11-5-2020

Genetic Algorithm based Convolutional Neural Network for Few Shot Learning in Disease type prediction on RNA-Seq Data

Kowshik Sharan Subramanian
University of Windsor

Genetic Algorithm based Convolutional Neural Network for Few Shot Learning in Disease type prediction on RNA-Seq Data

by

Kowshik Sharan Subramanian

A Thesis
Submitted to the Faculty of Graduate Studies
through the School of Computer Science
in Partial Fulfillment of the Requirements for
the Degree of Master of Science at the
University of Windsor

Windsor, Ontario, Canada
2020

Genetic Algorithm based Convolutional Neural Network for Few Shot Learning in Disease type prediction on RNA-Seq Data

by

Kowshik Sharan Subramanian

APPROVED BY:

M.Hlynka

Department of Mathematics and Statistics

S.Saad

School of Computer Science

J.Chen, Advisor

School of Computer Science

October 8, 2020

Declaration of Originality

I hereby certify that I am the sole author of this thesis and that no part of this thesis has been published or submitted for publication. I certify that, to the best of my knowledge, my thesis does not infringe upon anyone's copyright nor violate any proprietary rights and that any ideas, techniques, quotations, or any other material from the work of other people included in my thesis, published or otherwise, are fully acknowledged in accordance with the standard referencing practices. Furthermore, to the extent that I have included copyrighted material that surpasses the bounds of fair dealing within the meaning of the Canada Copyright Act, I certify that I have obtained a written permission from the copyright owner(s) to include such material(s) in my thesis and have included copies of such copyright clearances to my appendix. I declare that this is a true copy of my thesis, including any final revisions, as approved by my thesis committee and the Graduate Studies office, and that this thesis has not been submitted for a higher degree to any other University or Institution.

Abstract

Diagnosing the correct types of the disease is essential to the effective treatment. The diagnosis may not always be straightforward from the biological tests especially during the early stages of the disease. Human body responds to the disease by producing certain proteins. If we know which genes are active, that is, which proteins are being produced, we can more accurately classify disease subtypes. This study is based on the genetic information extracted from the patient's biological sample and is used to classify cancer subtypes. Among different types of genetic data, we consider RNA-seq data in this thesis. Studies based on genetic information often suffer from very limited samples and few shot learning has recently been studied for disease classification. Given the success of neural networks in assisting data analysis mostly with large amounts of data, we perform few shot learning by retraining the neural networks with genetic algorithmic processes. We follow the proposal from the Human Genome Organization (HUGO) to group genes based on their chemical composition and apply genetic algorithms to the HUGO gene groups to help retrain the neural networks. We apply our proposed approach to several different cancer datasets and compare our method across state-of-the-art methods. We have implemented our proposed approach and compared its performance with a wide variety of existing methods in machine learning and neural networks on three cancer datasets. According to our experiment, while performing similar to other methods when a relatively larger amount of data is available, our proposed approach outperforms Affinitynet by an average of 4 percent for few-shot learning with small datasets.

Dedication

I would like to dedicate this thesis to my family.

In memory of my father Subramanian Singaram

To my mother Umamaheswari Subramanian With love and eternal appreciation

Acknowledgements

There are many people whom I would like to acknowledge for their help and support during the course of working on my master thesis.

First and foremost I would pay my gratitude to my supervisor Dr. Jessica Chen. Under her guidance, I had enjoyed a lot working on my research work. It was a great pleasure to work and discuss with her. Without her support, this would not have been possible.

In addition to this, many thanks to my committee members Dr. Sherif Saad Ahmed and Dr. Myron Hlynka for their valuable time and their support. I would like to express my appreciation to Mrs. Karen Bourdeau, Mrs. Melissa Robinet and Mrs. Christine Weisener who always supported me when I needed assistance in various academic issues.

I would like to thank my mother for her counsel and her sympathetic ear and I would like to thank my grandfather for his support in my education. Next, I am also very thankful to my friends for their advice, moral support and for listening to my problems for long hours.

Kowshik Sharan Subramanian

Contents

List of Tables

List of Figures

Chapter 1

Introduction

Disease subtype classification is a problem that focuses on classifying diseases into their respective subtypes. Several methods have been proposed for disease subtype classification. Recent methods involve identifying the disease subtype based on the genetic information collected from the patient's blood. Since these diseases might not be common and data collection studies usually involve few patients, it is a very practical scenario to receive limited amount of patient data. However most of the methods do not address this data deficiency problem in genetic data analysis. Although existing methods such as convolutional neural networks can perform much better when enough data is provided, such amounts of data is not available for disease prediction. Even the methods that do, does not use domain knowledge to improve performance. Hence we tackle disease subtype classification by using domain knowledge in genetic algorithm-based convolutional neural networks.

1.1 Background

1.1.1 Precision Medicine

The Precision Medicine Initiative[9] is a long-term research endeavor, involving the National Institutes of Health (NIH) of the United States of America and multiple other research centers, which aims at understanding how a person's genetics, environment, and lifestyle can help determine the best approach to prevent or treat disease.

According to the Precision Medicine Initiative, precision medicine is "an emerging approach for disease treatment and prevention that takes into account individual variability in genes, environment, and lifestyle for each person." This approach will allow doctors and researchers to predict more accurately which treatment[3] and prevention strategies for a particular disease will work in which groups of people. It is in contrast to a one-size-fits-all approach, in which disease treatment and prevention strategies are developed for the average person, with less consideration for the differences between individuals.

The Precision Medicine Initiative has both short-term and long-term goals. The short-term goals involve expanding precision medicine in the area of cancer research. Researchers at the National Cancer Institute (NCI)[1] of the United States of America hope to use an increased knowledge of the genetics and biology of cancer to find new, more effective treatments for various forms of this disease. The long-term goals of the Precision Medicine Initiative focus on bringing precision medicine to all areas of health[14] and healthcare on a large scale. To this end, the NIH has launched a study, known as the All of Us Research Program, which involves a group (cohort) of at least 1 million volunteers from around the United States. Participants are providing genetic data, biological samples, and other information about their health. To encourage open data sharing, participants can access their health information, as well as research that uses their data, during the study. Researchers can use these data to study a large

range of diseases, with the goals of better predicting disease risk, understanding how diseases occur, and finding improved diagnosis and treatment strategies.

Although the term "precision medicine" is relatively new, the concept has been a part of healthcare for many years. For example, a person who needs a blood transfusion is not given blood from a randomly selected donor; instead, the donor's blood type is matched to the recipient to reduce the risk of complications. Although examples can be found in several areas of medicine, the role of precision medicine in day-to-day healthcare is relatively limited. Researchers hope that this approach will expand to many areas of health and healthcare in coming years.

1.1.2 Disease Prediction

Disease prediction is the first outcome of the precision medicine initiative where the disease is predicted before the physical symptoms manifest. This is done by analysing the genetic changes in the body in response to the disease. The genetic expression in the body is tracked and any fluctuation is directly correlated to the disease. This is done to assist in the diagnosis of the disease. This process also acts as the first step of the precision medicine initiative to create custom gene-based cures for diseases.

As this is primarily used for cancer, each cancer type has several subtypes based on the location of the metastasis. Identifying the subtype of cancer accurately is essential for proper treatment. Hence classification among the different subtypes is essential and it can be done by analysing the genetic information in the blood. For Example, let us consider Kidney cancer. Based on the type of cell being affected, it is classified into three subtypes.

- Chromophobe Carcinoma - it affects the chromophobe cells in the kidney.
- Papillary Carcinoma - it affects the papillary cells in the kidney.
- Clear Cell Carcinoma - it affects the clear cells in the kidney.

Since the method for treatment for each type differs, correctly identifying the type of cancer is essential for proper treatment.

1.1.3 Few-shot Learning

Few-shot learning is an abstract term which involves processes and methods which are used to train machine learning and deep learning models with as little data as possible. This few-shot learning approach is used in cases where there is a lack of data such as in disease classification and prediction studies or used for class labelling when there is less amount of labelled data such as in image data.

1.1.4 Convolutional Neural Network

Neural networks are computational constructs that mimic the biological neural system in order to learn and identify inherent patterns in data. These patterns in turn help us to predict variables or classify them.

Convolutional Neural Network (CNN) is a type of Neural Network that is efficient in performing classification. The first layer of the convolutional neural network is the input layer which contains as many neurons as the number of features sent as input into the CNN. After the input layer we add one or more convolutional layers and pooling layers. In these layers, there are filters that summarize the data and output summarized vectors. Finally, it consists of one or more dense layers and has an output layer. The dense layer consists of different types of neurons for learning. The output layer consolidates the learning and outputs the final summarized vector with the weights. Our approach utilizes the retraining of the CNN with genetic operations for improved performance in few-shot learning

1.1.5 Genetic Algorithms

In computer science, Genetic Algorithm (GA) is a metaheuristic inspired by the process of natural selection based on the concept of Darwin's theory of evolution for solving both constrained and unconstrained optimization problems. Genetic algorithm repeatedly modifies a population of individual solutions by selecting the best individuals from the current population as parents and using them to produce children for the next generation through a number of bio-inspired operators such as mutation, crossover and selection. Over successive generations, the population evolves towards an optimal solution. As the training process of a CNN is basically an optimization problem, we use genetic algorithm towards this end.

Population

In a genetic algorithm, a population of solutions to an optimization problem is evolved towards better solutions. The population can be sets of data, solution functions, etc.

Chromosome

Each solution, encoded with real or binary numbers or some other encoding, is represented by a chromosome which can be changed and altered. Chromosomes are essentially the members of the population.

Evolution

The evolution, which is an iterative process, usually starts from a population of randomly generated individuals, with the population in each iteration called a generation.

Fitness Function

In each generation, the fitness measurement of each and every individual in the population is evaluated, which is usually the value of the objective function in the opti-

mization problem being solved. The more fit individuals are selected from the current population to be modified by genetic operators for the formation of a new generation. The new generation of candidate solutions is then used in the next iteration of the algorithm. Commonly, the algorithm terminates when either a maximum number of generations has passed, or a satisfactory fitness level has been reached for the population. The fitness level can be any evaluation metric or any threshold value.

A typical genetic algorithm requires a genetic representation of the solution domain and a fitness function to evaluate the solution domain. A standard representation of each solution is by using an array. The main property that makes these genetic representations convenient is that its parts are easily aligned due to their fixed size, which enables simple crossover operations. Once the genetic representation and the fitness function are defined, the algorithm proceeds to initialize a population of solutions and then improving it through repetitive application of the selection, crossover, and mutation operators. Next we describe the flow of how a genetic algorithm generally operates.

1. Initialization: The initial population contains several hundreds or thousands of possible solutions often generated randomly. This allows a desirable sampling of the entire range of possible solutions.

2. Evaluation: A fitness function is defined over the genetic representations of the solutions to assign a fitness value to each member based on measuring the quality of the represented solution according to some performance evaluation criteria. The fitness function is problem dependent, henceforth making its correct determination an important step in the configuration of a working genetic algorithm.

3. Selection: During each successive generation, whole or a portion of the current population is selected to reproduce a new generation. Solutions involved in the

reproduction are selected through a fitness-based process, where fitter solutions are typically more likely to be selected. Some examples of selection strategies are fitness-proportion selection (or most commonly known as roulette-wheel selection), tournament selection, and stochastic universal sampling.

4. Genetic Operators: The next step is to create a next generation population of solutions from the selected solutions through a combined use of genetic operators such as crossover and mutation. For the production of each new solution, a pair of "parent" solutions (or sometimes, just one solution) is selected from the pool of previously selected fit individuals for breeding. By producing a "child" solution using the mentioned methods of breeding, a new solution is created which mostly shares many of the characteristics of its "parent(s)". Again, new parent(s) are selected for each new child, and this process continues until a new population of solutions of appropriate size is generated. In order to produce a new generation of solutions, this newly bred children can replace some or the entirety of the members of a current generation. These processes ultimately result in the next generation population of chromosomes that is different from the previous generation. Generally, the average fitness of a population increases by this procedure, since only the best organisms, along with a small proportion of lesser fit solutions, are selected for breeding. These less fit solutions ensure genetic diversity within the genetic pool of the parents and therefore, ensure the genetic diversity of the subsequent generation of children. Although crossover and mutation are known as the main genetic operators, it is possible to use other operators such as regrouping, colonization-extinction, or migration in genetic algorithm. To find a reasonable configuration of settings for the problem class being worked on, parameters such as the mutation probability, crossover probability and population size must be tuned properly. A very small mutation rate may lead to genetic drift. A recombination rate that is too high may lead

to premature convergence of the genetic algorithm. A mutation rate that is too high may lead to loss of good solutions. So, the correct configuration of tuning these parameters is an important step towards making a good genetic algorithm.

5. Termination: The above process is repeated until some termination condition has been reached. Common terminating conditions are finding a solution that satisfies minimum criteria, reaching a fixed number of generations, or reaching a plateau in the highest ranking solution's fitness.

1.2 Problem Definition

The lack of genetic data prevents widespread analysis and prediction of diseases and their subtypes. The problem considered in this thesis is to perform classification of disease subtypes with the limited amount of data accurately.

1.3 Thesis Motivation

Diseases in general cannot be diagnosed until the symptoms manifest in the body. Recent studies have confirmed that even before symptoms manifest, the signs of the disease can be found by observing the genetic changes in the body such as gene expression. This is the key to predicting diseases and for genetic medicine. Also, several diseases have different subtypes which can alter the way of treatment. So accurate diagnosis is important for proper treatment. Fast and accurate diagnosis can help the treatment and recovery in a long way.

Disease prediction at the current level is only used in prediction and diagnosis. In the future this can be used to tailor genetic medicines which can improve specific

bodily immune functions and greatly reduce the side effects thus approaching illness and recovery in a new manner. The cures are also specifically tailored for each individual's needs and not as a blanket cure. This has already been proposed in the precision medicine plan in several countries.

In this initiative, several research works attempt to properly track genetic activity. The problem with such an analysis is that there is scare availability of data for each disease, so work must be done to train models with less amounts of data. This process is called as few shot learning and has been addressed by some previous works[28]. The identification of cancer types is essential for proper treatment, hence previous works focus on cancer type classification. The limitation of the existing work is that it does not attempt to incorporate domain knowledge in the model. It also has a range where the performance can be improved.

The limitations of the existing work motivated the author to research and propose a new architecture which can alleviate the existing problems. This research aims to propose an architecture that performs few shot learning and that utilizes domain knowledge(gene grouping) to improve the performance of the classifier.

1.4 Thesis Statement

The objective of this research is to propose a new approach for few shot learning such that inherent domain knowledge can be used to improve the performance of a classifier. Our thesis attempts to introduce domain knowledge in the form of gene grouping and exploit the grouping with genetic algorithms to create a few shot training architecture

Our proposed approach is to use domain knowledge to improve the few-shot learning capability of the existing approach. Since the existing approach uses a convolutional neural network and modifies its architecture, we also modify it in such a way that it utilizes the domain knowledge. The domain knowledge is incorporated in the form of gene grouping which was proposed by the Human Genome Organization[4] based on the molecular structure of each gene.

We solve the few-shot learning problem by proposing a new algorithm that utilizes domain knowledge.

1.5 Thesis Contribution

This thesis addresses the lack of data problem in genetic disease prediction and proposes a novel architecture that improves the performance of cancer subtype classification when less amounts of data are available. The proposed algorithm utilizes genetic algorithm concepts to retrain convolutional neural networks to prevent overfitting and selects the best model for classification.

The proposed GACNN model consists of three distinct variations to previous work which makes it better for solving the problem at hand. Each step is designed to address the limitations of the existing algorithm.

- Genetic Algorithms: The GA is used to optimize the solution and to introduce variations to prevent overfitting and to select a fit model.
- Convolutional neural networks: The CNNs are repeatedly retrained to familiarize similar features and make the distinct features stand out.
- Gene grouping: This is a highly unique step as previous works do not consider gene grouping for consolidation. This imparts an inherent grouping which can be exploited using genetic algorithms.

1.6 Thesis Organization

The rest of the thesis/research work is organized in the following manner.

In chapter II, we discuss related work/literature review in the field of cancer subtype classification , classification methods, genetic algorithms, types of genetic operations and other disease type classification methods

In chapter III, We introduce our proposed few shot GACNN which is a novel few shot algorithm that works with genetic data such as mRNA data. It also acts as a self regulatory algorithm which prevents overfitting and chooses the best network. In this chapter, we give detailed description of the GACNN and how it integrates gene grouping to give accurate classification results.

Chapter IV, We explain our experimental setup. This chapter presents the details of tested environmental setups, details of tested datasets and evaluation methods. It also presents the test results of the GACNN and compared the GACNN performance on four different types of cancer datasets. This chapter consists of detailed representation of AMI scores for each proportion of the dataset

Chapter V concludes the research, explains insights received during the work and sets up the field of opportunities for the future work.

Chapter 2

Related Work and Literature Review

This chapter focuses on related works and researches used for background study, concept-building, and theoretical background of our thesis. We discuss and analyze works of literature that are relevant to the few shot problem specific to the disease prediction paradigm

2.1 Disease Subtype Classification methods

Based on the background study, related works, and limitations of the previous works, we propose a novel approach for few shot learning in RNA-Seq data. We describe our proposed model in detail in Chapter 3.

Disease classification has become an active research field ever since the announcement of the Precision medicine initiative[9] and has become more active since the advent of the COVID 19 pandemic. Genetic disease prediction works on the premise that any kind of disease, before it shows any physical symptoms must show changes in the genetic content in the blood. This helps in prognosis of the disease. Diseases

also have subtypes based on the area affected and other factors which also affects the method of treatment. We consider cancer and its various subtypes for our study because accurate diagnosis of the disease is essential for proper treatment.

The recent work about cancer subtype classification by Tianle ma et al [28] addresses a core issue in genetic disease prediction. It involves addressing the lack of data by using few shot learning. We have chosen this paper as our base and have addressed the issues mentioned in it and have attempted to improve on it. This paper describes the problem of lack of data in genetic disease prediction and proposes a few shot semi supervised learning model called as "AffinityNet"[27] which performs better with less amount of data. This paper utilizes kNN attention pooling to group similar features and emphasize on distinct features. They also incorporate the attention pooling in the convolutional neural network. Neural networks tend to work better with more data. Hence attention pooling is incorporated to compensate for the lack of data. This step is to incorporate few shot learning into the convolutional neural network. Although this paper is comprehensive in addressing the few shot problem using a convolutional neural network, it does not use domain knowledge to improve few shot learning or optimize it. This has been mentioned in its future work, which is what we are trying to address. We implement the algorithm in the paper in conjunction with our dataset in order to compare the performance of our algorithm with affinitynet.

Other similar work regarding cancer subtype classification[45][17][44] include the work by Pooya Mobaderasany, Lee ad Cooper et al.[30], in which they use tissue biopsy images to predict and classify disease subtypes. Although this is a much accurate method, this involves the utilization of an invasive procedure and the data for tissue biopsy is much harder to retrieve than the genetic data which is retrieved from blood and hence we go with RNA-seq data which can be easily gathered from blood samples rather than from surgery.

Ekaterini Blaveri et al.'s[2] work on bladder cancer subtype classification takes a similar problem and tries to solve it. Instead of m-RNA, they use cDNA dataset for classification. they also use conventional supervised machine learning methods rather than neural networks to solve the problem. They also incorporate the cherry-picking strategy which involves manually studying the predictor genes to see if they can actually be helpful for classification and prediction. Other cancer identification, prognosis, classification and class identification methodologies have also been widely explored in[19][7][41][20][21][42]

Class discovery methods proposed by Golub et al., [15] is essential for determining the classes in cancer. The author proposed a method to determine when a variation of a cancer can be classified as a subtype such that it warrants a different treatment procedure. Class discovery is performed till today when new findings about cancer are released and the cancer subtype classification problem evolves further.

2.2 Computational Classification Methods

The computational method of optimizing a convolutional neural network using genetic algorithms was recently proposed by Parsa Esfahanian and Mohammad Akhavan[13] in their work. They have implemented the genetic algorithm as a layer to improve training speed and optimization but does not address the issue of few shot learning which we address in our method.

Naive Bayes classifier[47] is a prominent machine learning method used for classification. It utilizes conditional probabilities to classify the dataset. Since it is a tried and tested method, it serves as a benchmark for classification.

Random forest algorithm[5] emerged after bagging and boosting techniques as a dominant classification method. It is a combination of tree predictors such that each tree depends on the values of a random vector sampled independently and with the

same distribution for all trees in the forest.Since it is predominantly used for classification and has consistently given reliable scores, we take it for benchmarking. The work by Stephane Wenric et al.[43] uses random forest algorithm on the same TCGA dataset which we use and show good accuracy scores. They perform differential expression analysis in order to identify the genes which are good indicators of cancer and indicators of survival.

Convolutional neural networks[22][18] are predominantly used for image classification. It involves sliding a filter across an image or matrix to summarize the values such that the final layer contains a feature representation of the original data. Since it gives a good representation of the input data, retraining it with similar features in a single group would make the model sensitive to unique features in the data.

2.3 Genetic Algorithms

Genetic algorithm repeatedly modifies a population of individual solutions by selecting the best individuals from the current population as parents and using them to produce children for the next generation through a number of bio-inspired operators[25] such as mutation,crossover and selection. There are different ways of performing such genetic operations. For example the work by Survana Patil et al. [32] details on the types of mutation operations that can be done and for what problems each mutation strategy works the best. It describes mutation methods such as dynamic mutation , schema mutation, compound mutation, cluster based adaptive mutation and hyper mutation. We use schema mutation in our approach where we perform mutation whenever the fitness value stabilizes.

For crossover[39], the work by Umbarker et al.[40] describes various crossover strategies used for the crossover operation in a genetic algorithm. They list methods such as 1-point crossover, K-Point crossover, shuffle crossover, reduced surro-

gate crossover, uniform crossover, average crossover, discrete crossover, flat crossover, heuristic crossover, elitist crossover and other multivariate crossover techniques. We use elitist crossover technique to perform crossover.

There have also been several instances where genetic algorithms have been used for optimizing neural networks[36]. [29] The work by Paulito Palmes et al. [31] uses mutation to introduce variation in order to prevent overfitting in neural networks. They are used in optimizing the weights[11][46], the intermediate values, the training rate and every other aspect of the neural network

2.4 Few Shot Learning Methods

Few-shot learning involves using less amounts of data to train algorithms. Some commonly used few-shot learning methods are described as follows.

The prototypical networks[37] for few- shot learning work by Jake Snell et al., depicts a method of few shot learning which involves using distances between the prototypes of classes and the actual classes using softmax and SGD. It is a simplistic approach that is essential in reducing the complexity in image data, but does not work well with RNA-Seq data.

Another approach involves meta Learning[34] which uses meta data to perform few- shot learning. Although this method has been proposed for semi supervised labelling of images using meta data, this method has a larger scope in terms of few shot learning for genetic data, as plenty of meta data will be available for few-shot learning in patient samples. This approach has not yet been explored for few shot learning using genetic data.

The work by [33] Sachin Ravi et al. implements an LSTM in a deep learning neural network which is modified for few-shot learning. They also use a modified form of meta learning for few-shot learning by automatically self-adjusting weights.

The work by Yanbin Liu et al. [26] uses a transducive propagation network for few-shot learning. The model performs good with image datasets. It exploits feature embeddings and transfer learning to similar datasets to perform few-shot learning.

The work by Boyang Deng et al. [12] implements few-shot learning in Convolutional neural networks This few-shot learning method is implemented on image dataset. It works by training the algorithm on a well known visual dictionary and then transferring the weights to try and classify unlabelled images. They also incorporate some unsupervised clustering for class discovery.

Chapter 3

Proposed Approach

3.1 Introduction

As discussed in Chapter 1, cancer subtype classification[24] is important for diagnosis and proper treatment. According to the precision medicine initiative, we use genetic data in order to analyze and classify cancer subtypes. But the problem is the lack of sufficient samples available for analysis. Hence we resort to few-shot learning techniques. We try to incorporate the available data in such a manner that accurate classifiers can be trained. We also introduce genetic algorithms to help with introducing variations in the available data to enable training the convolutional neural network and to prevent overfitting.

3.2 Architecture

The architecture of our proposed approach is depicted in figure 3.1. It shows how the data is split by incorporating domain knowledge, and how the CNNs are trained with genetic algorithms to select the optimal model and the optimal featureset. Our proposed approach has four major steps that define it. They are

1. Gene Grouping
2. Convolutional Neural Network
3. Genetic Operations
4. Model Selection

These steps ensure that few shot learning is performed with the use of domain knowledge in our proposed approach.

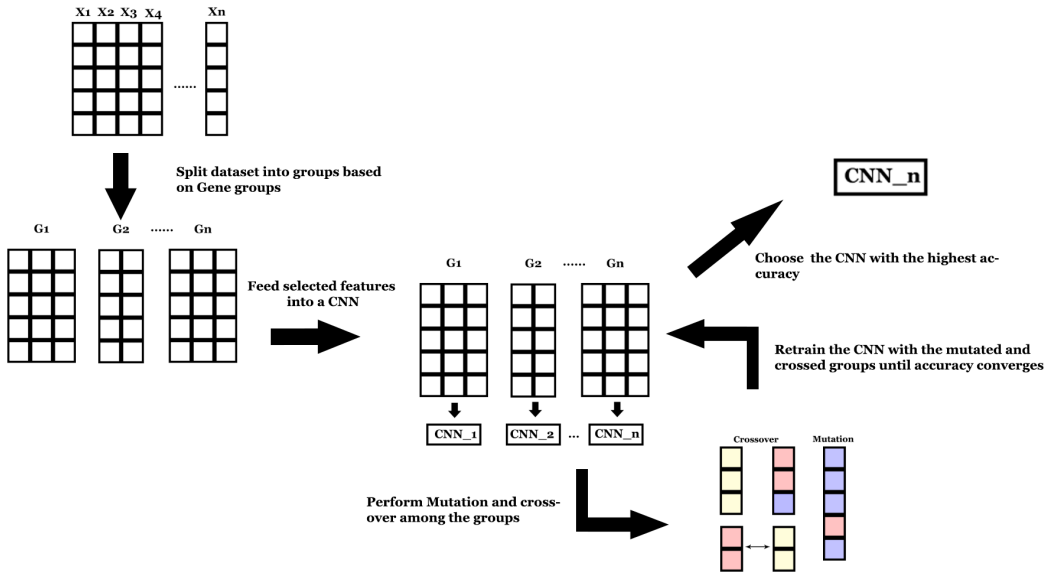

Figure 3.1: Architecture of the proposed approach

3.3 Gene grouping

A gene group[4] is a set of genes that share important characteristics. In many cases, genes in a group share a similar sequence of DNA building blocks (nucleotides). These genes provide instructions for making products (such as proteins) that have a similar structure or function. In other cases, dissimilar genes are grouped together because proteins produced from these genes work together as a unit or participate in the same process.

HGNC ID (gene)	Approved symbol	Approved name	Previous symbols	Aliases	Chromosome
HGNC:324	AGPAT1	1-acylglycerol-3-phosphate O-acyltransferase 1		LPAAT-alpha	6p21.32
HGNC:325	AGPAT2	1-acylglycerol-3-phosphate O-acyltransferase 2	BSCL	LPAAT-beta	9q34.3
HGNC:326	AGPAT3	1-acylglycerol-3-phosphate O-acyltransferase 3		LPAAT-gamma, LPAAT3	21q22.3
HGNC:20885	AGPAT4	1-acylglycerol-3-phosphate O-acyltransferase 4		LPAAT-delta, dJ473J16.2	6q26
HGNC:20886	AGPAT5	1-acylglycerol-3-phosphate O-acyltransferase 5		FLJ11210, LPAAT-e, LPAAT-epsilon	8p23.1
HGNC:20880	GPAT4	glycerol-3-phosphate acyltransferase 4	AGPAT6	DKFZp586M1819, LPAAT-zeta, TSARG7	8p11.21
HGNC:30059	LPCAT4	lysophosphatidylcholine acyltransferase 4	AYTL3, AGPAT7	FLJ10257, LPAAT-eta, LPEAT2	15q14
HGNC:26756	LCLAT1	lysocardiolipin acyltransferase 1	LYCAT	FLJ37965, ALCAT1, AGPAT8	2p23.1
HGNC:28157	GPAT3	glycerol-3-phosphate acyltransferase 3	AGPAT9	MGC11324, LPAAT-theta, MAG1, HMFN0839, AGPAT10	4q21.23
HGNC:25718	LPCAT1	lysophosphatidylcholine acyltransferase 1	AYTL2	FLJ12443, AGPAT9, AGPAT10	5p15.33
HGNC:26032	LPCAT2	lysophosphatidylcholine acyltransferase 2	AYTL1	FLJ20481, AGPAT11, LysoPAFAT	16q12.2

Figure 3.2: Example of a Gene group AGPAT

Classifying individual genes into groups helps us describe how genes are related to each other. Researchers can use gene groups to predict the function of newly identified genes based on their similarity to known genes. Similarities among genes in a group can also be used to predict where and when a specific gene is active (expressed). Additionally, gene groups may provide clues for identifying genes that are involved in particular diseases. Sometimes not enough is known about a gene to assign it to an established group. In other cases, genes may fit into more than one group. No formal guidelines define the criteria for grouping genes together. Classification systems for genes continue to evolve as scientists learn more about the structure and function of genes and the relationships between them. Hence exploring the gene grouping would help us in identifying key indicators for a disease subtype[38]. Since some groups have less than ten features, we rearrange the groups such that each group has more than

50 features. We get a total of 113 groups after consolidation. The figure below shows the distribution of the features across the groups.

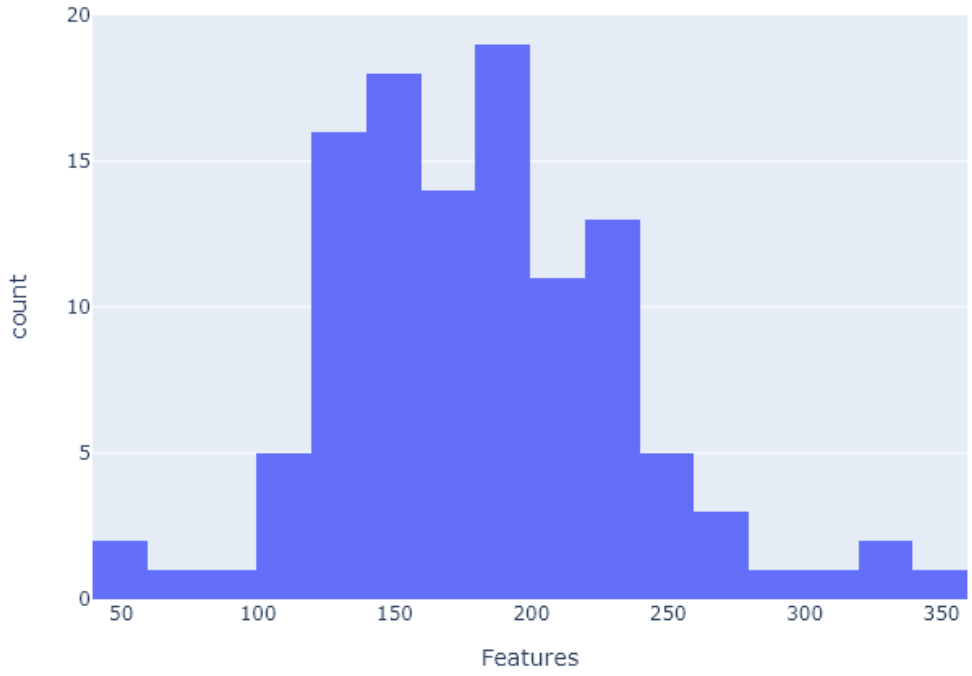

Figure 3.3: Histogram of number of features in groups

3.4 Genetic Operations

Genetic algorithm consists of genetic operations which are recursively performed to introduce variation and to generate a new population. The two genetic operators which we are going to implement in our approach are mutation and crossover.

3.4.1 Mutation

Let us consider the gene groups $G_1, G_2 \ldots G_n$ and let us consider the set of features in the group as G_1 as $G_{11}, G_{12}, G_{13}, G_{14}, \ldots, G_{1n}$

The top N groups are chosen based on the Adjusted Mutual Information score

and they are randomly paired with each other. We create a mutation pool M which contains the features that were not selected during the feature selection process.

For performing mutation, any feature is randomly chosen in the group and in the mutation pool and the chosen feature in the group is replaced with the random feature in the mutation pool.

Before Mutation

$G_1 = \{G_{11}, G_{12}, G_{13}, G_{14}, \ldots, G_{1n}\}$

$M = \{M_1, M_2, M_3, M_4, \ldots, M_n\}$

After Mutation

$G_1 = \{G_{11}, G_{12}, G_{13}, M_3, \ldots, G_{2(n/2)}, \ldots . G_{2n}\}$

We perform mutation by following the schema mutation methodology where mutation is performed whenever the fitness function gets stuck in a local optima

3.4.2 Crossover

Lets us consider the gene groups consisting of groups $G_1, G_2, \ldots, G_n$. The top N groups are chosen based on the AMI score and they are randomly paired with each other. Let us consider a pair of groups G_1, G_2. For performing crossover, any half of the features are chosen in random in the first group and they are exchanged with corresponding set of features in the second group of the pair. The crossover operation is depicted in figure 3.4.

Before Crossover

$G_1 = \{G_{11}, G_{12}, G_{13}, G_{14}, \ldots, G_{1(n/2)}, \ldots, G_{1n}\}$

$G_2 = \{G_{21}, G_{22}, G_{23}, G_{24}, \ldots, G_{2(n/2)}, \ldots, G2n\}$

After Crossover

$G_1 = \{G_{11}, G_{12}, G_{13}, G_{14}, \ldots, G_{2(n/2)}, \ldots, G_{2n}\}$

$$G_2 = \{G_{21}, G_{22}, G_{23}, G_{24}, \ldots, G_{1(n/2)}, \ldots, G_{1n}\}$$

We perform crossover by following the elitist crossover strategy where the top N groups are selected and crossover is performed among them.

The crossover operation performed in our approach is presented in the figure below

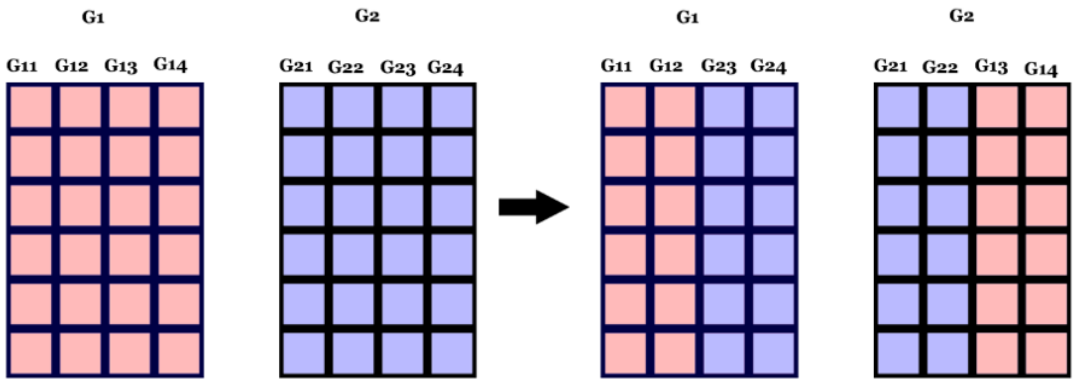

Figure 3.4: Crossover operation performed in the proposed approach

3.5 Convolutional Neural Network

Neural networks are computational constructs that mimic the biological neural system in order to learn and identify inherent patterns in data. These patterns in turn help us to predict variables or classify them.

Convolutional Neural Network (CNN) is a type of Neural Network that is efficient in performing classification. The first layer of the convolutional neural network is the input layer which contains as many neurons as the number of features sent as input into the CNN. After the input layer we add one or more convolutional layers and pooling layers. In these layers, there are filters that summarize the data and output summarized vectors. Finally, it consists of one or more dense layers and has an output layer. The dense layer consists of different types of neurons for learning. The output layer consolidates the learning and outputs the final summarized vector

with the weights. An example CNN is illustrated with its different layers in Figure 3.5.

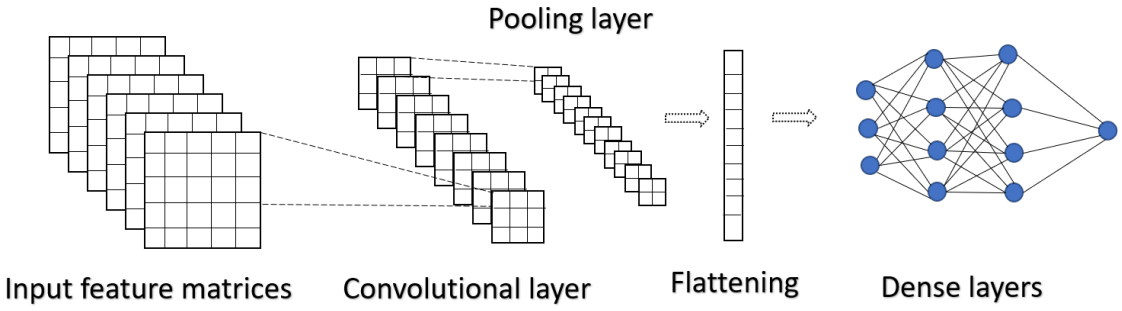

Figure 3.5: Convolutional Neural Network

The convolutional layer is special as it used to extract features from input matrices. Neurons in this layer are positioned as two-dimensional arrays and referred to as the activation map. Weights are arranged in a three-dimensional array known as the kernel. Height, width, and depth might be changed for different input sizes. There can be multiple kernels with different sizes, even for the same input matrix. The kernel slides through the input matrix and creates a set of activation maps.

Convolutional layers reduce the weight optimization problem as they extract spatial features and feed them to neurons rather than connecting one neuron to each cell for the input matrix. In addition to the weights in kernels, a bias value is also added while creating an activation map. The activation value calculation by convolution layer can be expressed as follow in the equation 3.1

$$x_{j,k}^{l} = \sigma\left(y^{l-1} + \sum_{m=0}^{p}\sum_{n=0}^{q} w_{m,n}^{l-1} x_{j+m,k+n}^{l-1}\right) \tag{3.1}$$

where $x_{j,k}^{l}$ is the result activation value of the k^{th} neuron in the j^{th} row of the l^{th} layer, y is the bias shared among the layer, w is the weight parameter of the $p \times q$ size kernel, and $\sigma(z)$ is the activation function.

The pooling layer is usually added just after the convolutional layer. Pooling layer reduces dimensionality but does not reduce depths or channels. Convolutional layer outputs feature maps as patches. Pooling layer kernels slides through these patches and creates reduced and summarized feature maps. There are three types of kernels used in CNN, which are Max Pooling, Min pooling, and Average pooling. Max pooling takes maximum values of the window while filter slides; Min pooling takes minimum value of the window while filter slides, and Average pooling takes an average of all value in the window while filter slides through. The most common type of pooling used in CNN is Max pooling. There can be multiple sets of convolutional layers and pooling layers. After the convolutional layer and pooling layer, the resulting matrix is flattened and fed into a dense layer, which is usually a multi-layer perceptron network. Finally, the output layer will perform classification based on the summarized weights. Our proposed approach uses a convolutional network in the core for classification. But for the training process, we use multiple CNNs and we use a fitness score(AMI score) to select the best CNN after the required number of generations are elapsed. This ensures that every feature is given a fair chance in training and enables the genetic fitness function to work optimally. We did not have a situation where the fitness score does not not converge, so we did not apply for any steps to improve convergence. Whenever the AMI scores stabilize, we introduce mutation and crossover until further changes does not positively affect the score.

3.6 Evaluation Metrics

3.6.1 Adjusted Mutual Information

AMI is an adjustment of the Mutual Information (MI) score to account for chance, which is suitable to measure the performance of clustering and classification[28] with multiple unbalanced classes (AUROC is a similar metric but is mainly suitable for

binary classification). AMI score is good for evaluating unbalanced classes which is the case with the dataset we use for this thesis. We also measure accuracy, precision, recall and F1 scores, but as they were unreliable in accurately measuring the few-shot learning potential, they are not considered for this thesis.

3.6.2 Performance Comparison

In order to make sure that our proposed approach is better than the existing methods of few-shot learning, we choose a wide variety of existing methods in machine learning and neural networks to compare with our proposed approach. We also select the approach of 'AffinityNet' which has been proposed by the base paper of this thesis and compare it with our method. The list of methods used for comparison are

- Multinomial Naive Bayes
- Random Forest
- Support Vector Machines
- Neural Networks
- AffinityNet

Multinomial Naive Bayes

Naive Bayes[47] is a machine learning classifier that uses probability. It is based on the Bayes theorem on conditional probability with the "naive" assumption of conditional independence between every pair of features given the value of the class variable. Despite its simplistic assumptions, it is a pretty good classifier and is used as a benchmark for new methods. According to Bayes rule, the probability of an example $E = (x_1, x_2, ..., x_n)$ being class c is

$$p(c|E) = \frac{p(E|c)p(c)}{p(E)}$$

E is classified as the class $C = +$ if and only if

$$f_b(E) = \frac{p(C=+|E)}{p(C=-|E)} \geq 1,$$

where $f_b(E)$ is called as a Bayesian classifier.

Assume that all attributes are independent given the value of the class variable

$$p(E|c) = p(x_1, x_2, ..., x_n|c) = \prod_{i=1}^{n} p(x_i|c)$$

the resulting classifier is then:

$$f_{nb}(E) = \frac{p(C=+)}{p(C=-)} \prod_{i=1}^{n} \frac{p(x_i|C=+)}{p(x_i|C=-)} \tag{3.2}$$

where f_{nb} is the Naive Bayes classifier

Since we have multiple classes, we use multinomial Naive Bayes classifier for classification.

Random Forest

Random forests[5] are a combination of tree predictors such that each tree depends on the values of a random vector sampled independently and with the same distribution for all trees in the forest. The generalization error for forests converges as the number of trees in the forest becomes large. The generalization error of a forest of tree classifiers depends on the strength of the individual trees in the forest and the correlation between them. Using a random selection of features to split each node yields error rates that are favourable and robust with respect to noise.

A random forest is a classifier consisting of a collection of tree-structured classifiers $h(x, \Theta_k), k = 1, ...$ where the Θ_k are independent identically distributed random vectors and each tree casts a unit vote for the most popular class at input x.

Hence, we chose the random forest algorithm to compare performance in classification.

Support Vector Machine

A support vector machine constructs a hyper-plane or set of hyper-planes in a high or infinite dimensional space, which can be used for classification, regression or other tasks. Intuitively, a good separation is achieved by the hyper-plane that has the largest distance to the nearest training data points of any class (so-called functional margin), since in general the larger the margin the lower the generalization error of the classifier. LinearSVC implements "one-vs-the-rest" multi-class strategy[10], thus used for multi-class classification.

Neural Network

Neural networks[16] are computational constructs that mimic the biological neural system in order to learn and identify inherent patterns in data. These patterns in turn help us to predict variables or classify them. Since we use convolutional neural networks, in order to bench mark against traditional neural networks, we choose a simple ten layer neural network and test it with our data.

AffinityNet

AffinityNet[28] is the approach that is described in the base paper of our thesis. It involves using semi supervised clustering and attention pooling layers in a convolutional network in order to optimize the model for few shot learning. We use this in order to benchmark our proposed approach.

Chapter 4

Experiments, Discussions, Comparisons and Analysis

In this chapter, we explain the details of the experimental setup and environments such as tools and libraries used to implement the few shot GACNN, systems configurations of data pre-processing step and CNN training and testing, dataset details, the detailed configuration of CNN and layer details, train and test details and evaluation method to test the results. We then present the results of the conducted various tests. We recorded our proposed GACNN's performance against machine learning methods such as Naive Bayes classifier, SVM classifier, random forest classifier. We also compare it against a traiditional 10 layer neural network. Finally we compare our approach to the method proposed in the base paper which is the 'AffinityNet' method. We conducted our experiments on three different types of cancer datasets with the experimental setup explained in Chapter 3. We depict the nature of each dataset by using Principal component analysis and represent it using scatterplots.

We illustrate the performance of all compared methods on each dataset as bar charts. Charts plots average AMI score of all methods on 10 test runs on each dataset. We verify the effectiveness of the methods in few-shot learning by utilizing

a fraction of the dataset in each method.

4.1 Tools and libraries

We implemented our GACNN using Keras[8] with the Python 3.6 programming language. The libraries and their versions that are used to implement our proposed approach are listed below

- NumPy 1.17.1
- SciPy 1.3.1
- Pandas 0.25.1
- Tensorflow GPU 1.14.0
- Keras 2.31
- Plotly 4.10.0

We used VS Code IDE to implement and test the GACNN.

4.2 System Configurations

Data pre-processing and GACNN training was created using a google cloud cluster with the specification of Intel Xeon Scalable Processor (Cascade Lake), 256 GB of RAM.

4.3 Datasets

The proposed GACNN approach aims to classify cancer subtypes using RNA Seq data.

- TCGA Kidney Cancer RNA Seq dataset[35] (Firehose Legacy). This dataset consists of datasets of three different types of kidney cancer specimens which are Chromophobe, clear cell and papillary Carcinoma and consists of a total of 891 samples.

- TCGA Lung Cancer RNA Seq dataset[6][23] (Firehose Legacy). About 80 percent to 85 percent of lung cancers are Non Small Cell Lung Cancer (NSCLC). The main subtypes of NSCLC are adenocarcinoma and squamous cell carcinoma and consists of 1018 samples.

- TCGA Uterine Cancer RNA Seq dataset (Firehose Legacy). This dataset consists of datasets of three different types of uterine cancer which are Uterine Endometrial Carcinoma, Uterine Carcinosarcoma and consists of

4.4 CNN configurations

Our proposed approach has a CNN as a classifier to classify cancer subtypes. CNN is configured with different layers and activation functions. Details of CNN layers and tunable parameters are given in the table 4.1.

We used four different proportions of training and test data. They are 80%,50%,20% and 10% of data for training and the rest for testing. We used the same train test split proportions for all the other state-of-the-art methods. We trained our model for 100 epochs during each generation. We used AMI score as the fitness function for our approach and trained it for 200 generations even though AMI scores stabilized earlier. After training, we loaded the best-saved model and tested the data.

Layers	**Specification**	**Activation func.**	**No. parameters**
Convolutional	32 filters with 4×4 size	relu	3104
Average Pooling	5×5 kernal size	N/A	N/A
Flatten	Flattens pooled matrix	N/A	N/A
Dense layer	300 neurons	relu	240300
Dense layer	128 neurons	relu	38528
Output	1 neuron	sigmoid	129

Table 4.1: Convolution Neural Network configuration for the Proposed Approach .

4.5 Training and Testing

Our GACNN takes the RNA-seq dataset and splits it into groups based on the gene grouping. The split groups are then fed into a respective convolutional neural network and each CNN is trained. While training each CNN the input groups are split row-wise based on the samples in order to split for training and testing. This way, each iteration through the CNN is validated with a train-test mechanism. Based on the resulting accuracy of each CNN, a top N number of groups are selected and genetic operations such as mutation and crossover is performed on the groups. The training process is repeated until a satisfactory Adjusted Mutual Information(AMI) score is obtained. The CNN with the best AMI score is chosen as the classifier.

4.6 Evaluation metric

In machine learning, the evaluation of a model is an essential task in order to measure its performance. Usually, accuracy is taken place as a performance metric. However, accuracy is not good enough to measure the performance of a model if it is a multi-class classification problem. In a multi-class problem, a model can make predictions randomly and still have a chance that its predictions may be correct. In such a case, we need another metric that can evaluate the model, whether it is predicting randomly or learned to separate classes. We select Adjusted Mutual Information as the evaluation metric. Details of Adjusted Mutual Information and its calculation methodology are given in the following subsection.

4.6.1 Adjusted Mutual Information

Adjusted Mutual Information(AMI) is an evaluation metric that widely used in multi-class classification problems. AMI adjusts the mutual information to account for chance. It accounts for the fact that the MI is generally higher for two classes with

a larger number of samples, regardless of whether there is actually more information shared[28]. For two classes, the AMI is given as:

$$AMI(U,V) = \frac{[MI(U,V) - E(MI(U,V))]}{[avg(H(U),H(V)) - E(MI(U,V))]} \tag{4.1}$$

This metric is independent of the absolute values of the labels: a permutation of the class or cluster label values won't change the score value in any way.

This metric is furthermore symmetric: switching label_true with label_pred will return the same score value.

4.7 Comparison and Analysis

4.7.1 Kidney Cancer Dataset

Kidney cancer dataset is sourced from The Cancer Genome Atlas (TCGA) program. It consists of m-RNA expression datasets for 891 patients. So it is essentially a matrix of 891 rows which are samples and 20532 columns which are featuresets. It consists of three classes which are the three subtypes of Kidney cancer. They are chromophobe carcinoma, clearcell carcinoma, and papillary carcinoma. The following figure 4.1 represents the principal component analysis of the dataset. It shows a distinction between the classes, but has an overlap.

Cancer Subtype	Samples
Clear Cell Carcinoma	534
Papillary Carcinoma	291
Chromophobe Carcinoma	66

Table 4.2: Distribution of Classes across Samples in Kidney Cancer Data

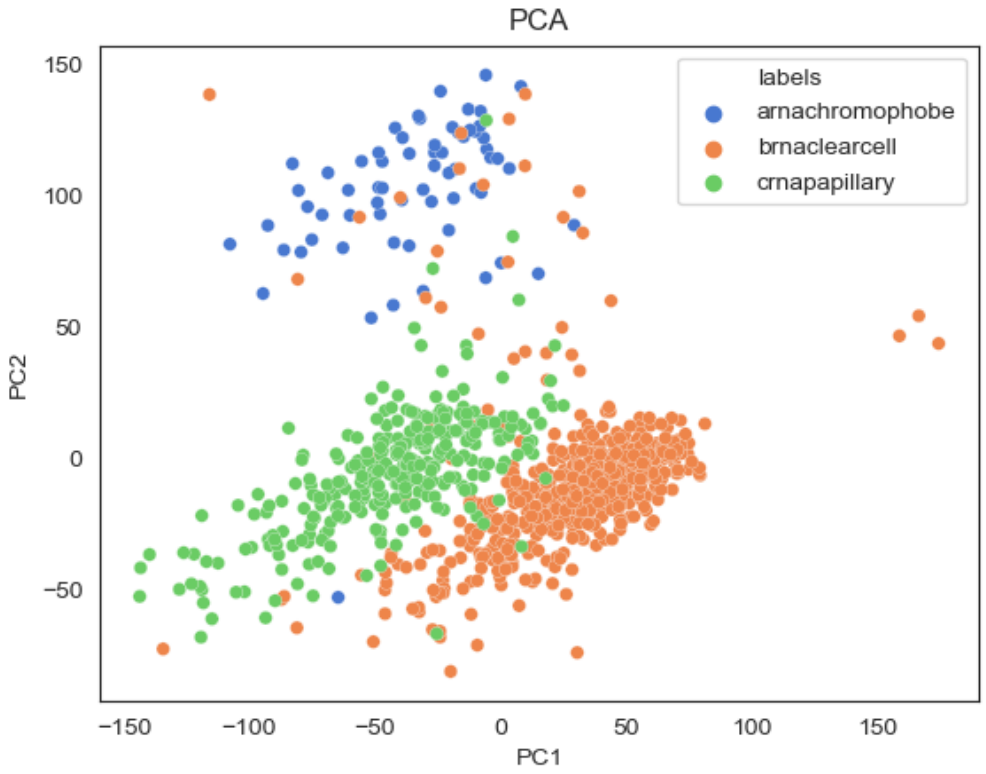

Figure 4.1: Principal Component Analysis of the TCGA Kidney Cancer Dataset

4.7.2 Lung Cancer Dataset

Lung cancer dataset is sourced from The Cancer Genome Atlas (TCGA) program. It consists of m-RNA expression datasets for 891 patients. So it is essentially a matrix of 891 rows which are samples and 20532 columns which are featuresets. It consists of three classes which are the three subtypes of Kidney cancer. They are chromophobe carcinoma, clearcell carcinoma, and papillary carcinoma. The following figure represents the principal component analysis of the dataset. It shows a distinction between the two available classes.

Cancer Subtype	Samples
Adenocarcinoma	517
Squamous Cell Carcinoma	501

Table 4.3: Distribution of Classes across Samples in Lung Cancer Data

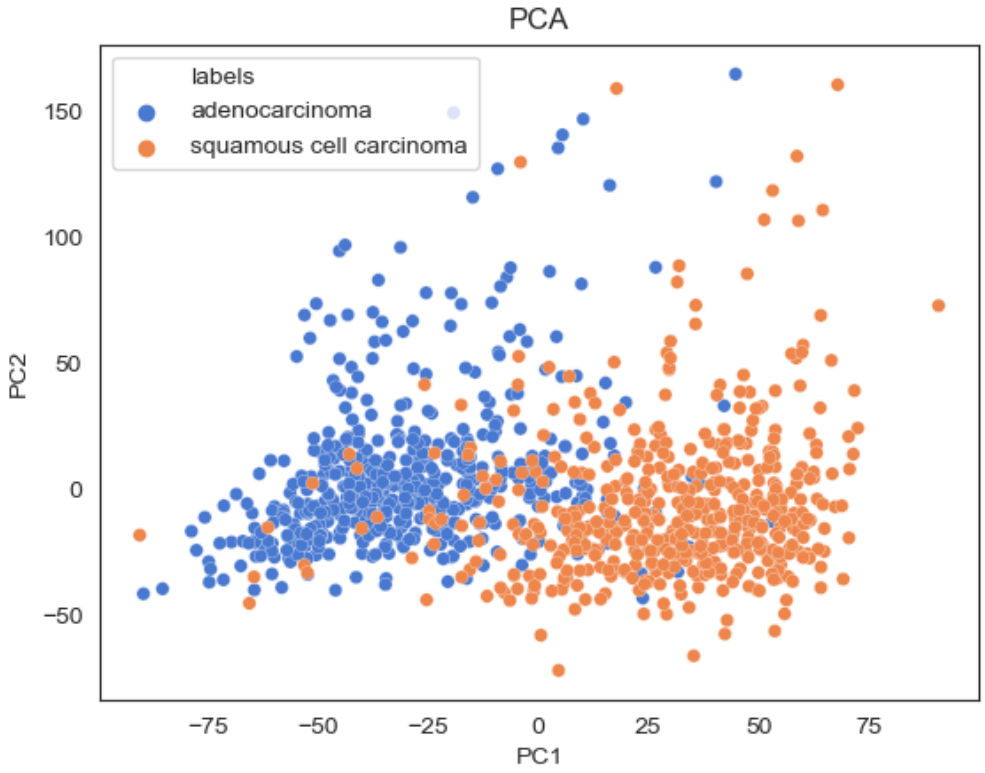

Figure 4.2: Principal Component Analysis of the TCGA Lung Cancer Dataset

4.7.3 PCA on Uterine Cancer Dataset

Uterine cancer dataset is sourced from The Cancer Genome Atlas (TCGA) program. It consists of m-RNA expression datasets for 891 patients. So it is essentially a matrix of 891 rows which are samples and 20532 columns which are featuresets. It consists of three classes which are the three subtypes of Kidney cancer. They are chromophobe carcinoma, clearcell carcinoma, and papillary carcinoma. The following figure 4.3 represents the principal component analysis of the dataset. It doesnot show

a distinction between the two available classes.

Cancer Subtype	Samples
Uterine Endometrial Adenocarcinoma	177
Uterine Carcinosarcoma	57

Table 4.4: Distribution of Classes across Samples for Uterine Cancer data

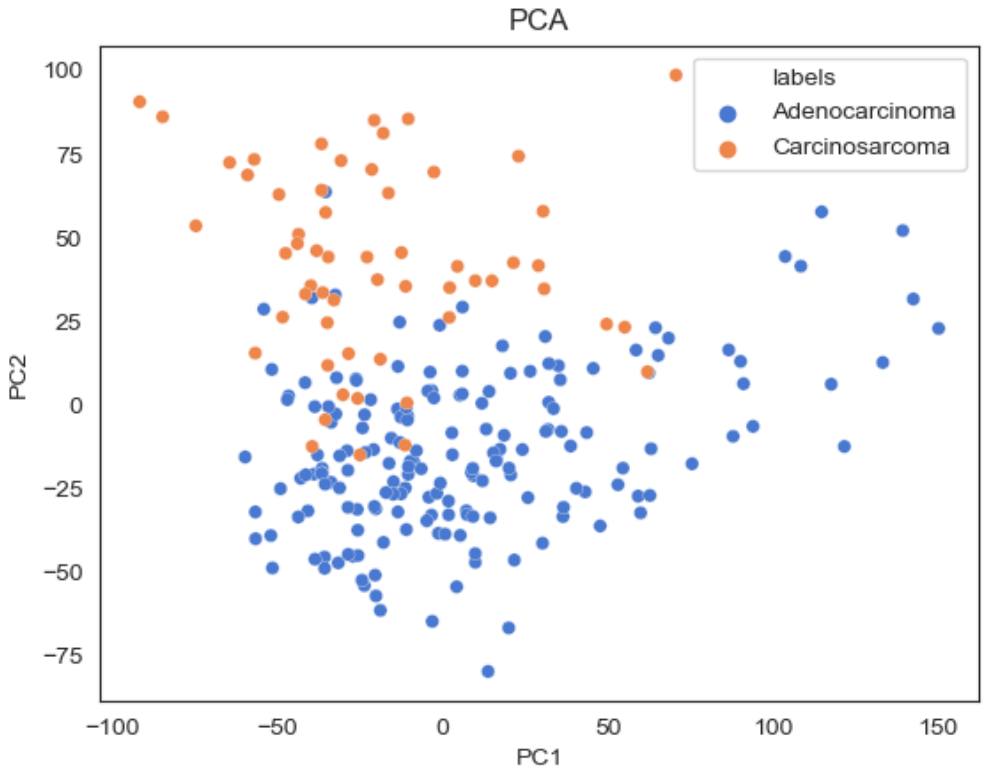

Figure 4.3: Principal Component Analysis of the TCGA Uterine Cancer Dataset

4.7.4 Performance comparison on Kidney Cancer Dataset

In order to effectively prove that our approach is better for few shot learning, we reduce the dataset into multiple fractions and compare the performance across several algorithms. The algorithms under consideration are

- GACNN(Proposed Approach)
- Affinitynet

- NeuralNet
- Support Vector Machine
- Naive Bayes
- Random Forest

The AMI scores for each algorithm is is taken for a particular dataset and is plotted for each fraction of the dataset.

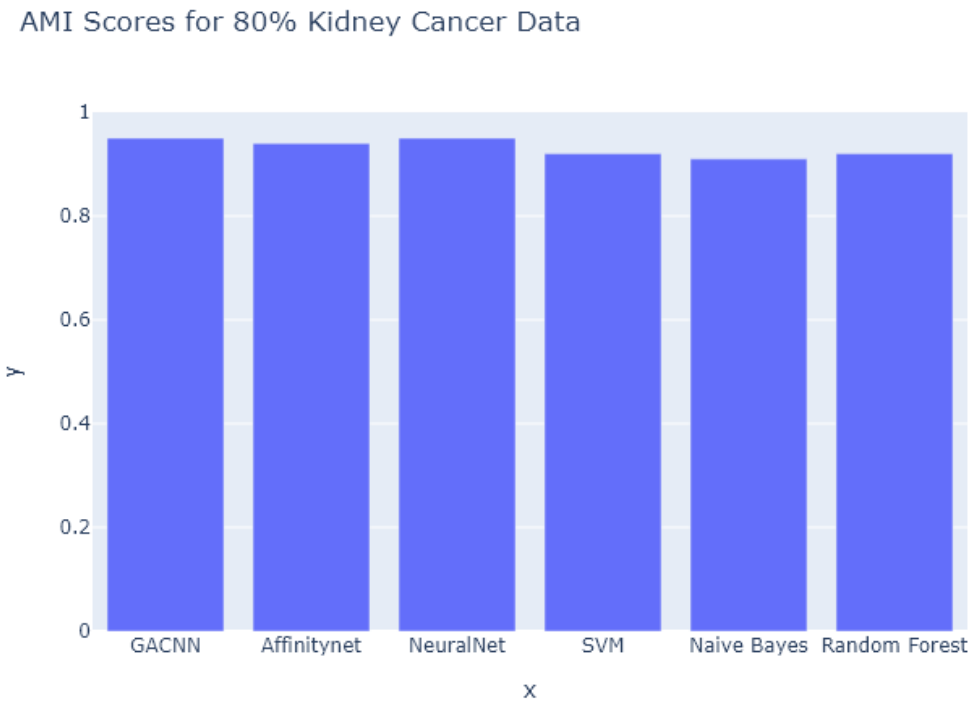

Figure 4.4: Performance Comparison for the TCGA Kidney Cancer Dataset on 80 percent of the data

With 80 percent of the dataset in the above figure 4.4, there is not much difference in performance amongst the chosen methods. This is to be expected due to the availability of data.

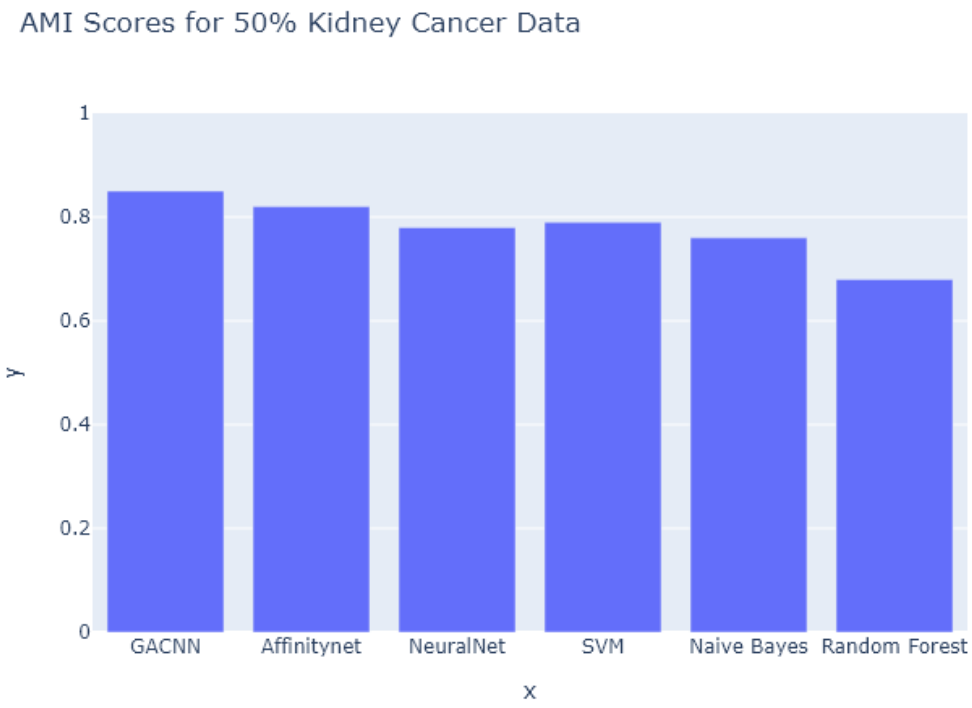

Figure 4.5: Performance Comparison for the TCGA Kidney Cancer Dataset on 50 percent of the data

With 50 percent of the dataset in the above figure **??**, there is a discernible amount of change amongst the AMI scores of the chosen methods. This is where our proposed approach and affinityNet works well with less amount of data.

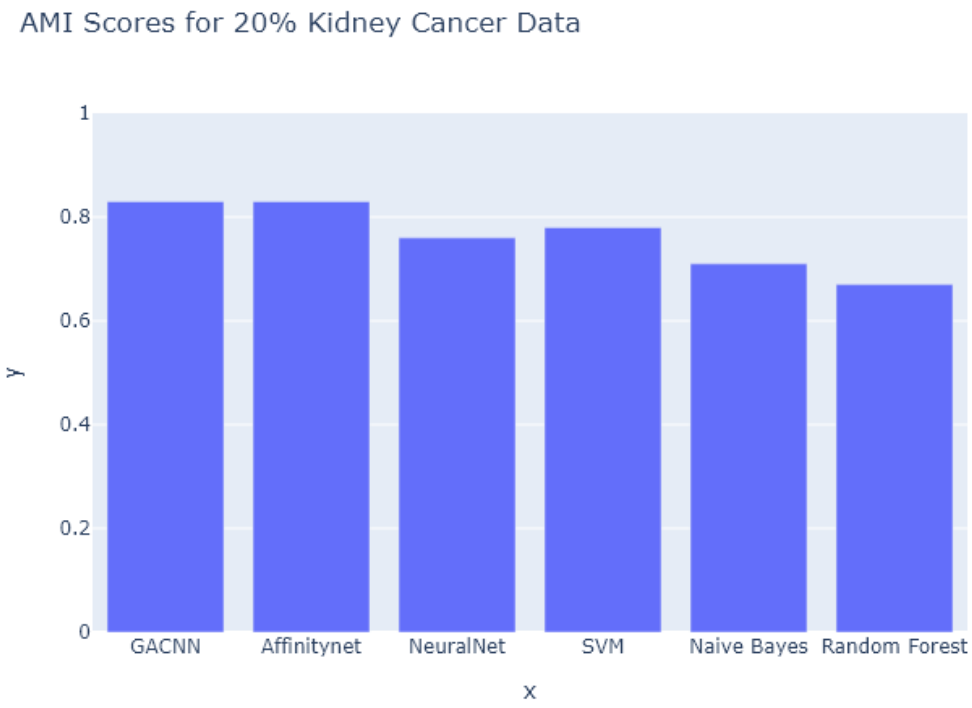

Figure 4.6: Performance Comparison for the TCGA Kidney Cancer Dataset on 20 percent of the data

With 20 percent of the dataset in the above figure 4.6, there is an increasing amount of change amongst the AMI scores of the chosen methods. This case further supports our proposed approach and affinityNet for few shot learning.

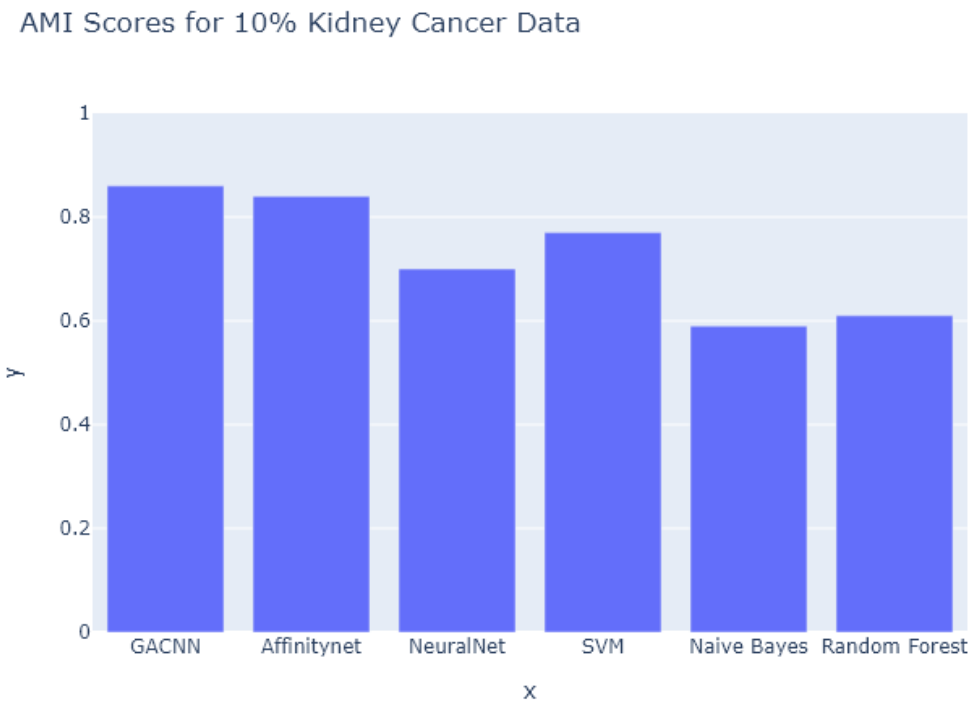

Figure 4.7: Performance Comparison for the TCGA Kidney Cancer Dataset on 10 percent of the data

With 10 percent of the dataset in the above figure 4.7, which is the norm for initial patient studies, a much reliable score is obtained from the proposed approach and affinitynet. Our method performs better due to using gene groups.

4.7.5 Performance comparison on Lung Cancer Dataset

The AMI scores of the various algorithms for the lung cancer dataset is plotted for each fraction of the dataset below.

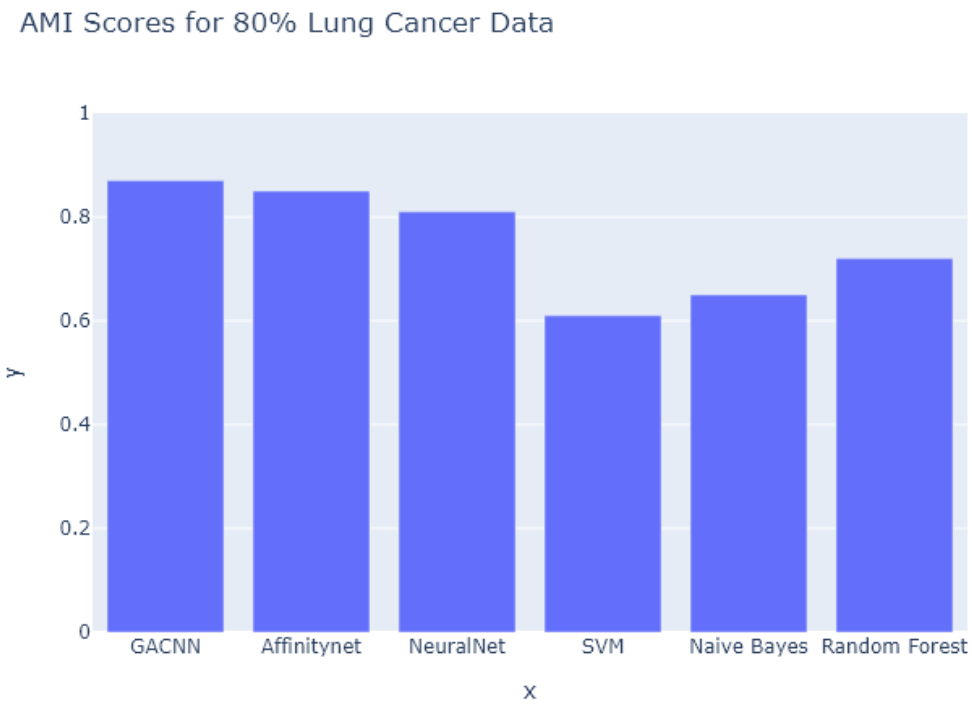

Figure 4.8: Performance Comparison for the TCGA Lung Cancer Dataset on 80 percent of the data

With 80 percent of the dataset in the above figure 4.8, there is not much difference in performance amongst the chosen methods. This is to be expected due to the availability of data. We have observed similar results in the kidney cancer dataset.

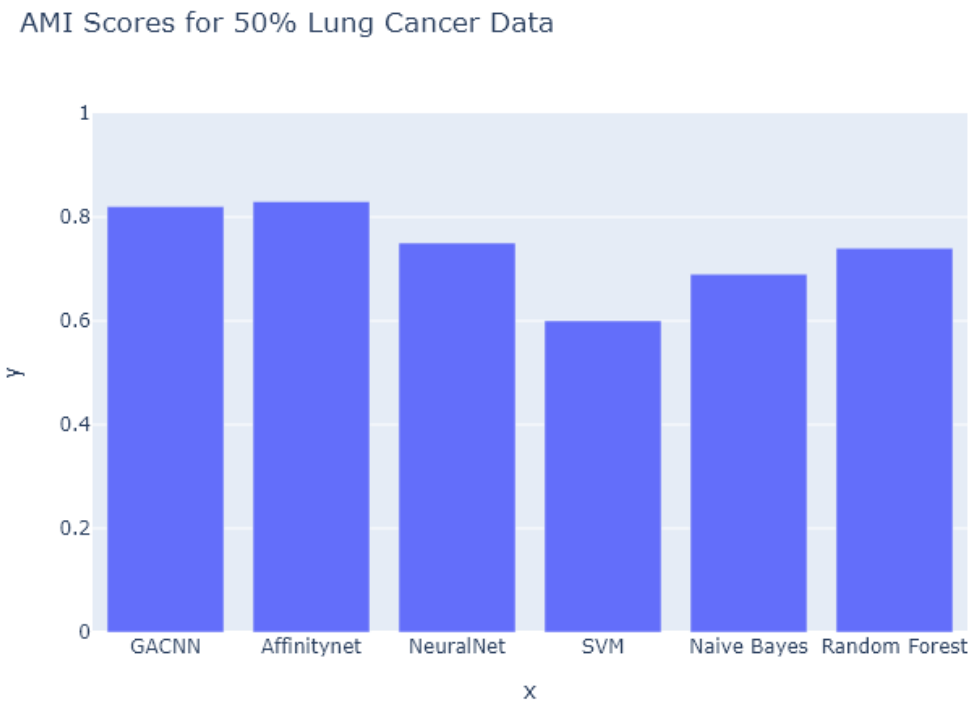

Figure 4.9: Performance Comparison for the TCGA Lung Cancer Dataset on 50 percent of the data

With 50 percent of the dataset in the above figure 4.9, there is a discernible amount of change amongst the AMI scores of the chosen methods. This is where our proposed approach and affinityNet works well with less amount of data.

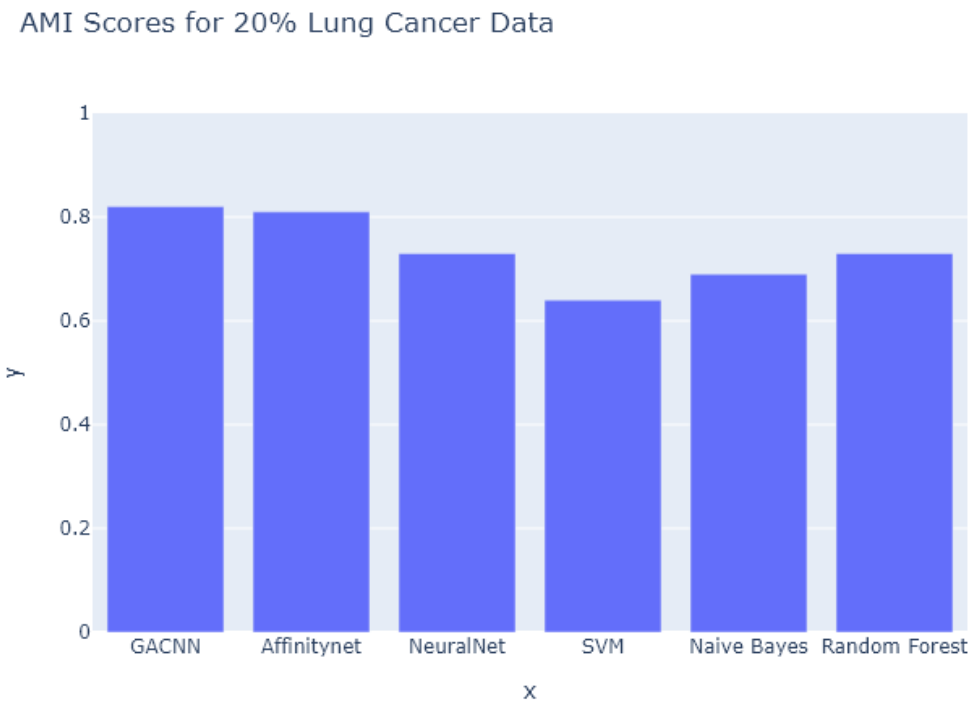

Figure 4.10: Performance Comparison for the TCGA Lung Cancer Dataset on 20 percent of the data

With 20 percent of the dataset in the above figure 4.10, there is an increasing amount of change amongst the AMI scores of the chosen methods. This case further supports our proposed approach and affinityNet for few shot learning.

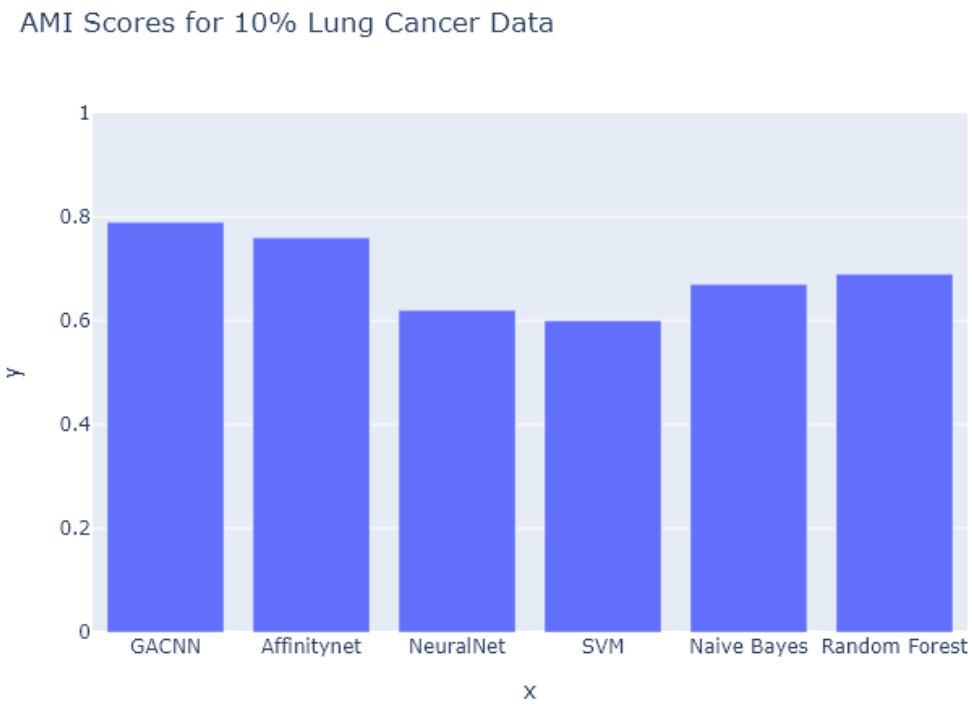

Figure 4.11: Performance Comparison for the TCGA Lung Cancer Dataset on 10 percent of the data

With 10 percent of the dataset in the above figure 4.11, which is the norm for initial patient studies, a much reliable score is obtained from the proposed approach and affinitynet. Our method performs better due to using gene groups.

4.7.6 Performance comparison on Uterine Cancer Dataset

The accuracy scores of the various algorithms for the Uterine cancer dataset is plotted for each fraction of the dataset below.

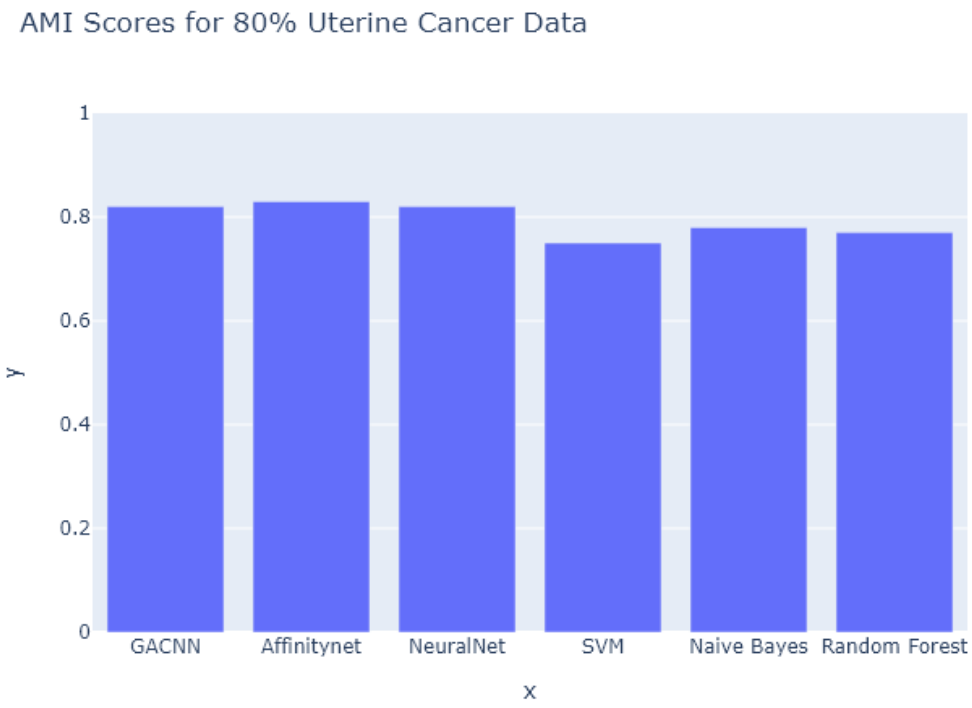

Figure 4.12: Performance Comparison for the TCGA Uterine Cancer Dataset on 80 percent of the data

With 80 percent of the dataset in the above figure 4.12, there is not much difference in performance amongst the chosen methods. This is to be expected due to the availability of data.

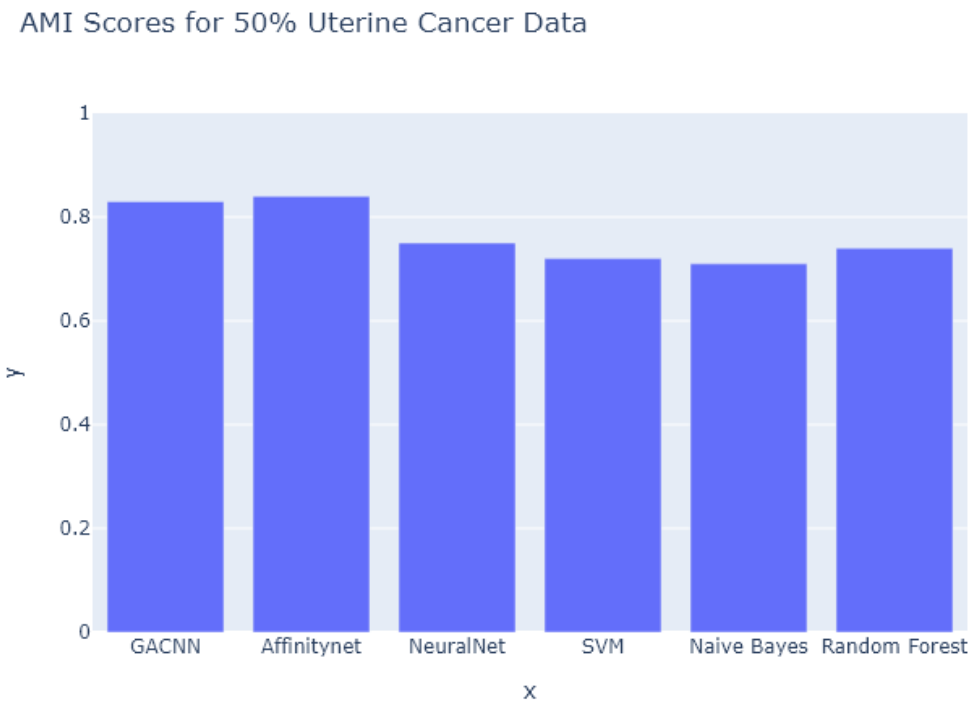

Figure 4.13: Performance Comparison for the TCGA Uterine Cancer Dataset on 50 percent of the data

With 50 percent of the dataset in the above figure 4.13, there is a discernible amount of change amongst the AMI scores of the chosen methods. This is where our proposed approach and affinityNet works well with less amount of data.

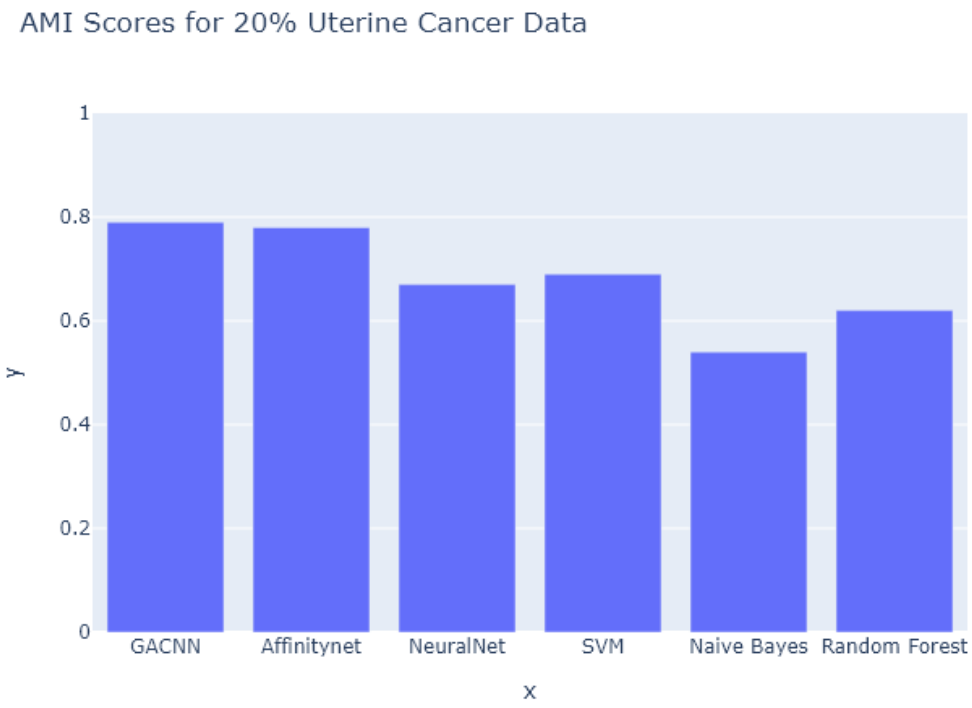

Figure 4.14: Performance Comparison for the TCGA Uterine Cancer Dataset on 20 percent of the data

With 20 percent of the dataset in the above figure 4.14, there is an increasing amount of change amongst the AMI scores of the chosen methods. This case further supports our proposed approach and affinityNet for few shot learning.

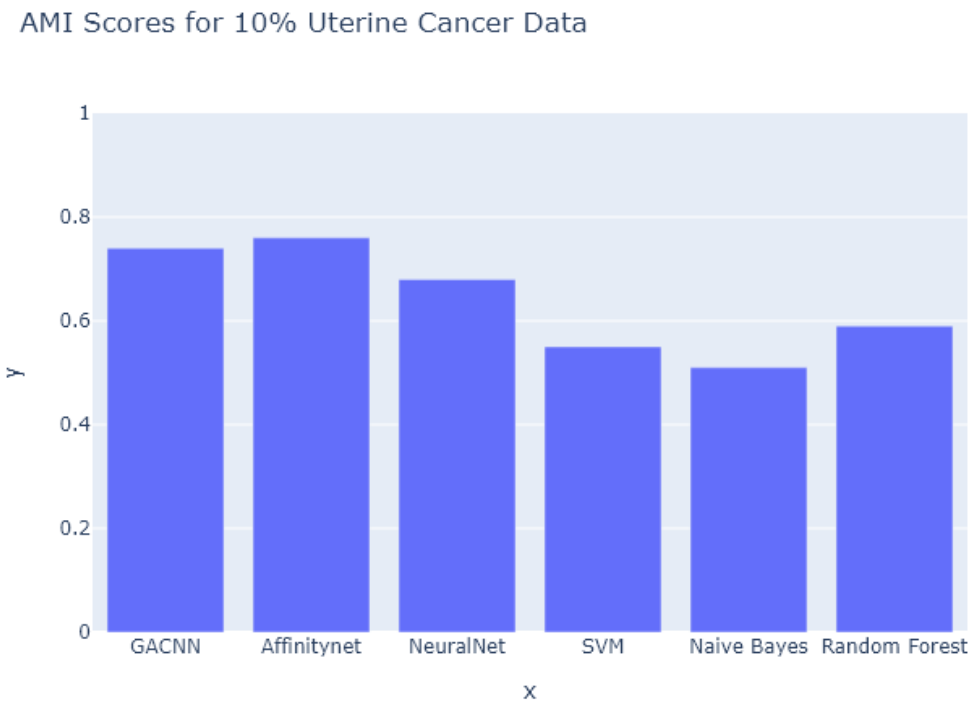

Figure 4.15: Performance Comparison for the TCGA Kidney Cancer Dataset on 10 percent of the data

With 10 percent of the dataset in the above figure 4.15, which is the norm for initial patient studies, a much reliable score is obtained from the proposed approach and affinitynet. Our method performs better due to using gene groups.

The following table 4.5 represents standard deviation of AMI scores for all the performance algorithms across the three cancer datasets

Method	Kidney Cancer	Lung Cancer	Uterine Cancer
GACNN	0.014	0.029	0.035
AffinityNet	0.012	0.033	0.033
Neural Networks	0.041	0.069	0.060
SVM	0.079	0.016	0.077
Naive Bayes	0.063	0.017	0.019
Random Forest	0.039	0.019	0.076

Table 4.5: Standard Deviation of AMI scores across all methods among the three datasets

4.7.7 Analysis of Performance across Datasets

From the above results, we can observe that our proposed approach performs better against other methods when there is less amount of data i.e. where there is few-shot learning involved. We can see from the results that our predecessor method, the affinitynet keeps up with our AMI scores but ours consistently perform better withan exception of a few cases. We can also see that uterine cancer dataset has not performed properly across the methods. This is due to the lack of proper identification of classes of cancer, a lack of data and the nature of the cancer. Even in this case, our method outperforms most methods in few-shot learning.

Chapter 5

Conclusion and Future Work

5.1 Conclusion

We proposed a genetic algorithm based convolutional neural network for few shot learning in disease type prediction on RNA-seq data. This is a novel architecture which constantly introduces variations in the form of genetic operations and retrain the convolutional neural networks in such a way that it gets trained accurately on less amount of data. We observe from the experimental results that our model has a better weighted average score than the existing affinitynet model. We also introduce domain knowledge in the form of gene grouping based on the molecular structure and nature of the genes as recommended by the Human Genome Organization. Existing latest method involves using attention pooling to achieve few shot learning. Although this is efficient in performing few shot learning, it does not use the characteristics of genetic data or any form of domain knowledge to optimize the results. We modify the architecture of the convolutional neural network completely using genetic algorithms such that it is fit for few shot learning. We also introduced genetic operations in the training process to introduce variations so that the model is not overfitted. This thesis focuses on elaborating the GACNN approach for few shot learning and also

incorporates domain knowledge.

5.1.1 Discussion

This introduction of a genetic algorithm to convolutional neural networks can also be applied to traditional neural networks, if the case demands for the utilization of conventional neural networks. Other traditional machine learning methods cannot be used in conjunction with this type of retraining as it gets overfitted really fast and genetic algorithms do not introduce sufficient variation to the algorithm. This architecture can also be broadly applied in disease classification problems in addition to the subtype classification problems, if the source data is RNA-seq data. And as an alternative version of the algorithm, the gene grouping can be replaced with random grouping where the initial dataset is arbitrarily split into groups and features are randomly assigned to it. This approach also works well for few-shot learning, although incorporating domain knowledge will increase the performance of the algorithm as it pre-assigns related features to each group.

5.2 Future Work

We tested our algorithm on different types of cancer datasets which use mRNA data. This procedure can be extrapolated to work with other types of datasets given any inherent grouping mechanism. As we have already discussed, even without the gene groups, random grouping also seems effective in producing results for few shot learning. So, the proposed algorithm with random grouping can be applied to other domains as well for semi-supervised learning or for few shot learning.

Bibliography

[1] J. Abrams, B. Conley, M. Mooney, J. Zwiebel, A. Chen, J. Welch, N. Takebe, S. Malik, L. McShane, E. Korn, Mickey P. Williams, L. Staudt, and J. Doroshow. National cancer institute's precision medicine initiatives for the new national clinical trials network. *American Society of Clinical Oncology educational book. American Society of Clinical Oncology. Annual Meeting*, pages 71–6, 2014.

[2] E. Blaveri, J. Simko, J. Korkola, Jeremy L. Brewer, F. Baehner, K. Mehta, S. Devries, T. Koppie, Sunanda Pejavar, P. Carroll, and F. Waldman. Bladder cancer outcome and subtype classification by gene expression. *Clinical cancer research : an official journal of the American Association for Cancer Research*, 11 11:4044–55, 2005.

[3] J. Bono and A. Ashworth. Translating cancer research into targeted therapeutics. *Nature*, 467:543–549, 2010.

[4] Bryony Braschi, P. Denny, Kristian A. Gray, Tamsin E. Jones, R. Seal, S. Tweedie, Bethan Yates, and E. Bruford. Genenames.org: the hgnc and vgnc resources in 2019. *Nucleic Acids Research*, 47:D786 – D792, 2019.

[5] L. Breiman. Random forests. *Machine Learning*, 45:5–32, 2004.

[6] Joshua D. Campbell, A. Alexandrov, J. Kim, J. Wala, A. Berger, C. Pedamallu, Sachet A. Shukla, Guangwu Guo, Angela N. Brooks, B. Murray, M. Imielin-

ski, X. Hu, Shiyun Ling, Rehan Akbani, M. Rosenberg, C. Cibulskis, A. Ramachandran, E. Collisson, D. Kwiatkowski, M. Lawrence, J. Weinstein, R. Verhaak, C. Wu, P. Hammerman, A. Cherniack, G. Getz, Maxim N. Artyomov, R. Schreiber, R. Govindan, and M. Meyerson. Distinct patterns of somatic genome alterations in lung adenocarcinomas and squamous cell carcinomas. *Nature genetics*, 48:607 – 616, 2016.

[7] Julio E. Celis, H. Wolf, and M. Østergaard. Bladder squamous cell carcinoma biomarkers derived from proteomics. *ELECTROPHORESIS*, 21, 2000.

[8] François Chollet. Keras: The python deep learning library. 2018.

[9] F. Collins and H. Varmus. A new initiative on precision medicine. *The New England journal of medicine*, 372 9:793–5, 2015.

[10] K. Crammer and Y. Singer. On the algorithmic implementation of multiclass kernel-based vector machines. *J. Mach. Learn. Res.*, 2:265–292, 2001.

[11] E. David and Iddo Greental. Genetic algorithms for evolving deep neural networks. In *GECCO Comp '14*, 2014.

[12] Boyang Deng, Q. Liu, Siyuan Qiao, and A. Yuille. Few-shot learning by exploiting visual concepts within cnns. *arXiv: Computer Vision and Pattern Recognition*, 2017.

[13] Parsa Esfahanian and Mohammad Reza Akhavan. Gacnn: Training deep convolutional neural networks with genetic algorithm. *ArXiv*, abs/1909.13354, 2019.

[14] A. Feldman, C. Kontos, J. McClung, G. Gerhard, K. Khalili, and J. Cheung. Precision medicine for heart failure: Lessons from oncology. *Circulation. Heart failure*, 10 6, 2017.

[15] T. Golub, D. K. Slonim, P. Tamayo, C. Huard, M. Gaasenbeek, J. Mesirov, H. Coller, M. Loh, J. Downing, M. Caligiuri, C. Bloomfield, and E. Lander. Molecular classification of cancer: class discovery and class prediction by gene expression monitoring. *Science*, 286 5439:531–7, 1999.

[16] I. Goodfellow, Yoshua Bengio, and Aaron C. Courville. Deep learning. *Nature*, 521:436–444, 2015.

[17] P. Joshi, S. Jeong, and T. Park. Cancer subtype classification based on super-layered neural network. *2019 IEEE International Conference on Bioinformatics and Biomedicine (BIBM)*, pages 1988–1992, 2019.

[18] Yoon Kim. Convolutional neural networks for sentence classification. In *EMNLP*, 2014.

[19] Knowles. The genetics of transitional cell carcinoma: progress and potential clinical application. *BJU International*, 84, 1999.

[20] J. Korkola, S. Devries, J. Fridlyand, E. Hwang, A. L. Estep, Y. Chen, K. Chew, S. Dairkee, R. M. Jensen, and F. Waldman. Differentiation of lobular versus ductal breast carcinomas by expression microarray analysis. *Cancer research*, 63 21:7167–75, 2003.

[21] Konstantina Kourou, T. Exarchos, K. Exarchos, M. Karamouzis, and D. Fotiadis. Machine learning applications in cancer prognosis and prediction. *Computational and Structural Biotechnology Journal*, 13:8 – 17, 2015.

[22] A. Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton. Imagenet classification with deep convolutional neural networks. In *CACM*, 2017.

[23] E. Lander. Comprehensive molecular profiling of lung adenocarcinoma. *Nature*, 511:543 – 550, 2014.

[24] Sangseon Lee, Sangsoo Lim, Taeheon Lee, Inyoung Sung, and S. Kim. Cancer subtype classification and modeling by pathway attention and propagation. *Bioinformatics*, 2020.

[25] Siew Mooi Lim, Abu Bakar Md. Sultan, Md. Nasir Sulaiman, Aida Mustapha, and Kuan Yew Leong. Crossover and mutation operators of genetic algorithms. *International Journal of Machine Learning and Computing*, 7:9–12, 2017.

[26] Y. Liu, Juho Lee, Minseop Park, Saehoon Kim, E. Yang, SungJu Hwang, and Y. Yang. Learning to propagate labels: Transductive propagation network for few-shot learning. In *ICLR*, 2019.

[27] Tianle Ma and A. Zhang. Integrate multi-omic data using affinity network fusion (anf) for cancer patient clustering. *2017 IEEE International Conference on Bioinformatics and Biomedicine (BIBM)*, pages 398–403, 2017.

[28] Tianle Ma and A. Zhang. Affinitynet: semi-supervised few-shot learning for disease type prediction. *ArXiv*, abs/1805.08905, 2019.

[29] G. Miller, P. Todd, and S. Hegde. Designing neural networks using genetic algorithms. In *ICGA*, 1989.

[30] Pooya Mobadersany, Safoora Yousefi, M. Amgad, D. Gutman, J. Barnholtz-Sloan, J. V. Velazquez Vega, D. Brat, and L. Cooper. Predicting cancer outcomes from histology and genomics using convolutional networks. *bioRxiv*, 2017.

[31] Paulito Palmes, T. Hayasaka, and S. Usui. Mutation-based genetic neural network. *IEEE Transactions on Neural Networks*, 16:587–600, 2005.

[32] S. Patil and M. Bhende. Comparison and analysis of different mutation strategies to improve the performance of genetic algorithm. 2014.

[33] S. Ravi and H. Larochelle. Optimization as a model for few-shot learning. In *ICLR*, 2017.

[34] Mengye Ren, E. Triantafillou, S. Ravi, J. Snell, Kevin Swersky, J. Tenenbaum, H. Larochelle, and R. Zemel. Meta-learning for semi-supervised few-shot classification. *ArXiv*, abs/1803.00676, 2018.

[35] C. Ricketts, A. A. de Cubas, Huihui Fan, C. Smith, M. Lang, E. Reznik, R. Bowlby, Ewan A. Gibb, Rehan Akbani, R. Beroukhim, D. Bottaro, T. Choueiri, R. Gibbs, A. Godwin, S. Haake, A. Hakimi, E. Henske, J. Hsieh, T. Ho, R. Kanchi, Bhavani Krishnan, David J. Kwaitkowski, Wembin Lui, M. Merino, G. Mills, J. Myers, M. Nickerson, V. Reuter, L. Schmidt, C. S. Shelley, H. Shen, Brian Shuch, S. Signoretti, R. Srinivasan, P. Tamboli, G. Thomas, B. Vincent, C. Vocke, D. Wheeler, Lixing Yang, William T. Kim, A. G. Robertson, P. Spellman, W. K. Rathmell, and W. Linehan. The cancer genome atlas comprehensive molecular characterization of renal cell carcinoma. *Cell reports*, 23:313 – 326.e5, 2018.

[36] J. Schmidhuber. Deep learning in neural networks: An overview. *Neural networks : the official journal of the International Neural Network Society*, 61:85–117, 2015.

[37] J. Snell, Kevin Swersky, and R. Zemel. Prototypical networks for few-shot learning. *ArXiv*, abs/1703.05175, 2017.

[38] A. Snyder, Vladimir Makarov, T. Merghoub, J. Yuan, J. Zaretsky, A. Desrichard, L. Walsh, M. Postow, P. Wong, Teresa S. Ho, Travis J Hollmann, Cameron Bruggeman, K. Kannan, Y. Li, Ceyhan Elipenahli, C. Liu, C. Harbison, L. Wang, A. Ribas, J. Wolchok, and T. Chan. Genetic basis for clinical response to ctla-4

blockade in melanoma. *The New England journal of medicine*, 371 23:2189–2199, 2014.

[39] Dirk Sudholt. How crossover speeds up building block assembly in genetic algorithms. *Evolutionary Computation*, 25:237–274, 2017.

[40] A. Umbarkar and P. Sheth. Crossover operators in genetic algorithms:a review. In *SOCO 2015*, 2015.

[41] S. Vaidyanathan, P. Mansour, Munehisa Ueno, K. Yamazaki, M. Wadhwa, B. Soni, G. Singh, P. Hughes, I. Watson, and P. Sett. Problems in early diagnosis of bladder cancer in a spinal cord injury patient: Report of a case of simultaneous production of granulocyte colony stimulating factor and parathyroid hormone-related protein by squamous cell carcinoma of urinary bladder. *BMC Urology*, 2:8 – 8, 2002.

[42] Dayong Wang, A. Khosla, Rishab Gargeya, Humayun Irshad, and A. Beck. Deep learning for identifying metastatic breast cancer. *ArXiv*, abs/1606.05718, 2016.

[43] Stephane Wenric and Ruhollah Shemirani. Using supervised learning methods for gene selection in rna-seq case-control studies. *Frontiers in Genetics*, 9, 2018.

[44] G. Wong, C. Leckie, and A. Kowalczyk. Fsr: feature set reduction for scalable and accurate multi-class cancer subtype classification based on copy number. *Bioinformatics*, 28 2:151–9, 2012.

[45] J. Xu, Peng Wu, Y. Chen, Q. Meng, H. Dawood, and Hassan Dawood. A hierarchical integration deep flexible neural forest framework for cancer subtype classification by integrating multi-omics data. *BMC Bioinformatics*, 20, 2019.

[46] Steven R. Young, Derek C. Rose, T. Karnowski, Seung-Hwan Lim, and Robert M. Patton. Optimizing deep learning hyper-parameters through an evolutionary algorithm. In *MLHPC '15*, 2015.

[47] H. Zhang. The optimality of naive bayes. In *FLAIRS Conference*, 2004.

Vita Auctoris

NAME:	Kowshik Sharan Subramanian
PLACE OF BIRTH:	Tiruchirapalli, Tamil Nadu, India
EDUCATION:	Bachelor of Engineering in Computer Science, Sri Krishna College of Engineering and Technology, Coimbatore, Tamil Nadu, India, 2018 Master of Science in Computer Science, University of Windsor, Windsor, Ontario, Canada, 2020.