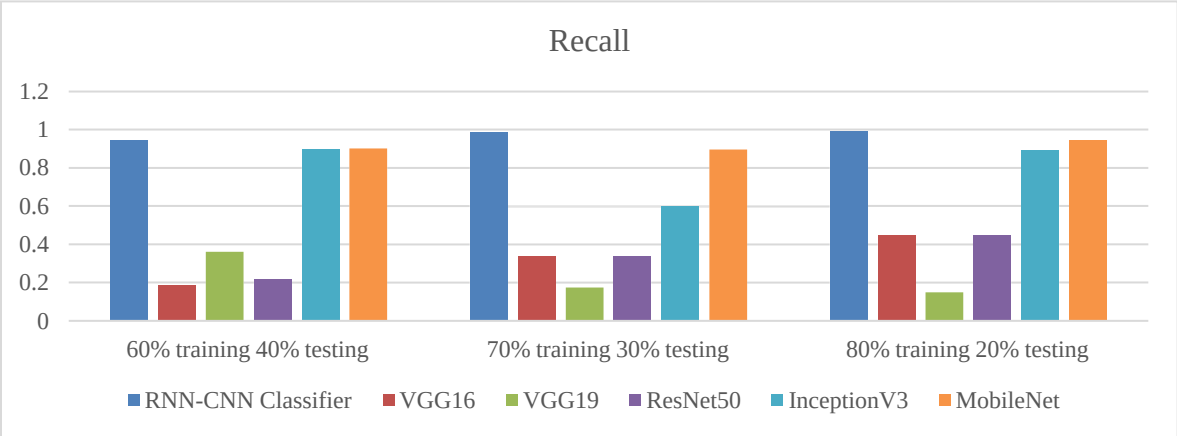

Fig 6.19 Recall comparison with DL Models at different training and testing ratios

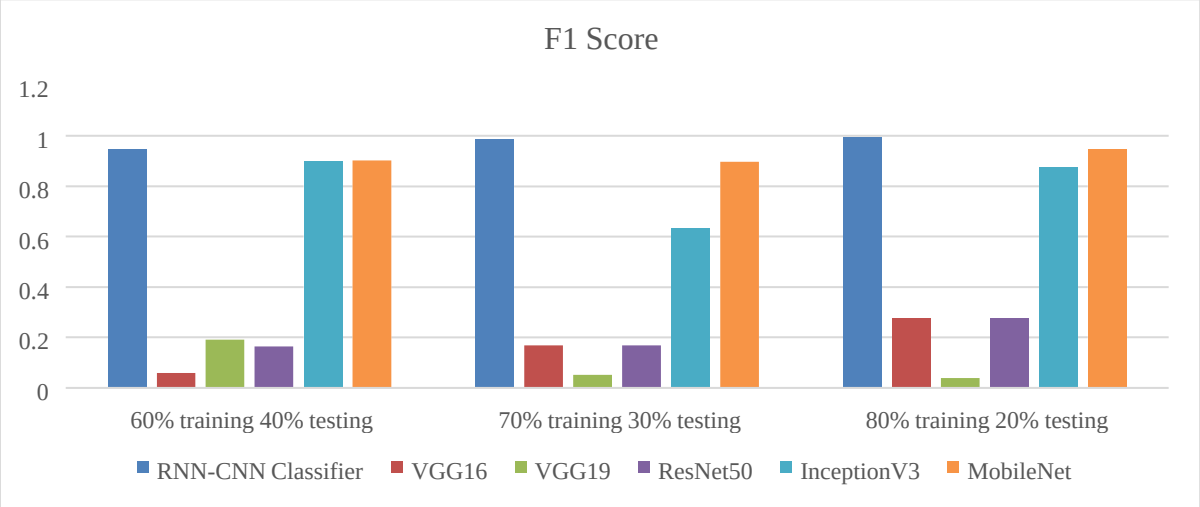

Fig 6.20 F1 Score comparison with DL Models at different training and testing ratios

As discussed above, MSE is also important for checking the effectiveness of the model. With 80% training data, the suggested classifier yields the lowest MSE of all—0.006. Of all the models provided, ResNet50 yields the greatest MSE. It provides an MSE of 4.919 with 60% training data as shown in Fig 6.21.

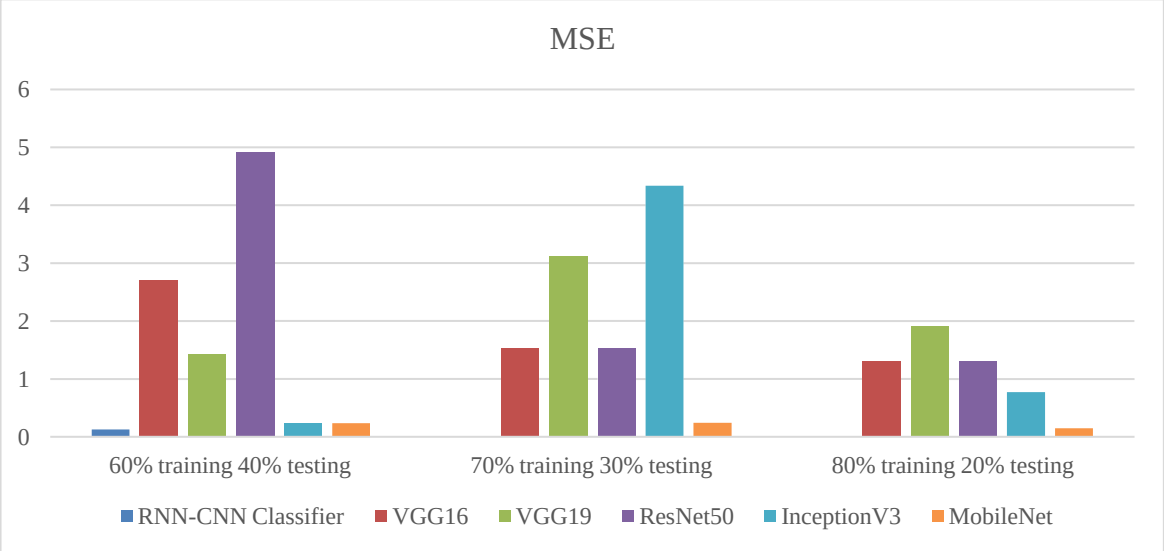

Fig 6.21 MSE comparison with DL Models at different training and testing ratios

6.5.3 Model Evaluation on Dataset 3

We also evaluated our proposed classifier on the Breast_GSE45827 dataset which is available online at https://www.kaggle.com/datasets/brunogrisci/breast-cancer-gene-expression-cumida. The model has been evaluated using various performance metrics as given.

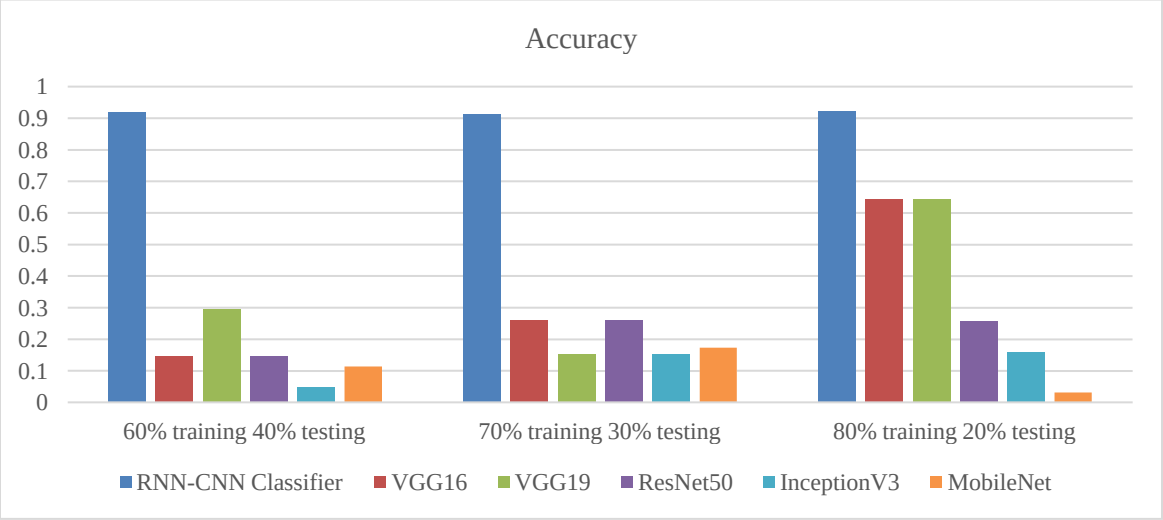

Fig 6.22 Accuracy comparison with DL Models at different training and testing ratios

The proposed RNN-CNN classifier gives the highest accuracies of all which are 0.924, 0.913, and 0.918 at 80%, 70%, and 60% training data as shown in Fig 6.22. On the other hand, all other models give very low accuracies on the given dataset. MobileNet gives the lowest accuracy among all at 80% training data which is 0.032.

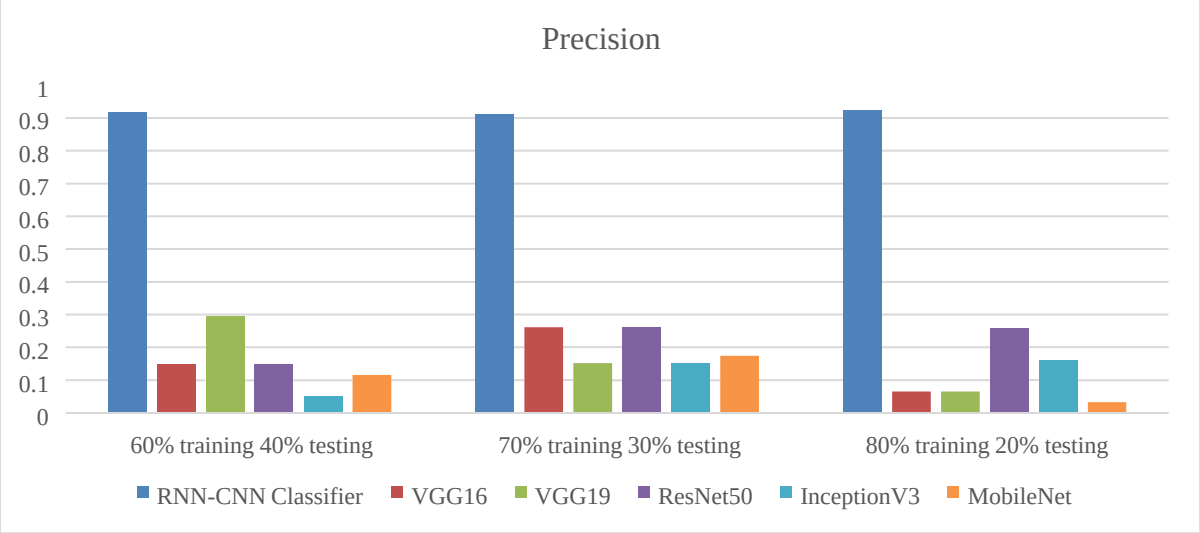

Fig 6.23 Precision comparison with DL Models at different training and testing ratios

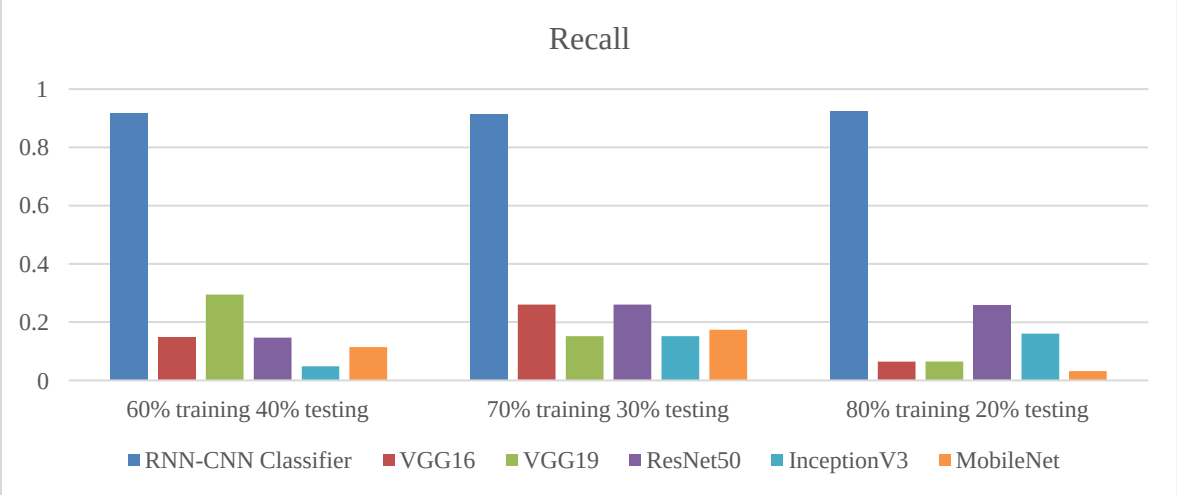

Fig 6.24 Recall comparison with DL Models at different training and testing ratios

As compared to other DL models given, the RNN-CNN classifier again performed well in terms of precision and recall. It gives the highest precision of 0.924 with 80% training data

and MobileNet gives the lowest precision among all which is 0.032 as shown in Fig 6.23. The highest recall value provided by the RNN-CNN is 0.924 as shown in fig 6.24. In the same way, MobileNet gives the lowest recall of 0.032. On the given dataset, none of the other models have performed well than the suggested RNN-CNN classifier which shows that the model is effective and efficient.

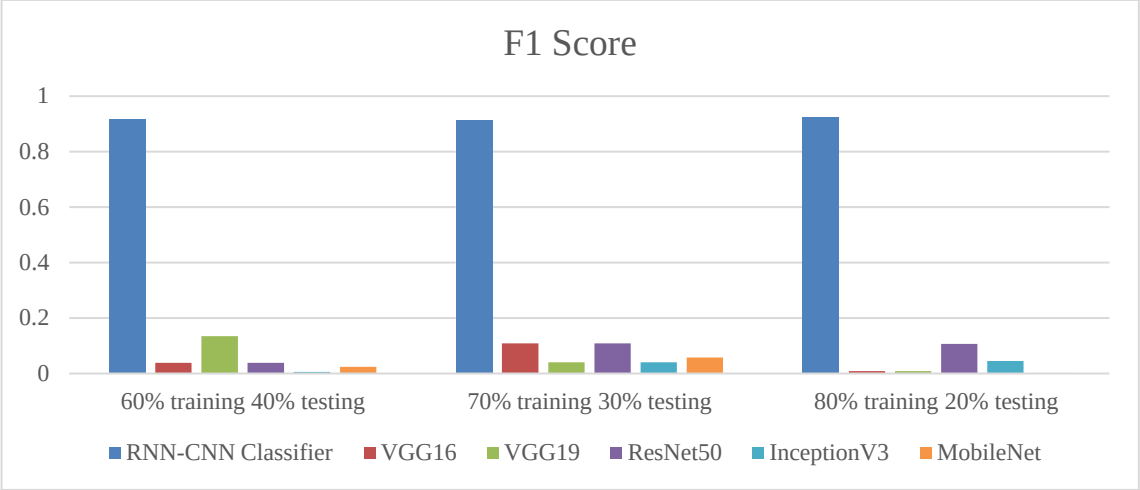

Fig 6.25 F1 Score comparison with DL Models at different training and testing ratios

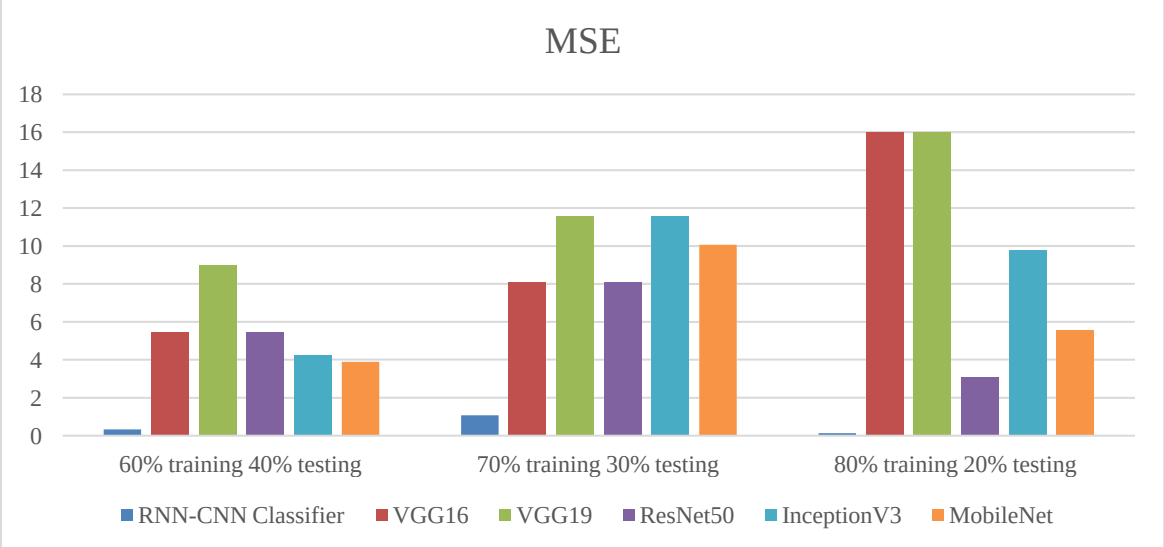

Fig 6.26 MSE comparison with DL Models at different training and testing ratios

As illustrated in fig 6.25 and 6.26, the proposed RNN-CNN classifier yields the highest F1 score of 0.924 and the lowest MSE among all which is 0.125. On the other hand, MobileNet gives the lowest F1 Score which is 0.002. In the case of MSE, VGG19 and Inception

V3 give the highest MSE of 11.543. This demonstrates that, when compared to all other DL and ML models, our suggested classifier performs the best according to several performance measures.

6.6 Summary

This chapter presents the proposed method for cancer classification. We proposed a deep learning-based classifier using RNN and CNN as per the third objective of our research. The proposed model uses three 2D convolution blocks, an RNN, and an LSTM block. The model consists of various other layers such as batch normalization, activation, dropout, and dense layer. Each layer has its role in the proposed model. We have compared our proposed classifier with DT, GNB, GB, XGB, KNN, VGG16, VGG19, ResNet50, Inception V3, and MobileNet. Compared to other state-of-the-art models, ours performs better.

CHAPTER 7

CONCLUSION AND FUTURE SCOPE

This chapter presents the conclusion derived from the proposed research work along with the future scope of the work that may help to further improve the results on other datasets as well. Artificial intelligence is becoming a fascinating technology that is providing effective solutions to different ongoing challenges. Machine learning and deep learning as the subfields of artificial intelligence have been used in different applications such as weather predictions, stock markets, pattern recognition, emotion detection, disease predictions, and many more. Seeing so many applications, wedecided to make some contributions in the field of cancer detection from gene expression data.

7.1 Conclusion

Diseases are a major threat to human life. One of the illnesses that poses the greatest risk to human health and harm a person's life is cancer. Cancer is one of the most prevalent causes of death in the human race. Cancer can be diagnosed using various types of samples such as blood, tissue, and various types of medical tests as well. It can also be diagnosed genetically, which is one of the significant techniques. From the literature review we have done so far, there are various machine learning methods such as decision tree, naïve Bayes, KNN, random forest, support vector classifier, and many more. Similarly, there are deep learning methods as well such as VGG16, VGG19, CNN, MobileNet, ResNet, XceptionNet, and many more with the help of which cancer prediction can be done.

As seen in the literature, there are a variety of tests available such as CT scans, MRI, and analysis using genetic data for detection of cancer. However, the use of microarray technology for genetic disease prediction is growing in popularity. Genetic data, on the other hand, has a problem with low sample numbers and large feature counts, or high dimensional data. It becomes difficult to implement the models on such data without dimensionality reduction either in terms of feature selection or feature extraction. By

keeping the two main concerns in mind such as dimensionality reduction and cancer classification, we carried out this research in terms of four objectives.

In order to achieve the first objective, we looked at and examined a number of current deep learning and machine learning techniques for cancer categorization or prediction using gene expression data. From the literature only, we got to know that gene expression data suffers from high dimensionality. So, we studied various techniques of dimensionality reduction. To figure out whether deep learning models are more efficient than machine learning models, we also compared the state-of-the-art methods for machine learning and deep learning.

The second objective is to perform the dimensionality reduction from the gene expression data. In this, we performed feature extraction. So, for this, we proposed a feature extraction method known as sandwich stacked bottleneck feature extraction for extracting the most prominent features. The proposed feature extraction method hasbeen developed using two pre-trained deep learning models namely VGG16 and VGG19. VGG19 has been stacked in between two VGG16s.

The third objective is to perform the classification of cancer from three different datasets, we proposed a second method known as RNN-CNN classifier. The proposed classifier consisted of 3 convolution blocks and one layer of simple RNN and LSTM for cancer prediction. The extracted features using the proposed feature extractor have been passed to the proposed classifier for the final classification of multiple types of cancer given in three different datasets.

To accomplish the fourth goal, the two proposed models have been evaluated using a range of performance metrics, including accuracy, precision, recall, F1 score, and MSE, across three distinct datasets. When compared to other ML and DL models found in the literature, the suggested models performed extremely well across all performance metrics. Compared to other current approaches, the suggested models exhibit superior efficacy in terms of high accuracy, precision, recall, f1 score, and lower MSE.

7.2 Future Scope

The use of deep learning techniques is providing promising results in terms of major

areas such as healthcare, fraud detection, agriculture, pattern recognition, and many more. Healthcare is one of those thrust areas which require major attention. With the advent of new technologies, it becomes easy nowadays to detect a disease and it becomes easier to find the drugs for the same. Cancer is one of those diseases which is dangerous for mankind. It should be detected in time for timely treatment.

Keeping the above scenario in mind, this research work has been carried out. This research work provides an efficient and effective feature extraction method and the classifier. The proposed feature extraction method extracts the most prominent features and then the classifier classifies the cancer into multiple types. The proposed methods have been implemented on secondary datasets. In the future, we may validate the efficacy of our work by applying the proposed methodologies to real-time data. Moreover, we can also employ a technique to detect any early signs of malignancy in the embryo, allowing for prompt action to prevent any repercussions down into the future. Also, gene expression datasets suffer from a problem of high dimensionality. That is why the datasets being used for the proposed research work carry only a few samples which makes the model data specific only. Also, DL models require high computation which does not make it fit for restricted environments. In the future, we might propose some data augmentation techniques to solve the problem of high dimensionality. Also, advanced structures of RNN can be explored. It can additionally be evaluated using more datasets and other DL models. Furthermore, other dimensionality reduction techniques can be explored.

List of Publications

- Gene expression-assisted cancer prediction techniques. *Journal of Healthcare Engineering, 2021*.
- A Comparative Analysis of various Machine Learning and Deep Learning Models for Gene Expression. In *2021 International Conference on Computing Sciences (ICCS)* (pp. 139-142). IEEE.
- Performance evaluation of various machine learning and deep learning models for gene expression. In *Journal of Physics: Conference Series* (Vol. 2327, No. 1, p. 012034). IOP Publishing.
- RNN-CNN Based Cancer Prediction Model for Gene Expression. *IEEE Access*, *11*, 131024-131044.
- Thakur, T., Batra, I., & Malik, A. (2025). Bottleneck Feature Extraction for Gene Expression Using Deep Learning. In *AI Techniques for Securing Medical and Business Practices* (pp. 311-332). IGI Global.

REFERENCES

[1] Slonim, D. K., Tamayo, P., Mesirov, J. P., Golub, T. R., & Lander, E. S. (2000, April). Class prediction and discovery using gene expression data.In *Proceedings of the fourth annual international conference on Computational molecular biology* (pp. 263-272).

[2] Gene regulation and expression. Accessed: Oct 2018. [Online]. Available: https://le.ac.uk/vgec/topics/gene-regulation#:~:text=The%20activity%20of%20a%20cell,organism%20from%20a%20single%20cell.

[3] Gene expression and function. Accessed: Oct 2018. [Online]. Available: https://www.khanacademy.org/test-prep/mcat/biomolecules/dna-technology/v/gene-expression-and-function

[4] DNA and RNA difference. Accessed: Oct 2018. [Online]. Available: byjus.com/biology/difference-between-dna-and-rna/

[5] Introduction to artificial intelligence. Accessed: Sept 2018. [Online]. Available: https://becominghuman.ai/introduction-to-artificial-intelligence-5fba0148ec99

[6] Mijwel, M. M. (2015). History of artificial intelligence. *Computer science, college of science*, 1-6.

[7] Artificial intelligence what it is and why it matters. Accessed: Sept 2018. [Online]. Available: https://www.sas.com/en_us/insights/analytics/what-is-artificial-intelligence.html

[8] Artificial intelligence. Accessed: Sept 2018. [Online]. Available: https://builtin.com/artificial-intelligence

[9] Artificial Intelligence (AI): What it is and How it is used. Accessed: Nov 2018. [Online]. Available: https://www.investopedia.com/terms/a/artificial-intelligence-ai.asp.

[10] Rupali, M., & Amit, P. (2017). A Review Paper on General Concepts of Artificial Intelligence and Machine Learning. *International Advanced Research Journal in Science, Engineering and Technology*, *4*, 79-82.

[11] Types of Artificial Intelligence. Accessed: Nov 2018. [Online]. Available: https://www.javatpoint.com/types-of-artificial-intelligence

[12] Lantz, B. (2013). *Machine learning with R*. Packt publishing ltd.

[13] Salas, J., de Barros Vidal, F., & Martinez-Trinidad, F. (2019). Deep Learning: Current State. *IEEE Latin America Transactions*, *17*(12), 1925- 1945.

[14] Goodfellow, I., Bengio, Y., & Courville, A. (2016). *Deep learning*. MITpress.

[15] What is deep learning. Accessed: Jan 2019. [Online]. Available: https://in.mathworks.com/discovery/deep-learning.html

[16] Shrestha, A., & Mahmood, A. (2019). Review of deep learning algorithms and architectures. *IEEE Access*, *7*, 53040-53065.

[17] Golub, T. R., Slonim, D. K., Tamayo, P., Huard, C., Gaasenbeek, M., Mesirov, J. P., ... & Bloomfield, C. D. (1999). Molecular classification of cancer: class discovery and class prediction by gene expression monitoring. *science*, *286*(5439), 531-537.

[18] Khan, J., Wei, J. S., Ringner, M., Saal, L. H., Ladanyi, M., Westermann, F., ... & Meltzer, P. S. (2001). Classification and diagnostic prediction of cancers using gene expression profiling and artificial neural networks. *Nature medicine*, *7*(6), 673-679.

[19] Radmacher, M. D., McShane, L. M., & Simon, R. (2002). A paradigm for class prediction using gene expression profiles. *Journal of computational biology*, *9*(3), 505-511.

[20] Pomeroy, S. L., Tamayo, P., Gaasenbeek, M., Sturla, L. M., Angelo, M., McLaughlin, M. E., ... & Allen, J. C. (2002). Prediction of central nervous system embryonal tumour outcome based on gene expression. *Nature*, *415*(6870), 436-442.

[21] Weng, S., Zhang, C., & Zhang, X. (2004). PCA-FA: Applying supervised learning to analyze gene expression data. *Tsinghua Science and Technology*, *9*(4), 428-434.

[22] Duan, K. B., Rajapakse, J. C., Wang, H., & Azuaje, F. (2005). Multiple SVM-RFE for gene selection in cancer classification with expressiondata. *IEEE transactions on nanobioscience*, *4*(3), 228-234.

[23] Berger, J. A., Hautaniemi, S., Mitra, S. K., & Astola, J. (2006). Jointly analyzing gene expression and copy number data in breast cancer using data reduction models. *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, *3*(1), 2-16.

[24] Xu, R., Anagnostopoulos, G. C., & Wunsch, D. C. (2007). Multiclass cancer classification using semisupervised ellipsoid ARTMAP and particle swarm optimization with gene expression data. *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, *4*(1), 65-77.

[25] Chiang, J. H., & Ho, S. H. (2008). A combination of rough-based feature selection and RBF neural network for classification using gene expression data. *IEEE transactions on nanobioscience*, *7*(1), 91-99.

[26] Yu, Z., & Wong, H. S. (2009). Class discovery from gene expression data based on perturbation and cluster ensemble. *IEEE Transactions on NanoBioscience*, *8*(2), 147-160.

[27] Hou, J., Aerts, J., Den Hamer, B., Van Ijcken, W., Den Bakker, M., Riegman, P., ... & Grosveld, F. (2010). Gene expression-based classification of non- small cell lung carcinomas and survival prediction. *PloS one*, *5*(4), e10312.

[28] Salazar, R., Roepman, P., Capella, G., Moreno, V., Simon, I., Dreezen, C., ... & Bruin, S. (2011). Gene expression signature to improve prognosis prediction of stage II and III colorectal cancer. *Journal of clinical oncology*, *29*(1), 17-24.

[29] Yuan, Y., Curtis, C., Caldas, C., & Markowetz, F. (2011). A sparse regulatory network of copy-number driven gene expression reveals putative breast cancer oncogenes. *IEEE/ACM transactions on computational biology and bioinformatics*, *9*(4), 947-954.

[30] Kim, H., & Gelenbe, E. (2012). Reconstruction of large-scale gene regulatory networks using bayesian model averaging. *IEEE Transactions on NanoBioscience*, *11*(3), 259-265.

[31] Ashraf, A. B., Gavenonis, S. C., Daye, D., Mies, C., Rosen, M. A., & Kontos, D. (2012). A multichannel Markov random field framework for tumor segmentation with an application to classification of gene expression-based breast cancer recurrence risk. *IEEE transactions on medical imaging*, *32*(4), 637-648.

[32] Liao, B., Jiang, Y., Liang, W., Zhu, W., Cai, L., & Cao, Z. (2014). Gene selection using locality sensitive Laplacian score. *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, *11*(6), 1146-1156.

[33] Liu, J. X., Xu, Y., Zheng, C. H., Kong, H., & Lai, Z. H. (2014). RPCA-based tumor classification using gene expression data. *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, *12*(4), 964-970.

[34] Chen, Y., Li, Y., Narayan, R., Subramanian, A., & Xie, X. (2016). Gene expression inference with deep learning. *Bioinformatics*, *32*(12), 1832-1839.

[35] Deng, S. P., Cao, S., Huang, D. S., & Wang, Y. P. (2016). Identifying stages of kidney renal cell carcinoma by combining gene expression and DNA methylation data. *IEEE/ACM transactions on computational biology and bioinformatics*, *14*(5), 1147-1153.

[36] Chen, L., Pan, X., Zhang, Y. H., Liu, M., Huang, T., & Cai, Y. D. (2019). Classification of widely and rarely expressed genes with recurrent neural network. *Computational and Structural Biotechnology Journal*, *17*, 49-60.

[37] Elbashir, M. K., Ezz, M., Mohammed, M., & Saloum, S. S. (2019). Lightweight Convolutional Neural Network for Breast Cancer Classification Using RNA-Seq Gene Expression Data. *IEEE Access*, *7*, 185338-185348.

[38] Khalifa, N. E. M., Taha, M. H. N., Ali, D. E., Slowik, A., & Hassanien, A. E. (2020). Artificial Intelligence Technique for Gene Expression by Tumor RNA-Seq Data: A Novel Optimized Deep Learning Approach. *IEEE Access*, *8*, 22874-22883.

[39] Hedenfalk, I., Duggan, D., Chen, Y., Radmacher, M., Bittner, M., Simon, R., ... & Yakhini, Z. (2001). Gene-expression profiles in hereditary breast cancer. *New England Journal of Medicine*, *344*(8), 539-548.

[40] Bhattacharjee, A., Richards, W. G., Staunton, J., Li, C., Monti, S., Vasa, P., ... & Loda, M. (2001). Classification of human lung carcinomas by mRNA expression profiling reveals distinct adenocarcinoma subclasses. *Proceedings of the National Academy of Sciences*, *98*(24), 13790-13795.

[41] Su, A. I., Cooke, M. P., Ching, K. A., Hakak, Y., Walker, J. R., Wiltshire, T., ... & Patapoutian, A. (2002). Large-scale analysis of the human and mouse transcriptomes. *Proceedings of the National Academy of Sciences*, *99*(7), 4465-4470.

[42] Yu, Z., Wongb, H. S., You, J., Yang, Q., & Liao, H. (2011). Knowledge based cluster ensemble for cancer discovery from biomolecular data. *IEEE transactions on nanobioscience*, *10*(2), 76-85.

[43] Chin, K., DeVries, S., Fridlyand, J., Spellman, P. T., Roydasgupta, R., Kuo, W. L., ... & Chen, F. (2006). Genomic and transcriptional aberrations linked to breast cancer pathophysiologies. *Cancer cell*, *10*(6), 529-541.

[44] Uhlen, M., Zhang, C., Lee, S., Sjöstedt, E., Fagerberg, L., Bidkhori, G., ... & Sanli, K. (2017). A pathology atlas of the human cancertranscriptome.*Science*, *357*(6352).

[45] Zheng, C. H., Ng, T. Y., Zhang, L., Shiu, C. K., & Wang, H. Q. (2011). Tumor classification based on non-negative matrix factorization using gene expression data. *IEEE transactions on nanobioscience*, *10*(2), 86-93.

[46] Alon, U., Barkai, N., Notterman, D. A., Gish, K., Ybarra, S., Mack, D., & Levine, A. J. (1999). Broad patterns of gene expression revealed by clustering analysis of tumor and normal colon tissues probed by oligonucleotide arrays. *Proceedings of the National Academy of Sciences*, *96*(12), 6745-6750.

[47] Wang, X., Liu, J., Cheng, Y., Liu, A., & Chen, E. (2018). Dual Hypergraph Regularized PCA for Biclustering of Tumor Gene Expression Data. *IEEE Transactions on Knowledge and Data Engineering*, *31*(12), 2292-2303.

[48] Wang, H., Li, C., Zhang, J., Wang, J., Ma, Y., & Lian, Y. (2019). A new LSTM-based gene expression prediction model: L-GEPM. *Journal of Bioinformatics and Computational Biology*, *17*(04), 1950022.

[49] Khorshed, T., Moustafa, M. N., & Rafea, A. (2020). Deep Learning for Multi-Tissue Cancer Classification of Gene Expressions (GeneXNet). *IEEE Access*, *8*, 90615-90629.

[50] Ji, G., Yang, Z., & You, W. (2010). PLS-based gene selection and identification of tumor-specific genes. *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, *41*(6), 830-841.

[51] Hsu, F. H., Serpedin, E., Chen, Y., & Dougherty, E. R. (2012). Evaluating dynamic effects of copy number alterations on gene expression using a single transcription model. *IEEE transactions on biomedical engineering*, *59*(10), 2726-2736.

[52] Farouq, M. W., Boulila, W., Abdel-Aal, M., Hussain, A., & Salem, A. B. (2019). A novel multi-stage fusion based approach for gene expression profiling in non-small cell lung cancer. *IEEE Access*, *7*, 37141-37150.

[53] Kiranyaz, S., Avci, O., Abdeljaber, O., Ince, T., Gabbouj, M., & Inman, D. J. (2019). 1D convolutional neural networks and applications: A survey. *Mechanical Systems and Signal Processing*, *151*, 107398.

[54] Wu, Y., Yang, F., Liu, Y., Zha, X., & Yuan, S. (2018). A comparison of 1-D and 2-D deep convolutional neural networks in ECG classification. *arXiv preprint arXiv:1810.07088*.

[55] Géron, A. (2017). Hands-on machine learning with scikit-learn and tensorflow: Concepts. *Tools, and Techniques to build intelligent systems*.

[56] Russell, S. J. (2010). *Artificial intelligence a modern approach*. Pearson Education, Inc..

[57] Sazli, M. H. (2006). A brief review of feed-forward neural networks. *Communications Faculty of Sciences University of Ankara Series A2-A3 Physical Sciences and Engineering*, *50*(01).

[58] LeCun, Y., Bottou, L., Bengio, Y., & Haffner, P. (1998). Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, *86*(11), 2278-2324.

[59] LeCun, Y., & Bengio, Y. (1995). Convolutional networks for images, speech, and time series. *The handbook of brain theory and neural networks*, *3361*(10), 1995.

[60] Taylor, G. W., Fergus, R., LeCun, Y., & Bregler, C. (2010, September). Convolutional learning of spatio-temporal features. In *European conference on computer vision* (pp. 140-153). Springer, Berlin, Heidelberg.

[61] Du, K. L., & Swamy, M. N. (2013). *Neural networks and statistical learning*. Springer Science & Business Media.

[62] Medsker, L., & Jain, L. C. (Eds.). (1999). *Recurrent neural networks: design and applications*. CRC press.

[63] Zeng, M., Li, M., Fei, Z., Wu, F. X., Li, Y., Pan, Y., & Wang, J. (2019). A deep learning framework for identifying essential proteins by integrating multiple types of biological information. *IEEE/ACM transactions on computational biology and bioinformatics*, *18*(1), 296-305.

[64] Liu, M., Hu, L., Tang, Y., Wang, C., He, Y., Zeng, C., ... & Huo, W. (2022). A deep learning method for breast cancer classification in the pathology images. *IEEE Journal of Biomedical and Health Informatics*, *26*(10), 5025- 5032.

[65] Xu, J., Wu, P., Chen, Y., Meng, Q., Dawood, H., & Khan, M. M. (2019). A novel deep flexible neural forest model for classification of cancer subtypes based on gene expression data. *IEEE Access*, *7*, 22086-22095.

[66] Peng, C., Zheng, Y., & Huang, D. S. (2019). Capsule network based modeling of multi-omics data for discovery of breast cancer-related genes. *IEEE/ACM transactions on computational biology and bioinformatics*, *17*(5), 1605-1612.

[67] Zhong, C., Yan, K., Dai, Y., Jin, N., & Lou, B. (2019). Energy efficiency solutions for buildings: Automated fault diagnosis of air handling units using generative adversarial networks. *Energies*, *12*(3), 527.

[68] Yan, K., Zhong, C., Ji, Z., & Huang, J. (2018). Semi-supervised learning for early detection and diagnosis of various air handling unit faults. *Energy and Buildings*, *181*, 75-83.

[69] Wang, S. L., Li, X., Zhang, S., Gui, J., & Huang, D. S. (2010). Tumor classification by combining PNN classifier ensemble with neighborhood rough set based gene reduction. *Computers in Biology and Medicine*, *40*(2), 179-189.

[70] Dong, H., Li, T., Ding, R., & Sun, J. (2018). A novel hybrid genetic algorithm with granular information for feature selection andoptimization. *Applied Soft Computing*, *65*, 33-46.

[71] Lu, H., Gao, H., Ye, M., & Wang, X. (2019). A hybrid ensemble algorithm combining AdaBoost and genetic algorithm for cancer classification with gene expression data. *IEEE/ACM transactions on computational biology and bioinformatics*.

[72] Zhang, X., He, D., Zheng, Y., Huo, H., Li, S., Chai, R., & Liu, T. (2020). Deep learning based analysis of breast cancer using advanced ensemble classifier and linear discriminant analysis. *IEEE Access*, *8*, 120208-120217.

[73] Wang, Y., Ma, Z., Wong, K. C., & Li, X. (2020). Evolving Multiobjective Cancer Subtype Diagnosis from Cancer Gene Expression Data. *IEEE/ACM transactions on computational biology and bioinformatics*.

[74] Hu, F., Zhou, Y., Wang, Q., Yang, Z., Shi, Y., & Chi, Q. (2019). Gene expression classification of lung adenocarcinoma into molecular subtypes. *IEEE/ACM transactions on computational biology and bioinformatics*, *17*(4), 1187-1197.

[75] De Souza, J. T., De Francisco, A. C., & De Macedo, D. C. (2019). Dimensionality reduction in gene expression data sets. *IEEE Access*, *7*, 61136-61144.

[76] Yang, W. H., Dai, D. Q., & Yan, H. (2008). Feature extraction and uncorrelated discriminant analysis for high-dimensional data. *IEEE transactions on knowledge and data engineering*, *20*(5), 601-614.

[77] Zhang, D., Zou, L., Zhou, X., & He, F. (2018). Integrating feature selection and feature extraction methods with deep learning to predict clinical outcome of breast cancer. *IEEE Access*, *6*, 28936-28944.

[78] Tang, Y., Zhang, Y. Q., Huang, Z., Hu, X., & Zhao, Y. (2008). Recursive fuzzy granulation for gene subsets extraction and cancer classification. *IEEE Transactions on Information Technology in Biomedicine*, *12*(6), 723-730.

[79] Park, K. H., Batbaatar, E., Piao, Y., Theera-Umpon, N., & Ryu, K. H. (2021). Deep learning feature extraction approach for hematopoietic cancer subtype classification. *International Journal of Environmental Research and Public Health*, *18*(4), 2197.

[80] Li, S., Liao, C., & Kwok, J. T. (2006, October). Gene feature extraction using T-test statistics and kernel partial least squares. In *International Conference on Neural Information Processing* (pp. 11-20). Springer, Berlin, Heidelberg.

[81] Yu, K., Huang, M., Chen, S., Feng, C., & Li, W. (2022). GSEnet: feature extraction of gene expression data and its application to Leukemia classification. *Mathematical Biosciences and Engineering*, *19*(5), 4881-4891.

[82] Mondol, R. K., Truong, N. D., Reza, M., Ippolito, S., Ebrahimie, E., &Kavehei, O. (2021). AFExNet: An adversarial autoencoder for differentiating breast cancer sub-types and extracting biologically relevant genes. *IEEE/ACM Transactions on Computational Biology and Bioinformatics*.

[83] Arya, N., & Saha, S. (2020). Multi-modal classification for human breast cancer prognosis prediction: proposal of deep-learning based stacked ensemble model. *IEEE/ACM transactions on computational biology and bioinformatics*.

[84] Chen, L., Pan, X., Zhang, Y. H., Liu, M., Huang, T., & Cai, Y. D. (2019). Classification of widely and rarely expressed genes with recurrent neural network. *Computational and structural biotechnology journal*, *17*, 49-60.

[85] Uhlen, M., Zhang, C., Lee, S., Sjöstedt, E., Fagerberg, L., Bidkhori, G., ... & Ponten, F. (2017). A pathology atlas of the human cancer transcriptome. *Science*, *357*(6352), eaan2507.

[86] Özgül, O. F., Bardak, B., & Tan, M. (2020). A convolutional deep clustering framework for gene expression time series. *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, *18*(6), 2198-2207.

[87] Guillemin, K., Salama, N. R., Tompkins, L. S., & Falkow, S. (2002). Cag pathogenicity island-specific responses of gastric epithelial cells to Helicobacter pylori infection. *Proceedings of the National Academy of Sciences*, *99*(23), 15136-15141.

[88] Tan, K., Huang, W., Liu, X., Hu, J., & Dong, S. (2021). A hierarchical graph convolution network for representation learning of gene expression data. *IEEE Journal of Biomedical and Health Informatics*, *25*(8), 3219-3229.

[89] Sun, D., Wang, M., & Li, A. (2018). A multimodal deep neural network for human breast cancer prognosis prediction by integrating multi-dimensional data. *IEEE/ACM transactions on computational biology and bioinformatics*, *16*(3), 841-850.

[90] Jasim, M., Machado, L., Al-Shamery, E. S., Ajit, S., Anthony, K., Mu, M., & Agyeman, M. O. (2022). A Survey of Machine Learning Approaches Applied to Gene Expression Analysis for Cancer Prediction. *IEEE Access*.

[91] Mudiyanselage, T. K. B., Xiao, X., Zhang, Y., & Pan, Y. (2019). Deep fuzzy neural networks for biomarker selection for accurate cancer detection. *IEEE Transactions on Fuzzy Systems*, *28*(12), 3219-3228.

[92] Zhang, J., Chen, Q., & Liu, B. (2019). DeepDRBP-2L: a new genome annotation predictor for identifying DNA-binding proteins and RNA-binding proteins using convolutional neural network and long short-term memory. *IEEE/ACM transactions on computational biology and bioinformatics*, *18*(4), 1451-1463.

[93] Rehman, M. U., Shafique, A., Ghadi, Y. Y., Boulila, W., Jan, S. U., Gadekallu, T. R., ... & Ahmad, J. (2022). A Novel Chaos-Based Privacy- Preserving Deep Learning Model for Cancer Diagnosis. *IEEE Transactions onNetwork Science and Engineering*, *9*(6), 4322-4337.

[94] Zheng, R., Wang, Q., Lv, S., Li, C., Wang, C., Chen, W., & Wang, H. (2022). Automatic liver tumor segmentation on dynamic contrast enhanced mri using

4D information: deep learning model based on 3D convolution and convolutional lstm. *IEEE Transactions on Medical Imaging*, *41*(10), 2965-2976.

[95] Tomasetti, C., & Vogelstein, B. (2015). Variation in cancer risk amongtissues can be explained by the number of stem cell divisions. *Science*, *347*(6217), 78-81.

[96] Using bottleneck features for multi-class classification in keras andtensorflow. Accessed: Jan 2021. [Online]. Available: https://www.codesofinterest.com/2017/08/bottleneck-features-multi-class-classification-keras.html.

[97] Grézl, F., & Karafiát, M. (2016). Bottle-neck feature extraction structures for multilingual training and porting. *Procedia Computer Science*, *81*, 144-151.

[98] Koh, P. W., Nguyen, T., Tang, Y. S., Mussmann, S., Pierson, E., Kim, B., & Liang, P. (2020, November). Concept bottleneck models. In *International Conference on Machine Learning* (pp. 5338-5348). PMLR.

[99] Qi, J., Wang, D., & Tejedor Noguerales, J. (2013). Subspace models for bottleneck features. In *Interspeech*. International Speech Communication Association.

[100] Yu, D., & Seltzer, M. L. (2011). Improved bottleneck features using pretrained deep neural networks. In *Twelfth annual conference of the international speech communication association*.

[101] Adametz, D., Rey, M., & Roth, V. (2014, September). Information Bottleneck for Pathway-Centric Gene Expression Analysis. In *German Conference on Pattern Recognition* (pp. 81-91). Springer, Cham.

[102] Lozano-Diez, A., Silnova, A., Matejka, P., Glembek, O., Plchot, O., Pesan, J., ... & Gonzalez-Rodriguez, J. (2016, June). Analysis and Optimization of Bottleneck Features for Speaker Recognition. In *Odyssey* (Vol. 2016, pp. 352-357).

[103] Naronglerdrit, P. (2019, August). Facial Expression Recognition: A Comparison of Bottleneck Feature Extraction. In *2019 Twelfth International Conference on Ubi-Media Computing (Ubi-Media)* (pp. 164-167). IEEE.

[104] Lo, W. W., Yang, X., & Wang, Y. (2019, June). An xception convolutional neural network for malware classification with transfer learning. In *2019 10th IFIP International Conference on New Technologies, Mobility and Security (NTMS)* (pp. 1-5). IEEE.

[105] Khaki, S., Pham, H., Han, Y., Kuhl, A., Kent, W., & Wang, L. (2021). Deepcorn: A semi-supervised deep learning method for high-throughput image-based corn kernel counting and yield estimation. *Knowledge-Based Systems*, *218*, 106874.

[106] Rezende, E., Ruppert, G., Carvalho, T., Theophilo, A., Ramos, F., & Geus, P. D. (2018). Malicious software classification using VGG16 deepneural network's bottleneck features. In *Information Technology-New Generations* (pp. 51-59). Springer, Cham.

[107] Li, W., Wang, Z., Wang, Y., Wu, J., Wang, J., Jia, Y., & Gui, G. (2020). Classification of high-spatial-resolution remote sensing scenes methodusing transfer learning and deep convolutional neural network. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, *13*, 1986-1995.

[108] Swati, Z. N. K., Zhao, Q., Kabir, M., Ali, F., Ali, Z., Ahmed, S., & Lu, J. (2019). Content-based brain tumor retrieval for MR images using transfer learning. *IEEE Access*, *7*, 17809-17822.

[109] Chintha, A., Rao, A., Sohrawardi, S., Bhatt, K., Wright, M., & Ptucha, R. (2020). Leveraging edges and optical flow on faces for deepfake detection. In *2020 IEEE international joint conference on biometrics (IJCB)* (pp. 1-10). IEEE.

[110] Demir, F., Sobahi, N., Siuly, S., & Sengur, A. (2021). Exploring deep learning features for automatic classification of human emotion using EEG rhythms. *IEEE Sensors Journal*, *21*(13), 14923-14930.

[111] Kwak, B. I., Han, M. L., & Kim, H. K. (2020). Driver identification based on wavelet transform using driving patterns. *IEEE Transactions on Industrial Informatics*, *17*(4), 2400-2410.

[112] Theckedath, D., & Sedamkar, R. R. (2020). Detecting affect states using VGG16, ResNet50 and SE-ResNet50 networks. *SN Computer Science*, *1*(2), 1-7.

[113] Bansal, M., Kumar, M., Sachdeva, M., & Mittal, A. (2021). Transfer learning for image classification using VGG19: Caltech-101 image data set. *Journal of Ambient Intelligence and Humanized Computing*, 1-12.

[114] Filipp, F. V. (2017). Precision medicine driven by cancer systems biology. *Cancer and Metastasis Reviews*, *36*(1), 91-108.

[115] Archer, T. C., Fertig, E. J., Gosline, S. J., Hafner, M., Hughes, S. K., Joughin, B. A., ... & Shajahan-Haq, A. N. (2016). Systems Approaches to Cancer BiologySystems Approaches to Cancer Biology. *Cancer research*, *76*(23), 6774-6777.

[116] Vos, T., Allen, C., Arora, M., Barber, R. M., Bhutta, Z. A., Brown, A., ... & Boufous, S. (2016). Global, regional, and national incidence, prevalence, and years lived with disability for 310 diseases and injuries, 1990–2015: a systematic analysis for the Global Burden of Disease Study 2015. *The lancet*, *388*(10053), 1545-1602.

[117] Liu, S., Xu, C., Zhang, Y., Liu, J., Yu, B., Liu, X., & Dehmer, M. (2018). Feature selection of gene expression data for cancer classification using double RBF-kernels. *BMC bioinformatics*, *19*(1), 1-14.

[118] Finotello, F., & Di Camillo, B. (2015). Measuring differential gene expression with RNA-seq: challenges and strategies for data analysis. *Briefings in functional genomics*, *14*(2), 130-142.

[119] Maienschein-Cline, M., Zhou, J., White, K. P., Sciammas, R., & Dinner, A. R. (2012). Discovering transcription factor regulatory targets using gene expression and binding data. *Bioinformatics*, *28*(2), 206-213.

[120] Schadt, E. E., Lamb, J., Yang, X., Zhu, J., Edwards, S., GuhaThakurta, D., ... & Lusis, A. J. (2005). An integrative genomics approach to infer causal associations between gene expression and disease. *Nature genetics*, *37*(7), 710-717.

[121] Shabana, K. M., Abdul Nazeer, K. A., Pradhan, M., & Palakal, M. (2015). A computational method for drug repositioning using publicly available gene expression data. *BMC bioinformatics*, *16*(17), 1-9.

[122] Danaee, P., Ghaeini, R., & Hendrix, D. A. (2017). A deep learning approach for cancer detection and relevant gene identification. In *Pacific symposium on biocomputing 2017* (pp. 219-229).

[123] Alizadeh, A. A., Eisen, M. B., Davis, R. E., Ma, C., Lossos, I. S., Rosenwald, A., ... & Staudt, L. M. (2000). Distinct types of diffuse large B- cell lymphoma identified by gene expression profiling. *Nature*, *403*(6769),503-511.

[124] Jasim, M., Machado, L., Al-Shamery, E. S., Ajit, S., Anthony, K., Mu, M., & Agyeman, M. O. (2022). A Survey of Machine Learning Approaches Applied to Gene Expression Analysis for Cancer Prediction. *IEEE Access*.

[125] Swain, P. H., & Hauska, H. (1977). The decision tree classifier: Design and potential. *IEEE Transactions on Geoscience Electronics*, *15*(3), 142-147.

[126] Safavian, S. R., & Landgrebe, D. (1991). A survey of decision tree classifier methodology. *IEEE transactions on systems, man, and cybernetics*, *21*(3), 660-674.

[127] Priyanka, & Kumar, D. (2020). Decision tree classifier: A detailed survey. *International Journal of Information and Decision Sciences*, *12*(3),246-269.

[128] Lusa, L. (2017). Gradient boosting for high-dimensional prediction of rare events. *Computational Statistics & Data Analysis*, *113*, 19-37.

[129] Peter, S., Diego, F., Hamprecht, F. A., & Nadler, B. (2017). Cost efficient gradient boosting. *Advances in neural information processing systems*, *30*.

[130] Kamel, H., Abdulah, D., & Al-Tuwaijari, J. M. (2019, June). Cancer classification using gaussian naive bayes algorithm. In *2019 International Engineering Conference (IEC)* (pp. 165-170). IEEE.

[131] Jahromi, A. H., & Taheri, M. (2017, October). A non-parametric mixture of Gaussian naive Bayes classifiers based on local independent

features. In *2017 Artificial intelligence and signal processing conference (AISP)* (pp. 209-212). IEEE.

[132] Ontivero-Ortega, M., Lage-Castellanos, A., Valente, G., Goebel, R., & Valdes-Sosa, M. (2017). Fast Gaussian Naïve Bayes for searchlight classification analysis. *Neuroimage*, *163*, 471-479.

[133] Viswanath, P., & Sarma, T. H. (2011, September). An improvement to k-nearest neighbor classifier. In *2011 IEEE Recent Advances in Intelligent Computational Systems* (pp. 227-231). IEEE.

[134] Horton, P., & Nakai, K. (1997, June). Better Prediction of Protein Cellular Localization Sites with the it k Nearest Neighbors Classifier.In *Ismb* (Vol. 5, pp. 147-152).

[135] Saadatfar, H., Khosravi, S., Joloudari, J. H., Mosavi, A., & Shamshirband, S. (2020). A new K-nearest neighbors classifier for big data based on efficient data pruning. *Mathematics*, *8*(2), 286.

[136] Tax, D. M., & Duin, R. P. (2004). Support vector data description. *Machine learning*, *54*(1), 45-66.

[137] Parvin, H., Alizadeh, H., & Minati, B. (2010). A modification on k-nearest neighbor classifier. *Global Journal of Computer Science and Technology*.

[138] Lau, K. W., & Wu, Q. H. (2003). Online training of support vector classifier. *Pattern Recognition*, *36*(8), 1913-1920.

[139] Elfatimi, E., Eryigit, R., & Elfatimi, L. (2022). Beans leaf diseases classification using MobileNet models. *IEEE Access*, *10*, 9471-9482.

[140] Kadam, K., Ahirrao, S., Kotecha, K., & Sahu, S. (2021). Detection and localization of multiple image splicing using MobileNet V1. *IEEE Access*, *9*, 162499-162519.

[141] Van Der Maaten, L., Postma, E. O., & Van Den Herik, H. J. (2009). Dimensionality reduction: A comparative review. *Journal of Machine Learning Research*, *10*(66-71), 13.

[142] Ishaque, M., & Hudec, L. (2019, May). Feature extraction using deep learning for intrusion detection system. In *2019 2nd International Conference*

on Computer Applications & Information Security (ICCAIS) (pp. 1-5). IEEE.

[143] Nair, R., & Bhagat, A. (2019). Genes expression classification using improved deep learning method. *International Journal on Emerging Technologies*, *10*(3), 64-68.

[144] Osama, S., Shaban, H., & Ali, A. A. (2023). Gene reduction and machine learning algorithms for cancer classification based on microarray gene expression data: A comprehensive review. *Expert Systems with Applications*, *213*, 118946.

[145] Ravindran, U., & Gunavathi, C. (2023). A survey on gene expression data analysis using deep learning methods for cancer diagnosis. *Progress in Biophysics and Molecular Biology*, *177*, 1-13.

[146] Talukder, M. A., Islam, M. M., Uddin, M. A., Akhter, A., Hasan, K. F., & Moni, M. A. (2022). Machine learning-based lung and colon cancer detection using deep feature extraction and ensemble learning. *Expert Systems with Applications*, *205*, 117695.

[147] Shi, J., & Luo, Z. (2010). Nonlinear dimensionality reduction of gene expression data for visualization and clustering analysis of cancer tissue samples. *Computers in biology and medicine*, *40*(8), 723-732.

[148] Liu, W., Pan, Y., Teng, Z., & Xu, J. (2024). scDMAE: A Generative Denoising Model Adopted Mask Strategy for scRNA-Seq Data Recovery. *IEEE Journal of Biomedical and Health Informatics*.

[149] Meng, Q., Lyu, Y., Peng, X., Xu, J., Tang, J., & Guo, F. (2024). EPIMR: Prediction of Enhancer-Promoter Interactions by Multi-Scale ResNet on Image Representation. *Big Data Mining and Analytics*, *7*(3), 668-681.

[150] Kandhro, I. A., Manickam, S., Fatima, K., Uddin, M., Malik, U., Naz, A., & Dandoush, A. (2024). Performance evaluation of E-VGG19 model: Enhancing real-time skin cancer detection and classification. *Heliyon*, *10*(10).

[151] Bakasa, W., & Viriri, S. (2023). Vgg16 feature extractor with extreme gradient boost classifier for pancreas cancer prediction. *Journal of Imaging*, *9*(7), 138.

[152] Babichev, S., Liakh, I., & Kalinina, I. (2024). Applying the deep learning techniques to solve classification tasks using gene expression data. *IEEE Access*.

[153] Ravindran, U., & Gunavathi, C. (2024). Cancer Disease Prediction Using Integrated Smart Data Augmentation and Capsule Neural Network. *IEEE Access*.

[154] Haznedar, B., Arslan, M. T., & Kalinli, A. (2017). Microarray gene expression cancer data. *Mendeley Data*, *2*, V4.

[155] Das, A., Neelima, N., Deepa, K., & Özer, T. (2024). Gene selection based cancer classification with adaptive optimization using deep learning architecture. *IEEE Access*.

[156] Sethi, B. K., Singh, D., Rout, S. K., & Panda, S. K. (2023). Long Short-Term Memory-Deep Belief Network based Gene Expression Data Analysis for Prostate Cancer Detection and Classification. *IEEE Access*.

[157] Bappi, J. O., Rony, M. A. T., Islam, M. S., Alshathri, S., & El-Shafai, W. (2024). A novel deep learning approach for accurate cancer type and subtype identification. *IEEE Access*.

[158] Thakur, T., Batra, I., Malik, A., Ghimire, D., Kim, S. H., & Hosen, A. S. (2023). RNN-CNN based cancer prediction model for gene expression. *IEEE Access*, *11*, 131024-131044.