Mutual Information-based Machine Learning with Microarray Cancer Data

by

Zahra Jandaghi

(Under the Direction of Liming Cai)

Abstract

Big data are indispensable for machine learning and complex data modeling. Such computation tasks with big data are expensive, requiring extensive computer memory and computing time. Thus methods are often sought that can scale down the amount of raw data or model size without compromising substantial information in the original data. Such approximation aims at reducing required computing resources while achieving high performance of the related machine learning tasks. In this research, we investigate the approximation issue in machine learning areas: feature selection to learn a small set of critical features in big data and neural network sparsification to determine a small set of pertinent connections between neurons in neural networks. An additional goal of this research is to reveal pertinent relationships across these two research areas.

Information theory has great potential in machine learning, offering an alternative way for information extraction from data and for the approximation of data models. In particular, mutual information-based methods have been developed for feature learning and sparsifying neural networks, nevertheless with mixed results. The previous work has yet to establish connections across the two machine learning areas. We propose a mutual information-based framework that aims at addressing the approximation issue in these two subtopics and revealing pertinent relationships between them.

In this research, the proposed mutual information-based is tested on a large amount of microarray gene expression of human cancer data for disease classification. Microarray expression data containing tens of thousands of genes are ideal for the evaluation of methods for both feature learning and neural network sparsification.

In particular, the significant gene subset identified by our method decreases the required number of genes to perform classification tasks ten to hundred folds and outperforms the previous methods' performance. Sparsification neural networks with mutual information between neuron outputs let us removing up to 90% unnecessary connections while maintaining or even improving the performance. Our experiments reveals sparsified neural network ignores unimportant (irrelevant) genes and only considers significant or pseudo-significant genes that we identify in the first part of this research through gene filtering.

INDEX WORDS: Mutual Information, Feature Selection, Sparse Neural Networks, Microarray

Mutual Information-based Machine Learning with Microarray Cancer Data

by

Zahra Jandaghi

B.S.c, Iran University of Science and Technology (IUST), 2012

A Dissertation Submitted to the Graduate Faculty of the
University of Georgia in Partial Fulfillment of the Requirements for the Degree.

Doctor of Philosophy

Athens, Georgia

2022

Mutual Information-based Machine Learning with Microarray Cancer Data

by

Zahra Jandaghi

Major Professor:	Liming Cai
Committee:	Khaled Rasheed
	Ismailcem B. Arpinar
	Xiuzhen Huang

Electronic Version Approved:

Ron Walcott

Dean of the Graduate School

The University of Georgia

December 2022

Dedication

To my amazing husband, Omid

Acknowledgments

First, I dedicate this dissertation to my one-of-a-kind husband, Omid. He has always been an attentive and loving friend throughout my Ph.D. journey, and I have the privilege of getting his best advice. I am sure I could never have done this without his invaluable support in different critical situations as he entered my life miraculously. This Ph.D. program made us first friends, later wife and husband.

I want to thank my parents and siblings, who I've missed the chance to see since I joined this Ph.D. program, and I miss them so much. I could not have come this far without their emotional support. Special gratitude goes to my sister, Parvaneh, who I used to describe as my first major blessing. I see myself as a product of her generous efforts and true love.

Moreover, I am grateful to have Dr. Liming Cai as my advisor. I have learned from him many life lessons along with research skills. He reminds me to have patience, consistency, and passion in my professional life and sets an excellent example for them. I also thank Dr. Russell Malmberg for being an outstanding mentor to me. I have benefited from his advice many times. Unfortunately, he is unavailable to be on my advisory committee. I sincerely wish him the best in life. I thank Dr. Khaled Rasheed for providing me with insightful advice in approaching research challenges. I also thank Dr. Xiuzhen Huang, who provided me with microarray cancer data I used in my experiments and agreed to be on my committee. Last but not least, I thank Dr. Ismailcem Arpinar for working with me and letting me have him on my advisory committee.

I also want to thank all my friends, close or far, old or new, through my whole life, who have touched different stages of my life drastically or slightly in some way. They also played a big role in who I am today and have contributed to my achievements.

This tough time coincides with much heartbreaking and saddening news, especially in my home country, Iran. I hope all these unpleasant events will turnout something good for everybody worldwide with peace, happiness, and freedom.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Convenient access to data lead to efforts toward handling big data including customizing machine learning algorithms, so that they could work with high dimensional data. Big data includes many random variables and exact modeling them by machine learning algorithms is intractable since finding an optimal subset of features among high number of features is a NP-complete problem [29]. That's why we attempt to find an approximate solution instead of an exact answer to solve challenging problems involving too many features.

In this dissertation, we follow two approaches to address an approximate solution to model high-dimensional data with machine learning algorithms. First, by using an information theory-based method, we aim to learn essential feature subset to approximate high-dimensional data. Second, we investigate the sparsification of neural networks by considering mutual information between two connected neurons. We show the effectiveness of each of these methods independently and explain the connection between them utilizing experiments on microarray cancer data.

Microarray data is one of the popular biology data types, including gene expressions. Modeling microarray cancer data accurately can help with cancer detection more accurately. Microarray data provides a massive amount of gene expression information for comparing tissues and treatments. Microarray data are considered high-dimensional as they often have a limited number of samples (patients) but with hun-

dreds or thousands of genes, which are usually considered features for machine learning [56, 55, 30, 54, 13].

Many researchers [67, 35, 11, 43, 39, 68, 10, 16, 49, 64, 62, 14] have taken different approaches to address the problem of having too many variables in input data for classification tasks . Some works focus on studying different ways of selecting significant features from the high-dimensional data to reduce the dimensionality. Among those works, there has been a research to improve prediction accuracy with the help of mutual information (MI). Roberto Battiti in [11] used filtering genes by maximizing mutual information (MI). They considered two MI values, between a given label and a gene, and between two genes. Since they calculated MI among continuous values, they approximated MI.

A recent deep learning study [43] shows that decreasing the number of genes while utilizing different neural networks allows the model to perform well. The study used the analysis of variance (ANOVA) F-test and information gain (IG) between gene and label to filter genes. Furthermore, another study used gene-label MI to filter genes, and defined a discriminative margin to select essential genes to conduct semi-supervised classification [39]. Mutual information based research in [68] detects relevant genes, then performed the *lasso* method to filter redundant information.

High-dimensional data could also be tackled by sparsifying neural networks used as classifiers and directing neural networks toward employing only essential variables rather than the provided whole feature set. A recent study [26] follows a mutual information- based approach and proposed pruning a neural network by considering the MI of two connected neurons and then ignoring neurons with low MI.

In this work, we propose an information theory-based method to identify critical variables from a given set of variables. We consider 3 different types of mutual information for this purpose: mutual information between two individual variables and between one variable and a given label. We also include conditional mutual information, which is mutual information of two variables conditioned on a specific label. This third type of mutual information captures correlation between two variables when it cannot be captured by the first type. A typical example is the logical operator Exclusive OR.

Our second research utilizes a neural network as the task classifier and explores pruning neural network models with the proposed mutual information method. This approach ignores unnecessary connections in a neural network in the classification task. Pruning strategies compress the network, making it sparser and requiring less computation. This work aims to optimize computation resource usage while providing an accurate prediction.

In this dissertation, we consider two approaches in classification problem of microarray cancer data. Each such dataset is a matrix $M \times N$ with M patients, and N genes, and each element of the matrix being gene expression of a specific patient associated with a known gene. To improve classification of microarray data, we propose a filtering method to identify significant genes based on the aforementioned 3 types of mutual information. We test our method on several microarray cancer datasets and identify a significant gene subset for each. We integrate SVM-RFE, a machine learning feature selection (explained in 2.2.5), to acquire a stable significant gene subset from all the provided genes in an attempt to remove bias in our selection process. To validate our method, we monitor classification performance while also getting biological confirmation [5] to endorse our significant gene subset as significant ones. In particular, we experiment our method on 4 microarray cancer dataset with two classes (sick and healthy) showing efficiency and reliability compared to existing techniques in filtering tens of thousands of genes to just a few hundred which remain to be effective.

We also take the filtering method further and integrate the idea of filtering with mutual information into sparsifying a neural network. In this part, we keep only connections of pairs of neurons that have high enough mutual information. That approach makes the model size much smaller and computation would be cheaper while the performance is not changed noticeably.

This second research reveals only significant or pseudo-significant genes needed to be considered in neural network. Although, the datasets we used for our experiments are not extremely large and reducing model size are not significant, our results basically prove the concept of relating data filtering to sparse neural network design and can apply to any large datasets for machine learning.

The rest of this dissertation is organized as follows. In chapter 2, we briefly review some of the required background in information theory. We also summarize feature selection techniques, which form a broad field of study in machine learning, beneficial to our gene identification framework. We also introduce the problem of sparse neural networks and review some current research approaches. In chapter 3, we introduce the methodology forming the structure of this research. In Chapters 4, we explain the algorithms developed based on the methodology to conduct detailed research steps. Chapter 5 shows the data used in our experiments. In chapter 6 we demonstrate our experiment settings and results. Chapter 7 discusses our observations with conclusions. In particular Table 1.1 shows the flow of presented content in this dissertation.

Table 1.1: Content of this dissertation

Chapter 1	Introduction	Introduction to the dissertation and motivation of this work
Chapter 2	Preliminaries	Background knowledge review for this research
Chapter 3	Methodology	Proposed methods to tackle the defined problem
Chapter 4	Algorithms	Developed algorithms based on our methodology
Chapter 5	Data	Microarray cancer data utilized in our experiments
Chapter 6	Experiments	Experiments and results supporting our proposed method
Chapter 7	Discussion & Conclusion	Discussion and future work

Chapter 2

Preliminaries

2.1 Information Theory

The broad scientific field of information theory attempts to quantify the reliability of information transfer through unreliable mediums. It views information as a system and attempts to quantify its parameters. Information theory studies the relations of involved variables in different settings and analyzes their effects on one another [59, 42]. Because computer scientists deal mainly with improving machine implementation, information theory helps them realistically model a series of events with many variables by understanding each component's individual and organizational roles. Information theory has been widely used to assist in other fields, such as machine learning, which attempts to identify underlying patterns in data and draw generalizations from it. To identify a subset of the most important parameters from a big set of variables, we borrow this concept to help filter out a significant subset.

2.1.1 Entropy

The concept of entropy was first introduced by C. E. Shannon [53] to quantify the state of uncertainty in a random system through the probability of events. In other words, entropy measures the unpredictability of an outcome by considering its probability. Machine learning science is strongly tied to entropy since it

measures unpredictability and aligns well with the purpose of machine learning algorithms that identify similar patterns across samples. Equation 2.1 shows the formula to calculate the entropy of random variable A. Let A and B be two random variables with finite discrete domains $\mathcal{D}_{\mathcal{A}}$ and $\mathcal{D}_{\mathcal{B}}$ respectively.

$$H(A) = -\sum_{a \in \mathcal{D}_{\mathcal{A}}} P(A = a) \log_2 P(A = a) \tag{2.1}$$

The joint entropy of two random variables A and B is calculated by equation 2.2

$$H(A, B) = -\sum_{(a,b) \in \mathcal{D}_{\mathcal{A}} \times \mathcal{D}_{\mathcal{B}}} P(A = a, B = b) \log_2 P(A = a, B = b)) \tag{2.2}$$

Similarly, the conditional entropy of two random variables A and B is calculated by equation 2.3

$$H(A|B) = -\sum_{(a,b) \in \mathcal{D}_{\mathcal{A}} \times \mathcal{D}_{\mathcal{B}}} P(A = a|B = b) \log_2 P(A = a|B = b) \tag{2.3}$$

Definition 1. The Kullback–Leibler divergence [37] (also called relative entropy), denoted $D_{KL}(P \parallel Q)$ measures how much probability distribution P over $\mathbf{X}$ is different from probability distribution Q over $\mathbf{X}$. Assume distribution P represents the data, and distribution Q represents an approximation of P which indicates $D_{KL}(P \parallel Q) \to 0$ is desirable. Equation 2.4 shows how we calculate KL-divergence between two distributions P and Q.

$$D_{KL}(P||Q) = \sum_{x \in \mathbf{X}} P(x) \log_2 \frac{P(x)}{Q(x)} \tag{2.4}$$

2.1.2 Mutual information

Generalization requires certainty, and *mutual information* (MI) measures the provided information between several random variables in an attempt to quantify uncertainty. We utilize this concept to find a solution for the minimum variable identification problem 2.2.1. Mutual information helps us identify the most relevant involved variables.

Definition 2. Let A and B be two random variables with finite discrete domains $\mathcal{D}_A$ and $\mathcal{D}_B$ respectively. The *mutual information* between A and B is the KL-divergence between the joint probability $P(A, B)$ and independent probability $P(A)P(B)$. That is

$$\begin{aligned} I(A; B) &= D_{KL}(P(A, B)||P(A)P(B)) \\ &= \sum_{(a,b)\in\mathcal{D}_A\times\mathcal{D}_B} P(A = a, B = b) \log_2 \frac{P(A = a, B = b)}{P(A = a)P(B = b)} \end{aligned} \tag{2.5}$$

The term logodds $\log_2 \frac{P(A=a,B=b)}{P(A=a)P(B=b)}$ corresponding to value (a, b) in the summation of 2.5 is called *point mutual information*, denoted with $pmi(A = a; B = b)$. Therefore,

$$I(A; B) = \sum_{(a,b)\in\mathcal{D}_A\times\mathcal{D}_B} P(A = a, B = b)pmi(A = a; B = b)$$

Alternatively mutual information can be rewritten in the form of *information gain*:

$$I(A; B) = H(A) - H(A|B) = H(B) - H(B|A) \tag{2.6}$$

where

$$H(A) = -\sum_{a\in\mathcal{D}_A} P(A = a) \log_2 P(A = a)$$

and

$$H(A|B) = -\sum_{a\in\mathcal{D}_A, b\in\mathcal{D}_B} P(A = a, B = b) \log_2 \frac{P(A = a, B = b)}{P(B = b)}$$

are the Shannon *entropy* and *conditional entropy*, respectively. Figure 2.1 shows the relations among the MI of two random variables A and B with their individual and conditional entropy.

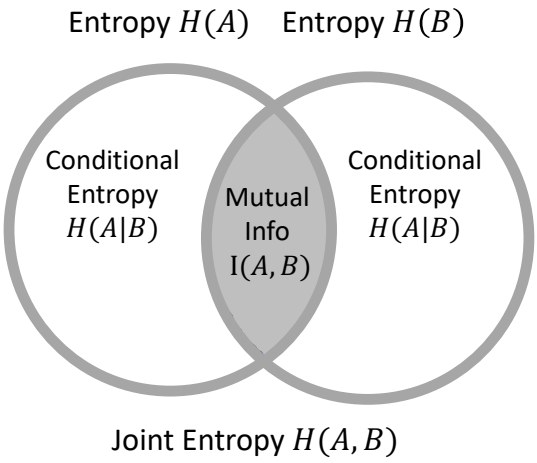
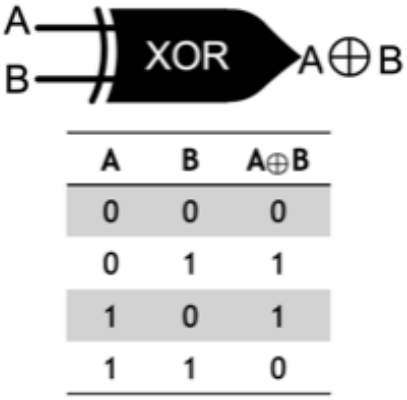


Figure 2.1: Mutual Information of two random variables.



A	B	A⊕B
0	0	0
0	1	1
1	0	1
1	1	0

Figure 2.2: Exclusive OR.

Definition 3. The mutual information between variables A and B *conditional upon* value $c \in \mathcal{D}_c$ of variable C is defined as expectation:

$$I(A;B|C=c) = \sum_{(a,b)\in\mathcal{D}_A\times\mathcal{D}_B} P(A=a,B=b|C=c)\log\frac{P(A=a,B=b|C=c)}{P(A=a|C=c)P(B=b|C=c)}$$

The *conditional mutual information* of A and B given C is defined as

$$I(A;B|C) = \sum_{c\in\mathcal{D}_C} P(C=c)I(A;B|C=c)$$

which is also written as

$$I(A;B|C) = \sum_{(a,b,c)\in\mathcal{D}_A\times\mathcal{D}_B\times\mathcal{D}_C} P(A=a,B=b,C=c)\log\frac{P(A=a,B=b|C=c)}{P(A=a|C=c)P(B=b|C=c)}$$

Conditional point mutual information (conditional pmi) at value (a, b) conditional upon c is defined as

$$\log_2 \frac{P(A = a, B = b|C = c)}{P(A = a|C = c)P(B = b|C = c)}$$

A good simple example of conditional mutual information is exclusive OR that measures mutual information between two random variables upon a third variable, performing better than just measuring the mutual information between the two. See Fig 2.2i.

The following properties of point mutual information will be used in our discussion.

Proposition 1. Let $I(A; B)$ be mutual information between variables $A, B \in \mathbf{X}$ over observed data $\mathcal{D}$. Then for any $a \in \{0, 1\}$, $pmi(A = a, B = 0) > 0$ if and only if $pmi(A = a; B = 1) < 0$. Furthermore, $pmi(A = a; B = 0) = 0$ if and only if $pmi(A = a; B = 1) = 0$.

Proof. We justify this claim as follows. $pmi(A = a; B = 0) = \log_2 \frac{P(A=a,B=0)}{P(A=a)P(B=0)} > 0$ implies

$$P(B = 0) < \frac{P(A = a, B = 0)}{P(A = a)} = \frac{P(A = a, B = 0)}{P(A = a, B = 0) + P(A = a, B = 1)} \tag{2.7}$$

Likewise, $pmi(A = a; B = 1) = \log_2 \frac{P(A=a,B=1)}{P(A=a)P(B=1)} \geq 0$ implies

$$P(B = 1) \leq \frac{P(A = a, B = 1)}{P(A = a)} = \frac{P(A = a, B = 1)}{P(A = a, B = 0) + P(A = a, B = 1)} \tag{2.8}$$

Putting together equation 2.7 and equation 2.8 yields

$$P(B = 0) + P(B = 1) < \frac{P(A = a, B = 0) + P(A = a, B = 1)}{P(A = a, B = 0) + P(A = a, B = 1)} = 1 \tag{2.9}$$

a contradiction to the fact that $P(B = 0) + P(B = 1) = 1$. Therefore $pmi(A = a; B = 0) > 0$ implies $pmi(A = a; B = 1) < 0$. With a similar argument, $pmi(A = a; B = 0) < 0$ implies $pmi(A = a; B = 1) > 0$.

Furthermore, it is not difficult to see that if the inequality 2.7 is replaced by the equality, the inequality 2.8 has to be changed to equality as well in order to make change the inequality in 2.9 to the equality. This concludes that $pmi(A = a; B = 0) = 0$ if and only if $pmi(A = a; B = 1) = 0$. ■

By information gain equation 2.6, since (conditional) entropy is non-negative, the highest value for mutual information between two variables is bounded by the entropy of the either variable. Therefore, we have

Proposition 2. Let A and B be two random variables with finite discrete domains $\mathcal{D}_A$ and $\mathcal{D}_B$ respectively. Then $I(A; B) \leq \min\{\log_2|\mathcal{D}_A|, \log_2|\mathcal{D}_B|\}$.

2.2 Machine Learning

Machine learning is a fast-growing scientific field noted as a data-driven knowledge acquired by computers that attempts to identify relations in example attributes and draw conclusions from this information. Based on the task, the machine learning algorithm examines different characteristics of sample attributes. Depending on the availability of labeled data, the learning process can be performed in supervised, semi-supervised, or unsupervised manner[44]. Providing proper attributes and an adequate number of samples are the minimum requirements to derive a reasonable and practical model from machine learning algorithms [8]. Often, data have extraneous and confusing attributes. Accordingly, different techniques have been explored to narrow down the essential features of a sample [64, 62, 14].

2.2.1 Problem Definition

Let $\mathbf{X} = \{X_1, ..., X_n\}$ be a set of n random variables. Let Y be the target, a function in variables $\mathbf{X}$. We assume both random variables and the target to take binary values [1]. Let D be a set of m data samples for $(\mathbf{X}, Y)$, which can be regarded as an $m \times (n + 1)$ matrix of binary values. Let subset $\mathbf{Z} \subseteq \mathbf{X} \cup \{Y\}$ and d be a sample (row) in D. We denote with $d(\mathbf{Z})$ the projection of d onto columns in $\mathbf{Z}$.

[1] For simplicity, our discussions assume binary domain. However, the proposed method also applies to non-binary, categorical domains of variables.

Given observed data D for $(\mathbf{X}, Y)$, a non-empty subset $\mathbf{S} \subseteq \mathbf{X}$ is called representative set for $\mathbf{X}$, if for every two rows d_1 and d_2 in D, $d_1(\mathbf{S}) = d_2(\mathbf{S})$ implies $d_1(Y) = d_2(Y)$. That is, the variables in $\mathbf{S}$ uniquely determine the function value of Y.

Definition 4. The *Minimum variable identification problem* (MIN VARIABLES) is defined as:
Input: sample data $\mathcal{D}$ over variables $\mathbf{X}$ and target Y ;
Output: a representative set $\mathbf{S}$ for $\mathbf{X}$ such that $|\mathbf{S}|$ is the minimum.

The well-known task of *Feature Selection* problem in machine learning can be cast as the *Min Variables* problem, where features can be quantified as random variables formulated for the problem of interest. However, the task of computing the minimum feature set is computationally intractable [22]. For example, exhaustively checking all subsets of $\mathbf{X}$ require time $\Omega(2^n)$, an exponential in the size $n = |\mathbf{X}|$.

2.2.2 Feature Selection

The subfield of machine learning that studies different techniques to choose a relevant and non-redundant feature subset from the provided feature set is feature selection. The goal of feature selection is to find the optimal subset that provides the best performance for machine learning tasks. Feature selection techniques reduce the size of problems, making many at least solvable by machine learning algorithms. Moreover, in many cases, improving the performance of machine learning algorithms initiates the idea of solving those cases using machines. Many studies have shown that providing convenient features for machine learning algorithms is critical to their performance [21, 50].

Input data generally has many features, so it is necessary to choose the essential set, either due to the impossibility of processing the whole or in an attempt to optimize the algorithm. These points emphasize the importance of using feature selection studies.

There are five general categories for feature selection techniques, which are explained below:

- **Filter based** are among the initial feature selection techniques and are very popular due to their simple implementation and cheap computation cost. (Figure 2.5)
- **Wrapper** arose primarily due to the weak performance of filter-based methods. Unlike filter-based techniques, wrappers are computationally expensive since they perform iteratively on machine learning algorithm output. (Figure 2.3)
- **Embedded** approaches attempt to avoid the disadvantages of filter-based and wrapper methods while including each of their benefits. (Figure 2.4)
- **Hybrid** Hybrid feature selection techniques combine at least two of the above techniques to obtain the optimal feature subset. The combination of feature selection techniques allows the model to benefit from the merits of each technique while compensating for its weak points. (Figure 2.7)
- **Ensemble** techniques have lately been utilized to increase confidence in feature selection. These approaches can provide a more stable final result, a serious concern in many feature selection approaches. (Figure 2.6)

Mutual Information-based Feature Selections

Many machine learning researchers utilized mutual information (MI) to select optimal feature sets. Since MI analysis has a theoretical foundation, selecting critical features could be well justified. MI measures the relevancy of a feature and could have a significant role in determining a feature inclusion/exclusion.

But considering only MI in selecting features causes issues such as overestimation of feature significance that also includes irrelevant or noisy features. In some research, MI has been used as a complementary component in the feature selection process to benefit from its potential while covering for its possible limitations. Effective approaches have been investigated in this topic [15, 47, 60, 68, 57]. Three-dimensional mutual information among features (including conditional MI, joint MI, and three-way interaction) was

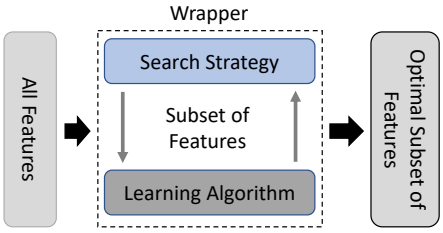


Figure 2.3: Wrapper style feature selection.

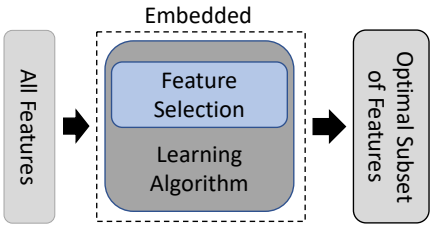


Figure 2.4: Embedded feature selection pipeline.

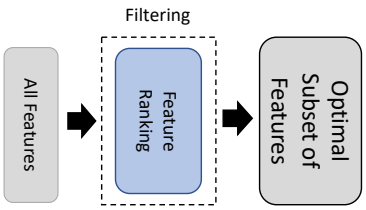


Figure 2.5: Filtered-based feature selection pipeline.

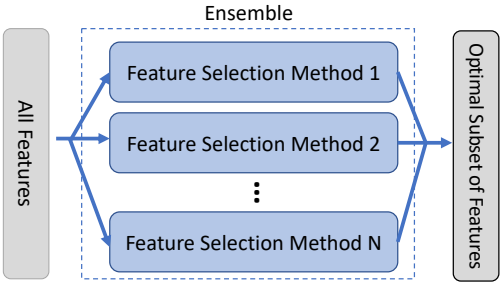


Figure 2.6: Ensemble feature selection pipeline.

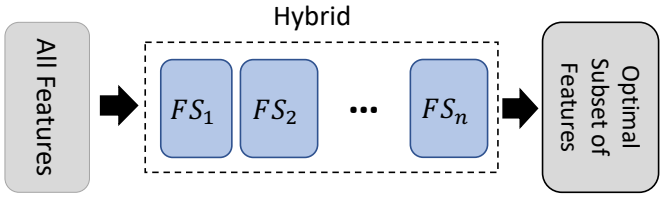


Figure 2.7: Hybrid feature selection method such that each FS_i could be a feature selection method of type filtering-based, embedded, or wrapper feature selection.

proposed to empower feature selection algorithms using MI [31].

2.2.3 Classification

Machine learning algorithms can handle many tasks. One popular usage of machine learning algorithms is predicting what class a sample belongs to. The present level of ground truth knowledge in a machine learning algorithm determines if the task will be handled in a supervised learning manner. Supervised classification could be a binary classification (where only two classes are present) or multi-class classification (where more than two classes are present).

Below, we discuss metrics that are useful in evaluating a binary classification. We denote T as a true prediction and F as a false prediction. Assuming that we have two classes, 1 annotates the first class, and 2 annotates the second one.

Metrics used for binary classification task

We use the below metrics to monitor the performance of the supervised classification task in this study.

- **Accuracy**= $\frac{T_1+T_2}{T_1+T_2+F_1+F_2}$
- **Precision**= $\frac{T_1}{T_1+F_1}$
- **Recall/Sensitivity**= $\frac{T_1}{T_1+F_2}$
- **Specificity**= $\frac{T_2}{T_2+F_1}$
- **F1 score**= $2 * \frac{Precision*Recall}{Precision+Recall}$

where T_1 is the number of correct predictions in the first class; T_2 is the number of correct predictions in the second class; F_1 is the number of incorrect predictions in the first class, and F_2 is the number of

incorrect predictions in the second class.

The main reason to monitor other metrics in addition to accuracy is that we do not want to be misled by focusing only on the correct predictions. In many cases, the data set could be imbalanced. If a minority of data were ignored in the model, then the accuracy would be insufficient to provide a comprehensive interpretation of predictions. Moreover, the cost of false classifications could vary depending on the task. For example, in the case of classifying whether a patient has or does not have cancer, a prediction of cancerous patients not having cancer could be very expensive and risk lives, while indicating healthy patients as having cancer might expose them to the side effects of many unnecessary treatments. These variations in the interpretation of predictions necessitate monitoring several metrics to gain a clear image of model performance, which can directly affect the user's policy.

2.2.4 Neural Network

Neural networks were inspired by the human brain and how information is perceived and transferred into knowledge through biological connections [44]. Many tasks are handled by neural networks, including detection [36, 48, 41], segmentation [24, 23], and classification [66, 12]. There are many types of neural networks, including convolutional neural networks and recurrent neural networks. Each is most suitable for a type of data or a specific task, or both[51]. Neural networks face some challenges in hardware utilization [18]. For example, their computation phase requires relatively more resources than other machine learning approaches, and they require a relatively large amount of data as input compared to other algorithms. Nonetheless, they have received considerable attention in studies that have confirmed their potential to address many complicated tasks. Figure 2.8 shows general architecture of a neural network when fully connected network is used.

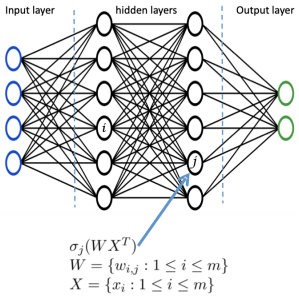


Figure 2.8: Neural Network architecture.

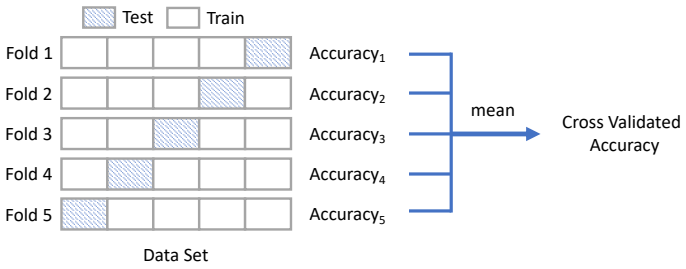


Figure 2.9: 5-Fold Cross-Validation.

2.2.5 Neural Network Sparsification

Neural networks show promising abilities to handle complicated tasks, including classifications. Not all of their connections flow useful information, and narrowing down their connectivity reduces computation cost and, in many cases, removes unnecessary computation and lets neural network show performance increase [17].

Recently neural network compression was received a great attention by researchers and that topic could be categorized into pruning and quantization techniques. Quantization focus more on reducing computation by utilizing different optimization techniques in implementation.On the other hand, pruning strategies compress the network, making it sparser and requiring less computation by filtering out criteria. Pruning strategies includes element-wise, channel-wise, shape-wise, filter-wise, layer-wise and network-wise [40]. Many pruning algorithms discard irrelevant input or/and hidden neurons [19, 25, 63, 65, 38]. This approach aims to optimize computation resource usage while providing an accurate prediction.

Mutual Information-based Neural Network Sparsification

Unlike the wide utilization of mutual information (MI) in traditional machine learning algorithms, specifically feature selection, there have been a limited number of researches to investigate MI's role in neural

network sparsification. Neural network sparsification based on MI has great potential, making it a great candidate for scholarly exploration.

A very recent study on neural network pruning proposes filtering out neurons based on their MI [26]. The study was inspired by mutual information-based feature selection in SVMs and logistic regression. The method attempts to avoid irregular memory access by squeezing the matrices. However, starting pruning from the output layer and goes back towards input carries heavy effect from the labels compared to our method. Another relevant research proposes **MINT** (Mutual Information-based Neuron Trimming) [27] that measures conditional geometric MI between adjacent layers as the importance of information flow. Their approach ensures passing useful information between layers by considering dependency between filters when pruning. Although MINT achieves good performance, focusing on individual layers causes inconsistent pruning among layers [28]. Similar to the dropout layer in the neural network, **DropMI** (an MI-based dropout) [20] was introduced to highlight essential neurons by creating a binary mask matrix based on their MI value. It is more effective compared to normal dropout (which is decided randomly), although calculating MI in each batch provides much computation load to the network.

Technical ML-tools used in this research

Here, we briefly explain some concepts in machine learning algorithms that we used in our implementation.

- **Recursive Feature Elimination (RFE)**

  It uses algorithms to rank features based on utilized criteria. In this research, we employed Super Vector Machines (SVM) as our classifier to assign each gene an importance score. This lets us acquire stability in the final derived subset [52].

- **Cross-validation**

  Many machine learning work on small data sets that inherently could lead to a miscalculation

by many machine learning algorithms. One known way to deal with this issue and gauge the generalization is to use cross-validation when splitting data and using it for training. This approach divides the data into k folds and considers $k - 1$ parts for training and one part for testing until each fold has been used for testing [45]. Figure 2.9 illustrates the functionality of cross-validation. Because of the small data size, we used cross-validation in our experiments.

– **Stratified K-Fold Cross-Validation**

  We employed stratified k-fold cross-validation in our experiments, similar to ordinary cross-validation. However, instead of random sampling, this approach conducts stratified sampling, which maintains the distribution of different classes across folds. Suppose we have a dataset of 100 samples with two classes, 80 samples from class A and 20 from class B. If random sampling is used, the dataset is split into a training set and a test set in an $8{:}2$ ratio. This could result in a training set comprising all class A samples and a test set comprising all class B samples, resulting in a weak model. In contrast, under stratified sampling, the training set consists of 64 of class 1 (80% of 80) and 16 of class 2 (80% of 20) samples for a total of 80, which represents the original dataset in equal proportion. Similarly, the test set consists of 16 negative class A (20% of 80) and 4 positive class B (20% of 20) samples, representing the entire dataset in equal proportion.

Chapter 3

Methodology

We briefly explained the background and required knowledge in the previous chapter. This chapter explains our methodology in this research.

3.1 MI-based identification of critical variables

We first offer a mutual information-based insight into the critical variables identification problem. In particular, the following theorem provides a necessary and sufficient condition to validate if a chosen subset of variables is a representative set.

Theorem 3.1.1. Let data $\mathbf{D}$ for $\mathbf{X}$, and Y, such that $P(Y = y) \neq 1$ for any $y \in \{0, 1\}$. Then $\mathbf{S} \subseteq \mathbf{X}$ is a representative set for $\mathbf{X}$, if and only if every $s \in \{0, 1\}^{|\mathbf{S}|}$ and $y \in \{0, 1\}$, $pmi(\mathbf{S} = s; Y = y) > 0$.

Proof. Assume that $\mathbf{S} \subseteq \mathbf{X}$ is a representative set for $\mathbf{X}$, over D. Then for any combination (s, y),

$$pmi(\mathbf{S} = s; Y = y) = \log_2 \frac{P(\mathbf{S} = s, Y = y)}{P(\mathbf{S} = s)P(Y = y)} > 0$$

due to the assumptions that $P(Y = y) < 1$ and that $\mathbf{s}$ is a representative for $\mathbf{X}$, which implies that $P(\mathbf{S} = s, Y = y) = P(\mathbf{S} = s)$.

On the other hand, assume that every point mutual information in $I(\mathbf{S}; Y)$ computed from $\mathcal{D}$ is positive. If for two rows d_1 and d_2 in D, $d_1(\mathbf{S}) = d_1(\mathbf{S}) = s$ implies $d_1(Y) = y_1$ and $d_1(Y) = y_2$ but $y_1 \neq y_2$, then for $i = 1, 2$,

$$pmi(\mathbf{S} = s; Y = y_i) = \log_2 \frac{P(\mathbf{S} = s, Y = y_i)}{P(\mathbf{s} = s)P(Y = y_i)} \tag{3.1}$$

Also for $i = 1, 2$, let $P(\mathbf{S} = s, Y = y_i) = p_i$, and, without loss of generality, $p_1 \leq p_2$. Then $P(\mathbf{S} = s) = p_1 + p_2$. In order for equation 3.1 to be positive for both $i = 1$ and $i = 2$, we need

$$P(Y = y_1) < \frac{p_1}{p_1 + p_2} \text{ and } P(Y = y_2) < \frac{p_2}{p_1 + p_2}$$

However, this leads to

$$1 = P(Y = y_1) + P(Y = y_2) < \frac{p_1}{p_1 + p_2} + \frac{p_2}{p_1 + p_2} = 1$$

Contradicts. So the assumption that $d_1(\mathbf{S}) = d_1(\mathbf{S}) = s$ implies $d_1(Y) = y_1$ and $d_1(Y) = y_2$ but $y_1 \neq y_2$ is wrong. That is, **S**, is a representative for **X**, over data $\mathcal{D}$. ■

We now redefine the *Min Variables*, the critical variables identification problem, and the mutual information measurement.

Definition 5. The k-Variables problem is defined as follows:

k-VARIABLES:

Input: sample data $\mathcal{D}$ over variables **X**, and target Y;

Output: representative set **S** for **X** such that $I(\mathbf{S}; Y)$ is the maximum, with $|\mathbf{S}| \leq k$.

The k-*Variables* problem is related to the original *Min Variables* by enforcing the size of the representative set to be bounded by a given parameter k. This avoids the situation that $\mathbf{S} = \mathbf{X}$ were to be returned as a trivial solution. We claim that problem k-*Variables* characterizes problem *Min Variables*. This is because if there is an algorithm $\mathcal{A}$ to solve problem k-*Variables*, we can run $\mathcal{A}$ on different values

of $k = 1, ..., n$ (with a binary search over k) to identify $\mathbf{S} \subseteq \mathbf{X}$ with the smallest $k = |\mathbf{S}|$ and all point mutual information in $I(\mathbf{S}; Y)$ are positive.

The following equation gives a relationship between a subset of variables and its incremental in terms of mutual information. It can be proved by the definition of mutual information and conditional mutual information. We omit the proof here.

Proposition 3. For any subset $\mathbf{S} \subseteq \mathbf{X}$ and subset $\mathbf{S} \cup \{X\}$, $X \in \mathbf{X} \backslash \mathbf{S}$ and target Y,

$$I(\mathbf{S} \cup \{X\}; Y) = I(\mathbf{S}; Y) + I(X; Y|\mathbf{S}) \tag{3.2}$$

Theorem 3.1.2. There is a $O(k2^k n)$-time dynamic programming algorithm for problem k-*Variables.*

Proof. Let $\mathbf{X} = \{X_1, \ldots X_n\}$. We define function $f(j, m)$ to be the maximum mutual information $I(\mathbf{S}; Y)$ for a subset $\mathbf{S}$ chosen from $\{X_1, \ldots, X_j\}$ and $|\mathbf{S}| \leq m$. Then we have the following recurrence for function f: for $1 \leq j \leq n$ and $1 \leq m \leq k$,

$$f(j, m) = \max \begin{cases} f(j-1, m-1) + I(X_j; Y|\mathbf{S}^*_{j-1,m-1}) \\ f(j-1, m) \end{cases}$$

where $\mathbf{S}^*_{j-1,m-1}$ is solution corresponding to the objective function value $f(j-1, m-1)$. The base cases for the recurrence are

$$\begin{cases} f(0, m) = 0 \quad m \geq 0 \\ f(j, 0) = 0 \quad j \geq 0 \end{cases}$$

To see that the recurrence is correct, consider variable X_j is a part of the solution $\mathbf{S}^*_{j,m}$ corresponding to the objective function $f(j, m)$. Let the rest of the variables in $\mathbf{S}^*_{j,m}$ be the $\mathbf{A}$. Then according to equation

3.2,

$$I(\mathbf{A}, X_j; Y) = I(\mathbf{A}; Y) + I(X_j; Y|\mathbf{A})$$

where $\mathbf{A}$ is actually $\mathbf{S}^*_{j-1,m-1}$.

We are able to compute dynamic programming tables to derive values for function $f(j, m)$ and its associated solution $\mathbf{S}^*_{j,m}$ for all j and m. The main table size will be $(k + 1) \times (n + 1)$. However, the computation of each cell in the table requires time factor $2^{j-1} = O(2^{k-1})$ to compute the conditional mutual information. This is basically to calculate all value combinations for variables in $\mathbf{A}$, assuming each variable has the domain $\{0, 1\}$. ■

We need to point out that, unfortunately, the above algorithm is only an approximation algorithm and may not find an optimal solution. This is due to the fact that the algorithm constructs solutions by starting from a single variable ($k = 1$) whose mutual information with the target Y is the maximum. The algorithm may miss the optimal solution containing k, e.g., $k = 2$, variables where either variable has the maximum mutual information with the target.

Nevertheless, the above algorithm has the advantage over the exhaustive search algorithm that may run in time $O(n^k)$, where n is usually very large. In the case of small values of k, the algorithm in theorem 3.1.2 is actually linear time and practically useful, especially to produce a collection of small subsets of variables, which together may form an optimal solution to the problem Min Variables and thus to the feature selection problem. In particular, it will be useful for the algorithm scheme proposed in the next section.

3.2 A 3-MI based method

Given that finding the exact (smallest) subset of critical variables is computationally intractable, approximation algorithms often resort. For example, many claimed optimal algorithms for feature selections are

heuristic, in the same spirit of theorem 3.1.2, with the solution snowballed from an initially very small subset of features of the maximum mutual information with the target. Since such solutions may potentially miss some critical variables/features, they should be considered a disadvantage in important applications, e.g., in identifying driver genes in cancer studies.

Therefore, methods that can yield a subset of variables, which include all the desired critical variables, should be more appealing. This has motivated us to look for some necessary characteristics of critical variables. In the following, we first outline our method to identify a set of critical variables and present theoretical bases for the proposed method.

We propose a new 3-MI based method for the critical variable identification problem where variables are selected based on their mutual information involved in one of the following types:

1. Variable $X_i \in \mathbf{X}$ is selected if $I(X_i; Y)$ has a high value;
2. Variables X_i, $X_j \in \mathbf{X}$ are selected if $I(X_i; X_j)$ has a high value;
3. Variables X_i, $X_j \in \mathbf{X}$ are selected if $I(X; X_j|Y)$ has a high value.

We give rationales for the choices of such variables. First, for variable X_i with a high $I(X_i; Y)$ value, while there is no guarantee it would belong to the minimum critical variable set $\mathbf{S}$, it is very likely it belongs to another critical set $\mathbf{S}'$ of a higher value of mutual information $I(\mathbf{S}'; Y)$ than $I(X_i; Y)$ based on Proposition 3. Therefore, X_i would serve as a good basis to construct a pertinent critical set.

The strategy of using the second type of mutual information presumes that variable X_i with a high value of $I(X_i; Y)$. Based on another research work in pseudo transitive of mutual information [4] if $I(X_i; Y) \geq \theta$ and $I(X_i; X_j) \geq \theta$, then $I(X_j; Y) \geq \delta_\theta$, where θ and θ_δ are not small. This strategy ensures that variable X_j is selected as well.

The third strategy relying on conditional mutual information needs an elaborated explanation, as justified in the following theorem and its proof. The consequence of $I(A; B|Y)$ being large makes it possible to select the meaningful pair of variables A and B.

Theorem 3.2.1. Let variables $A, B \in \mathbf{X}$ and target Y with observed data D. If $I(A; B) < \epsilon$ and $I(A; B|Y) \geq \theta$, where $\epsilon \ll \theta$ and θ, then $I(A, B; Y) \geq \delta_\theta$ for some $\delta_\theta \gg \epsilon$ that depends on θ.

Proof. For the convenience of discussions, we assume that D contains the same number of positive and negative samples, i.e., $P_Y(0) = P_Y(1) = \frac{1}{2}$. Given $I(A; B|Y) \geq \theta$, since

$$I(A; B|Y) = P_Y(0)I(A; B|Y = 0) + P_Y(1)I(A; B|Y = 1)$$

either $I(A; B|Y = 0) \geq \theta$ or $I(A; B|Y = 1) \geq \theta$. We assume $I(A; B|Y = 0) \geq \theta$.

The conditional mutual information $I(A; B|Y = 0)$ can be broken down as the sum

$$I(A; B|Y = 0) = \sum_{a,b} P_{AB|Y}(1, b|0) \log_2 \frac{P_{AB|Y}(1, b|0)}{P_{A|Y}(a|0)P_{B|Y}(b|0)} \tag{3.3}$$

where the rightmost term is the point mutual information, abbreviated with $pmi(a; b|0)$.

Without loss of generality, we assume that for some $b \in \{0, 1\}$, $pmi(0, b|0) > 0$. Then by Proposition 1, $pmi(1; b|0) < 0$, $pmi(0; \bar{b}|0) < 0$, and $pmi(1; \bar{b}|0) > 0$, where $\bar{b} = 0 \iff b = 1$. We obtain

$$P_{AB|Y}(0, b|0)pmi(0; b|0) + P_{AB|Y}(1, \bar{b}|0)pmi(1; \bar{b}|0) \geq I(A; B|Y = 0) \geq \theta \tag{3.4}$$

Let $p_1 = P_{AB|Y}(0, b|0)$, $p_2 = P_{AB|Y}(1, \bar{b}|0)$, $q_1 = P_{AB|Y}(0, \bar{b}|0)$ and $q_2 = P_{AB|Y}(1, b|0)$, constrained by $p_1 + p_2 + q_1 + q_2 = 1$. In order for inequality 3.4 to hold, it is necessary that $p_1, p_2 \gg q_1, q_2$. The two terms on the LHS of 3.4 are $p_i \log_2 \frac{p_i}{(p_i+q_1)(p_i+q_2)}$, for $i = 1, 2$, respectively, or expressed $\approx p \log_2 \frac{1}{p}$ in general. Given large θ, we may assume that $p_1 = \frac{1}{2} + c - \delta$, $p_2 = \frac{1}{2} - c - \delta$, for small $\delta \geq 0$ and some c.

A	B	Y	I(A;B | Y)	prob level
0	0	0	> 0	p_1 large
0	1	0	< 0	q_1 small
1	0	0	< 0	q_2 small
1	1	0	> 0	p_2 large
0	0	1	if > 0	can't be small
0	1	1	then < 0	
1	0	1	then < 0	
1	1	1	then > 0	can't be small

hypothetical scenario

Figure 3.1: Point conditional mutual information values for $I(A; B|Y)$, where the upper half table is for $I(A; B|Y = 0)$ and the lower half for $I(A; B|Y = 1)$. The part with red font indicates a hypothetical scenario that would result in $I(A; B) \not\ll \epsilon$, contradicting the given condition $I(A; B) \ll \epsilon$.

Now we examine mutual information $I(A; B|Y = 1)$. We claim that $pmi(0; b|1) < 0$.

Assume otherwise, $pmi(0; b|1) > 0$, then by Proposition 1, $pmi(1; b|1) < 0$, $pmi(0; \bar{b}|1) < 0$, and $pmi(1; \bar{b}|1) > 0$. This implies that neither $P_{AB|Y}(0, b|1)$ nor $P_{AB|Y}(1, \bar{b}|1)$ can be too small. Together with the estimated p_1 and p_2, we conclude that $I(A; B) \not\ll \epsilon$, contradicting the given assumption that $I(A; B) \ll \epsilon$. See Figure 3.1 for an illustration on the scenario where $b = 0$.

A	B	Y	I(A; B | Y)	prob level
0	0	0	> 0	p_1 large
0	1	0	< 0	q_1 small
1	0	0	< 0	q_2 small
1	1	0	> 0	p_2 large
0	0	1	< 0	
0	1	1	> 0	not small
1	0	1	> 0	not small
1	1	1	< 0	

B	Y	A	I(B; Y | A)	prob level
0	0	0	> 0	p_1 large
1	0	0	< 0	q_1 small
0	1	0	< 0	
1	1	0	> 0	not small
0	0	1	< 0	q_2 small
1	0	1	> 0	p_2 large
0	1	1	> 0	not small
1	1	1	< 0	

Figure 3.2: *left*: Point conditional mutual information values for $I(A; B|Y)$, drawn from Figure 3.1, where the portion with red font has been correct. *right*: redrawn from *left*, where point condition mutual information for $I(B; Y|A)$ is shown.

Therefore, $pmi(0; b|1) < 0$, which also implies $pmi(1; b|1) > 0$, $pmi(0; \bar{b}|1) > 0$, and $pmi(1; \bar{b}|1) < 0$. All pmis under conditions $Y = 0$ and $Y = 1$ are summarized in the table on the left in Figure 3.2,

which is then rearranged to the right table where conditional upon column A instead of Y. From the right table of Figure 3.2, we can see that $I(B;Y|A) \geq \delta_\theta$ for not small δ_θ. Then by Proposition 3, $I(A,B;Y) \geq \delta_\theta$. ■

The above logic allows us to introduce the following 3-MI based method to solve the Min Variables problem, which includes these criteria:

- Observed data $\mathcal{D}$ over a set of random variables $\mathbf{X} = \{X1, ..., X_n\}$, and target Y , percentile p;
- $\mathbf{S} = \emptyset$;
- Place all random variables X_i in **S**, if $I(X_i;Y)$ is above p percentile among all $I(X_j;Y)$, $j = 1, ..., n$;
- Place all random variables X_i and X_j in **S**, if $I(X_i;X_j)$ is above p percentile among all $I(X_k;X_l)$, where $k \neq l$, $k, l = 1, ..., n$;
- Place all random variables X_i and X_j in **S**, if $I(X_i;X_j|Y)$ is above p percentile among all $I(X_k;X_l|Y)$, where $k \neq l$, $k, l = 1, ..., n$.

In order to identify significant genes in we filter out genes based on their three different mutual information (MI): gene-gene MI (gg_mi), label-gene MI (lg_mi), and conditional MI (con_mi). The interpretation of gene-label MI and gene-gene MI are biologically different; label_gene MI gives us hints about which gene might be correlated to a label, whereas gene-gene MI or conditional MI shows us which two genes are potentially similar to each other in causing disease.

After calculating these three MIs, We filter out genes with $1 - \theta$ top values of MIs ($0 \leq \theta \leq 1$) such that we consider the union of the subsets given by MI filterings. We reduce the number of genes by between 100 and 1000 fold as features in the classification problem. Later we show that adjusting the magnitude of θ determines our prediction's reliability level. After gaining a subset of significant genes from MI filterings, we incorporate the Recursive Feature Elimination (RFE) feature selection technique

(wrapper model feature selection) to gain a stable gene subset. The pipeline of our gene selection method is shown in figure 3.3. We explain the calculation of each MI in detail in detail in chapter 4.

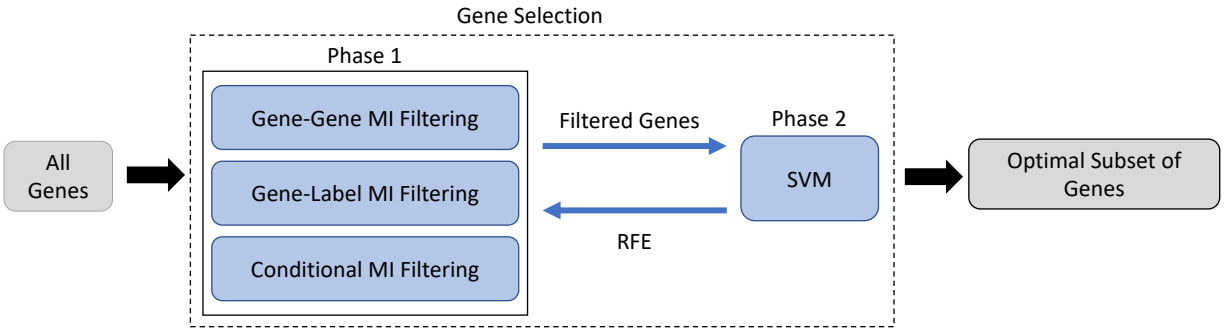


Figure 3.3: Gene selection pipeline.

3.3 Sparse Neural Network

Another approach we take in classifying microarray data is utilizing fully connected neural networks and making them sparse by removing unnecessary connections. The idea was borrowed from the gene filtering part by considering mutual information. Earlier, when filtering out genes and identifying the significant ones, we do the filtering task similar to a pre-processing task for our classifier (which is mainly neural network in our case). Now we are about to do the filtering part inside of our neural network. In order to this instead of doing the filtering for the network. Sparsification of neural networks means pruning connections based on criteria, and the main motivation for that is decreasing computations for neural networks while providing a reliable model. Pruning connections in a neural network might even compress the network and, as a result, lower memory usage significantly. Not all connections in a fully connected neural network pass useful information; thus we want to optimize the connectivity and as a result training cost. This would let resources to be spend effectively.

The idea of a recent study in [26], which considers the MI of two connected neurons and ignores neurons with low MI, is very similar to the idea of our work, except we ignore connections, not neurons.

We check all connections in a fully connected neural network to see which connections have value to keep and which are unnecessary to carry; without changing the network performance. The criteria to determine if a connection has enough value is to consider each node's output and then calculate their mutual information (MI) overall provided samples. Equation 3.5 shows our pruning strategy in a neural network where $O(i, l)$ is output for neuron i in layer l and $O(i, l)$ is output for neuron i in layer $l + 1$ for all train samples.

$$W_{ij} = \begin{cases} W_{ij}, & MI(O(i,l), O(j,l+1)) > \sigma. \\ 0, & \text{otherwise.} \end{cases} \tag{3.5}$$

Here we only leave connections with high enough MI between two neurons' output. We continue monitoring the network's performance to see the effect of removing connections on different performance metrics. Later we observe connections between significant genes left from gene filtering and important genes in sparsification which are assigned high enough MI.

Relation of *3-MI gene filtering* & *Sparsified Neural Network with Microarray Data*

We aim to find connections between first part of this research that focus on gene filtering with 3-MI and the second part that mainly focus on neural network connectivity pruning. To achieve this goal, we check all connections from provided input (with no filtering) to the first hidden layer of neural network and check which gene input has low mutual information (MI) with neurons in the first layer. We simply consider zero for connection weights of those connections with low MI from original input to the first hidden layer.

This let us observe significant genes identified by sparsified neural network. Later, we compare the gene subset gained from 3-MI gene filtering and the gene subset left from input gene set through sparsification. We expect the gene subset left from input gene set sparsification in larger size than the significant genes identified by 3-MI filtering. However seeing overlap in those subset shows this way of sparsification in neural networks directs the network to narrow down the input gene set in each layer and eventually converge to smaller subset.

In the next chapter, chapter 4 we explains the algorithms we developed based on the methodology.

Chapter 4

Algorithms

This chapter explains the algorithms we have developed based on the discussed methodology in chapter 3. We employ them in our experiments later.

4.1 Pre-processing

The original data is in a continuous format, but we bin them into discrete values to attempt to prepare them as inputs for machine learning algorithms. Many machine learning algorithms perform better with discrete input variables [61].

We discretize each cell in a microarray matrix based on the mean and standard deviation of healthy samples of that specific gene. First, we calculate each gene's mean and standard deviation among only healthy samples (creating one mean and standard deviation per gene). Eventually, each value in gene information for each patient is binned based on the distance of that gene expression value from the mean and standard deviation in healthy samples of that gene. (Algorithm 1).

Algorithm 1 Discretization

for $gene_0, gene_1, \ldots, gene_i$ **do**
  calculate $healthy\_mean_{gene_i}$ and $healthy\_std_{gene_i}$ for the specific $gene_i$
  for $patient_0, patient_1, \ldots, patient_{107}$ **do**
    considers the value of cell:
    $= 0$, if $|value-$(healthy_mean)$|<$(healthy_std)
    $= 1$, if (healthy_std) $\leq |value-$(healthy_mean)$|<$ 2(healthy_std)
    $= 2$, if 2(healthy_std) $\leq |value-$(healthy_mean)$|<$ 3(healthy_std)
    $= 3$, if $|value-$(healthy_mean)$|\geq$ 3(healthy_std)
  end for
end for

4.2 Gene-gene MI

We intend to identify those pairs of genes more relevant to each other under the assumption that two relevant genes have high mutual information.Gene_gene MI is calculated for all combinations of 2 genes. Since the values are discrete after the discretization step (explained in 4.1), the MI values are computed according to equation 2.5 (Algorithm 2).

Algorithm 2 Calculating gene-gene MI

for $gene_i \in \{gene_0, gene_1, \ldots, gene_n\}$ **do**
  for $gene_j \in \{gene_0, gene_1, \ldots, gene_n\}$ **do**
    if $gene_i != gene_j$ **then**
      MI$(gene_i, gene_j)$: Equation 2.5
    end if
  end for
end for

4.3 Label-gene MI

The dataset we are working with has labels (healthy and cancerous). Thus we utilize another type of mutual information that is between a gene and a label. Those genes with high label-gene MI have a more

significance role in accurate prediction as high label-gene shows a high correlation between those genes and a given label. To calculate label_gene MI, we consider all combination of gene and a label. Since the values are discrete, the MI calculation is according to equation 2.5(Algorithm 3).

Algorithm 3 Calculating label-gene MI

for $gene_i \in \{gene_0, gene_1, \ldots, gene_n\}$ **do**
  for $label_j \in \{label_0, label_1, \ldots, label_m\}$ **do**
    MI$(gene_i, label_j)$: Equation 2.5
  end for
end for

4.4 Gene-Gene-Label MI

The third type of mutual information we consider is the conditional one between two genes conditional upon a label. We consider conditional mutual information if two genes are not significant individually, but their collaboration, given a specific label, makes their set significant (Algorithm 4).

Algorithm 4 Calculating conditional MI

for $label_t \in \{label_0, label_1, \ldots, label_m\}$ **do**
  for $gene_i \in \{gene_0, gene_1, \ldots, gene_n\}$ **do**
    for $gene_j \in \{gene_0, gene_1, \ldots, gene_n\}$ **do**
      if $gene_i != gene_j$ **then**
        con_MI$(gene_i, gene_j | label_t)$: Equation 3
      end if
    end for
  end for
end for

4.5 3-MI Gene Selection

After calculating these three kinds of mutual information, we filter out genes to an initial gene subset (Algorithm 5). Then we use SVM-RFE (explained in 2.2.5) and acquire a stable gene subset as the significant genes set in the dataset. SVM-RFE helps us achieve a consistent significant gene subset by minimizing bias in gene filtering process.

This significant gene subset is validated in two ways: first, this subset of genes gives us the best performance; second, the significance of filtered genes is confirmed biologically through websites offering gene information like Pathway Browser Website [5]. Thus these two ways validate the subset of genes identified as significant ones are actually influential genes.

Algorithm 5 3-MI filtering

Define $\epsilon_{gene-gene}, \epsilon_{label-gene}, \epsilon_{gene-gene-label}$
Find $S_i \subset S$
Such that $S = \{gene_0, gene_1,\ldots, gene_n\}, S_i = \{gene_{S_{i0}}, gene_{S_{i1}},\ldots, gene_{S_{im}}\}, m < n$
AND $MI(gene_{S_{ip}}; gene_{S_{iq}}) > \epsilon_{gene-gene}, 0 \leq p, q \leq m$
AND $MI(gene_{S_{ip}}; label_j) > \epsilon_{label-gene}, \quad 0 \leq p \leq m$, For $label_j \in \{label_0, label_1,\ldots, label_m\}$
AND $MI(gene_{S_{ip}}; gene_{S_{iq}})|label_j) > \epsilon_{gene-gene-label}, 0 \leq p, q \leq m$, For $label_j \in \{label_0, label_1,\ldots, label_m\}$

4.6 Sparse Neural Network

We intend to sparsify our fully connected network by pruning connections based on the MI of neuron outputs of two connected neurons. In order to select unimportant connections, our method checks neurons of each layer. We assign zero to the connection weight for each connection if the MI of those neurons' outputs falls below threshold ϵ (Algorithm 6).

Algorithm 6 Neural Network Sparsification

Define ϵ
for $layer_i$ and $layer_{i+1} \in \{layer_0, layer_1, \ldots, layer_n\}$ **do**
  for $neuron_p \in \{neuron_0, neuron_1, \ldots, neuron_m\} = layer_i$ **do**
    if $neoron_p \in layer_i$ and $neuron_q \in layer_{i+1}$ are connected
      AND MI$((output(neuron_p), output(neuron_q)) < \epsilon$ **then**
      $weight_{p(i) \rightarrow q(i+1)} = 0$
    end if
  end for
end for

We also aim to observe which input genes are being kept for microarray data after sparsification, so for using algorithm 6, $layer_i$ would be the input genes and $layer_{i+1}$ would be the first hidden layer of our network (Algorithm 7).

Algorithm 7 Neural Network Sparsification - Input $\rightarrow$First Hidden Layer

Consider $input\_layer$ and $hidden\_layer_1$
for $n_p \in \{n_0, n_1, \ldots, n_m\} = input\_layer$ **do**
  if $n_p \in input\_layer$ and $n_q \in hidden\_layer_1$ are connected
    AND MI$((output(n_p), output(n_q)) < \epsilon$ **then**
    $weight_{p0 \rightarrow q1} = 0$
  end if
end for

In the next chapter, chapter 5, we talk about the data we used for our experiments.

Chapter 5

Data

5.1 Introduction

In recent years, machine learning came to help biology sciences ease their advancements and increase life conveniences. Undoubtedly, using machine learning in cancer detection is one primary interdisciplinary application marked as a life-saving research area. Early cancer diagnosis plays a remarkable role in survival rate, effectiveness, and, potentially, treatment cost reduction, but it depends heavily on accurate prediction. Without accuracy, attempting to improve the situation causes many complicated problems and might risk patients' lives. Many studies show promising results in this area using machine learning algorithms.

5.2 Microarray

Microarray data is one of the popular biology data types in biology that helps cancer detection accurately. It provides a massive amount of gene expression information for comparing tissues and treatments. The microarray data structure includes gene expression information for patients. A patient's genetic information may present as healthy or cancerous and may sometimes show different stages/types of cancer. This

type of data could be input for binary/multiclass classification tasks in a supervised manner as labels are available in this area of interest, and genes are considered features for machine learning algorithms.

Because conclusions drawn from microarray data are critical and potentially life-threatening, the data processing that leads to such conclusions is vital. Research on microarray data has received considerable scholarly interest for its potential to save many lives by providing critical information for biologists and physicians. Nonetheless, certain underlying challenges make accurate prediction difficult. First, most publicly available datasets often have a limited number of samples and vast amounts of genetic information (tens of thousands of data points). Thus, the high dimensionality of microarray data makes it an ideal candidate for a dimensionality reduction research approach. Second, samples in many datasets are imbalanced, and the small total size introduces sampling complications. Although the small sample size issue could theoretically be addressed by employing synthetic data, this approach is sometimes considered inappropriate for gene expression information. Many argue that synthetic data might mislead scientists as they may not catch underlying patterns fully. Finally, yet importantly, this interdisciplinary research area introduces the known challenges of tackling a problem in a topic that intersects multiple fields. Of course, microarray research, like all interdisciplinary research fields, needs different domain knowledge and familiarity to address each limitation. In the case of microarray data, pipeline results must be interpretable for biologists so they can identify driver genes (those genes with more impact in initiating a disease) and use this knowledge in other applicable studies.

Many studies have shown that microarrays contain redundant data, and only a relatively small portion of the genes present are critical for correct classification. This makes microarray an excellent candidate for **critical variable identification** problem to let us draw conclusions by analyzing a small subset of genes from thousands. Microarray data formatting emphasizes the importance of reducing dimensions by choosing the most representative features (genetic information) [32].

Theoretically, applying the concept of dimensionality reduction to microarray data lets us choose the most significant genes and produce accurate predictions while reducing computing costs, including time complexity and the required infrastructure. However, employing an effective dimensionality reduction technique would be challenging while simultaneously providing biological interpretation [9].

Researchers have assumed that cancerous genetic information in tissues is inherently different from healthy gene expression, meaning that classifiers can distinguish healthy from sick records. Gene expression data for a tissue could also demonstrate the growing trend of those cells, which gives us clues about where cancer might materialize and how it may progress [6].

Microarray data are inputs to our framework in our experiments. They have a matrix structure with $M \times N$ data entries. Each row represents gene expression information of N genes for a patient (sample) such that we have M patients (samples). Table 5.1 shows the data structure in microarray datasets. In our case that each patient with its actual label as cancerous or healthy; the problem is binary classification in a supervised manner. The labeling is done at the time of drawing samples from patients for the study.

Table 5.1: Data structure in microarray data. Each data entry has a continuous numeric value. $\forall i$ and $j, 1 \leq j \leq N, \forall 1 \leq i \leq M, d_{ij} \in \mathbb{R}$

	$patient_1$	$patient_2$	...	$patient_N$
$gene_1$	d_{11}	d_{12}	...	d_{1N}
$gene_2$	d_{21}	d_{22}	...	d_{2N}
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
$gene_M$	d_{M1}	d_{M2}	...	d_{MN}

We did our experiment on lung cancer [3], breast cancer [2], colon cancer [1], and other lung cancer data [58] that was pre-processed by Dr. Huang' lab [34]. All four datasets have microarray data with two classes, cancerous and healthy. Lung and breast cancer data are balanced datasets, meaning the number of samples for each class is relatively equal. The lung cancer dataset has 107 samples for patients, including 22283 gene information regarding the lung tissues. It includes 49 healthy and 58 cancerous patients. The breast cancer dataset includes 86 total patients, 43 cancerous and 43 healthy ones, and each sample has

gene information for 22283 genes for breast tissues. On the other hand, colon cancer and the other lung cancer (acquired from Dr. Huang's lab) datasets are imbalanced datasets. The colon cancer dataset has 62 samples for patients, including 2000 gene information regarding the colon cells. The second lung cancer dataset [58] includes 156 total patients, 139 cancerous and 17 healthy ones, and each sample has gene information for 12600 genes for lung tissues. Table 5.2 explains the data.

Table 5.2: Data sets

Dataset	Patients	Genes	Healthy	Cancer	Class Distribution
Colon cancer [1]	62	2000	40	22	Imbalanced
Breast cancer [2]	86	22283	43	43	Balanced
Lung cancer [3]	107	22283	49	58	Balanced
Lung cancer [58]	156	12600	17	139	Imbalanced

5.3 Data Analysis

This part provides more information about each dataset and reviews their attributes. Also, we analyze each dataset, including those after the discretization of their entries.

5.3.1 Lung Cancer Dataset [3]

Tobacco smoking has a great role (over 90%) in lung cancer disease, and acute molecular alterations caused by smoking in the lungs could develop cancer. Therefore researchers [33] targeted studying gene expressions of lung tissues acquired from chips on 135 fresh frozen samples in three main subject categories: previously smoker, never smoker, and current smoker.

The final dataset resulting has 107 sample expression values from 58 tumor and 49 non-tumor tissues from 20 never smokers, 26 former smokers, and 28 current smokers. The dataset was annotated and published publicly in 2008. Figure 5.1 shows a snapshot of the data.

	Patient_1	Patient_2	Patient_3	Patient_4	Patient_5		Patient__103	Patient_104	Patient_105	Patient_106	Patient__107
Gene_1	10.9270836	10.416978	10.6285381	10.1511797	GSM254630		10.4676932	10.9027785	10.8694022	10.2922849	10.40721752
Gene_2	6.89521651	6.9248556	7.55024499	6.69955669	GSM254631		6.79574994	6.83816182	6.62836349	6.79404954	6.358409625
Gene_3	8.11019039	7.76022798	7.9746764	7.7126764	GSM254632		7.85545661	8.01042793	7.88901889	8.16326622	7.973843751
Gene_4	9.4512861	9.52094251	9.80759698	9.52208736	GSM254633		9.64523947	9.87185063	9.86798792	9.82480077	9.850144101
Gene_5	4.81447671	4.71864027	4.90516304	4.81807643	GSM254634		4.75957087	4.78877362	4.96762649	4.81747449	5.128891631
Gene_22279	4.96644196	5.05467303	5.4005882	5.1417585	5.15120252		5.19268877	5.0710665	5.43245196	5.19477605	5.295231812
Gene_22280	4.454958	4.46100475	4.64221221	4.60006147	4.58001009		4.39180677	4.46944124	4.69387357	4.55890616	4.627642905
Gene_22281	4.03497864	4.04399008	4.06977349	3.99339817	4.06665715		4.09330634	4.09605117	4.24647373	4.09805442	4.140667993
Gene_22282	4.50350726	4.54011191	4.70001742	4.61119212	4.63296052		4.54655576	4.62490518	4.75321154	4.70407869	4.762314262
Gene_22283	4.39684767	4.46478081	4.71422513	4.57973243	4.63396236		4.63241182	4.62770799	4.88187754	4.69711887	4.736233287
label	1	0	1	0	1		0	1	1	0	0

Figure 5.1: A snapshot of microarray lung cancer dataset [3] including 22283 genes and 107 patients.

5.3.2 Breast Cancer Dataset [2]

Any genome information change in breast tissues could indicate breast cancer, so gene expressions of breast tissues in two groups of healthy and cancerous patients from multi races (Malays, Chinese and Indian) in different ages (30-79) were collected to study the potential impact of breast cancer in gene expressions. Their research could identify 33 significant expressed genes in the tumor vs. normal groups [46]. Figure 5.2 shows how the data looks.

5.3.3 Colon Cancer Dataset [1]

This dataset contains 2000 gene expressions for 62 patients in a microarray format. A clustering algorithm was used to separate healthy from cancerous data. The work in [7] could decrease number of involved genes to 500 in correct clustering. Figure 5.3 shows a snapshot of the data.

	Patient_1	Patient_2	Patient_3	Patient_4	Patient_5		Patient_82	Patient_83	Patient_84	Patient_85	Patient__86
Gene_1	1881.8	2317.51	1553.86	1915.57	1240.13		2993.61	1467.17	5501.41	2387.61	4809.98
Gene_2	78.0658	61.354	80.0525	79.8518	104.933		68.1142	104.309	69.8169	113.436	189.753
Gene_3	1299.98	775.547	1103.74	762.005	820.822		2026.06	1583.96	1603.27	2224.05	1357.03
Gene_4	3086.72	2335.15	3139.65	2338.48	2555.46		5061.03	3814.23	4844.52	5158.76	4454.04
Gene_5	353.89	303.653	523.873	222.552	401.749		1092.15	704.721	1007.8	698.488	836.205
Gene_22279	198.094	100.051	86.2519	87.6575	114.637		994.089	201.812	200.178	283.145	167.103
Gene_22280	77.9246	76.6267	92.2685	90.7732	66.2097		289.3	148.928	415.189	120.866	152.823
Gene_22281	5.40687	7.38941	8.36922	9.21324	10.5411		25.0605	18.3144	20.3891	26.3743	14.7864
Gene_22282	27.9838	19.484	35.8309	38.6235	52.1979		61.3352	44.8726	186.758	83.3411	51.9849
Gene_22283	130.756	35.2956	85.5188	65.0371	84.51		173.277	93.3868	64.3201	114.721	149.533
label	Normal	Cancer	Normal	Cancer	Normal		Cancer	Normal	Cancer	Normal	Cancer

Figure 5.2: A snapshot of microarray breast cancer dataset [2] including 22283 genes and 86 patients

	Patient_1	Patient_2	Patient_3	Patient_4	Patient_5		Patient_58	Patient_59	Patient_60	Patient_61	Patient_62
Gene_1	8589.4163	9164.2537	3825.705	6246.4487	3230.3287		4972.1662	9112.3725	6730.625	6234.6225	7472.01
Gene_2	5468.2409	6719.5295	6970.3614	7823.5341	3694.45		4173.9182	6824.4864	3472.125	4005.3	3653.9341
Gene_3	4263.4075	4883.4487	5369.9688	5955.835	3400.74		3668.5338	5982.8463	2559.4625	3093.675	2728.2162
Gene_4	4064.9357	3718.1589	4705.65	3975.5643	3463.5857		1567.5554	3147.0429	2624.6893	3183.0857	3494.4804
Gene_5	1997.8929	2015.2214	1166.5536	2002.6131	2181.4202		1570.4405	4847.3083	1596.2179	1795.3107	2404.6655
Gene_1996	39.667857	85.033333	224.62024	67.710714	223.35952		95.442857	143.87738	124.25357	180.94167	269.4369
Gene_1997	67.82875	152.195	31.225	48.33875	73.09875		31.1875	8.99625	96.465	68.93375	67.8625
Gene_1998	75.6775	186.5675	42.65625	42.52	57.59875		57.1525	106.87875	133.52125	118.20125	77.215
Gene_1999	83.5225	44.4725	16.0925	49.9825	7.48875		13.96	23.2025	93.09875	32.6875	49.8625
Gene_2000	28.70125	16.77375	15.15625	16.085	31.8125		10.5475	32.16625	7.4325	23.265	39.63125
label	0	1	0	1	0		0	0	1	0	1

Figure 5.3: A snapshot of microarray colon cancer dataset[1] including 2000 genes and 62 patients.

5.3.4 Lung cancer Dataset [58]

The dataset contains microarray data of lung tissues. The pre-processed data was provided to us by Dr. Xiuzhen Huang's lab [34] at the Center for No-Boundary Thinking (CNBT) at the Department of Computer Science, Arkansas State University, AR. The dataset consists of 139 cancer samples and 17 normal samples. Figure 5.4 shows a snapshot of the data.

	Patient_1	Patient_2	Patient_3	Patient_4	Patient_5		Patient_152	Patient_153	Patient_154	Patient_155	Patient_156
Gene_1	-12.7	-16.5133	-16.17	19.31	-4.07		-30.91	38.8	-19.78	-19.07	-13.9
Gene_2	-3.83	0.836667	2.75	16.2	4.22		-27.44	63.78	10.03	0.72	0.28
Gene_3	-10.04	3.88333	1.76	33.85	-6.83		-24.67	58.31	3.7	-5.88	-14.69
Gene_4	15.68	30.7067	16.7	39.04	2.15		3.07	88.24	14.54	-3.24	7.37
Gene_5	-34.87	-4.73333	-17.17	5.81	-9.59		-39.92	46.34	-8.04	-35.53	-6.81
Gene_12596	-4.71	4.30667	2.75	22.43	-1.31		-23.28	62.9	5.51	-2.58	-9.96
Gene_12597	29.86	42.1533	69.48	51.5	50.06		-0.4	82.31	47.03	44.42	17.62
Gene_12598	-8.26	10.0767	-20.16	6.85	-37.8		41.94	85.825	-14.36	-57.88	41.24
Gene_12599	-58.83	-50.7633	-44.06	-10.81	-74.77		-99.52	23.225	-73.1	-80.85	-28.87
Gene_12600	16.57	69.2567	88.4	85.77	51.46		2.37	91.19	56.06	37.78	107.37
label	0	1	1	1	1		1	1	1	1	1

Figure 5.4: A snapshot of microarray lung cancer dataset including 12600 genes and 156 patients. [58]

Chapter 6

Experiments and Results

6.1 Effectiveness of 3-MI-based gene filtering

We compared the result of classification with our 3-MI gene selection with filtering genes with a recent study [43]that used Analysis of Variance (ANOVA), which is a statistical test used to analyze the difference between the means of more than two groups). They also used a multi-layer perceptron neural network as a classifier. The results are shown in tables 6.1, 6.2, 6.3.

Table 6.1: Classification of microarray lung cancer dataset including 22283 genes [3] with NN

Filtering Method	#Genes	Accuracy(%)	Precision(%)	Recall(%)	Specificity(%)	F1-score
Ours	222	**95.28**	**95.25**	96.36	**93.77**	**0.95**
Gene selection using ANOVA [43]	222	77	62.5	1	62.5	0.77

Table 6.2: Classification of microarray breast cancer dataset including 22283 genes [2] with NN

Filtering Method	#Genes	Accuracy(%)	Precision(%)	Recall(%)	Specificity(%)	F1-score
Ours	243	**76.86**	**75.34**	81.67	**71.94**	0.78
Gene selection using ANOVA [43]	243	73.73	75	85.71	51	0.78

Table 6.3: Classification of microarray colon cancer dataset including 2000 genes [1] with NN

Filtering Method	#Genes	Accuracy(%)	Precision(%)	Recall(%)	Specificity(%)	F1-score
Ours	120	**70.76**	**74.97**	**85**	**43**	**0.79**
Gene selection using ANOVA [43]	120	51	60	60	34	0.6

As we observe, our 3-MI gene filtering outperforms state of the art in gene selection which is ANOVA in classification with neural networks [43].

We also confirm the significance of our detected subset of genes biologically through websites offering gene information like Pathway Browser Website [5].

In order to support the functionality of our gene selection method, we use other classifiers (except neural networks) for classifying microarray cancer datasets. Result are shown in table 6.4.

Table 6.4: Using other classifiers before and after applying the MI-based filtering method of ours

Classifier	Dataset	#Gene	Accuracy	Precision	Recall	F 1 -score
Random Forest (n= 12)	Lung cancer [3]	22283 **223**	0.94 **0.97**	0.90 **0.95**	1 1	0.94 **0.97**
Random Forest (n= 20)	Breast cancer [2]	22283 **1638**	0.79 **0.86**	0.78 **0.85**	0.78 **0.85**	0.78 **0.85**
Random Forest (n= 15)	Colon cancer [1]	2000 **120**	0.62 **0.67**	0.64 **0.67**	0.64 **0.73**	0.64 **0.7**
Random Forest (n= 15)	Lung cancer[58]	12600 **973**	0.96 **0.98**	0.96 **0.98**	0.99 **1**	0.98 **0.99**
SVM (kernel='poly', degree = 3)	Lung cancer [3]	22283 **223**	0.88 **0.94**	1 1	0.78 **0.89**	0.87 **0.94**
SVM (kernel ='rbf')	Breast cancer [2]	22283 **243**	0.86 **0.93**	0.92 **0.93**	0.78 **0.93**	0.87 **0.93**
SVM(kernel='poly', degree= 6)	Colon cancer [1]	2000 **120**	0.58 **0.63**	0.61 **0.64**	0.90 **0.97**	0.73 **0.77**
SVM(kernel='poly', degree= 3)	Lung cancer [58]	12600 **1951**	0.89 **0.92**	0.89 **0.92**	1 1	0.94 **0.96**

As the result in table 6.4 supports, our proposed gene filtering method allows other machine learning methods to achieve higher performance than unfiltered gene sets as well. The performance results from the classification with other classifiers with proper setting shows our proposed filtering method outperforms the unfiltered gene set.

We also investigate the effect of different types of MI individually on the filtering. Our proposed method consists of 3 different types of MI, but we consider each MI contribution individually to the performance. The results are shown in figure 6.1 for microarray data of lung cancer [3].

- 3-MI refers to the one participating all 3 types of MI in filtering.
- mi_con refers to the conditional mutual information between two genes that identify those two genes working together.
- mi_gg refers to gene-gene mutual information that identifies genes with more interaction with each other.
- mi_lg refers to label-gene mutual information that identifies genes with high correlation with a label.

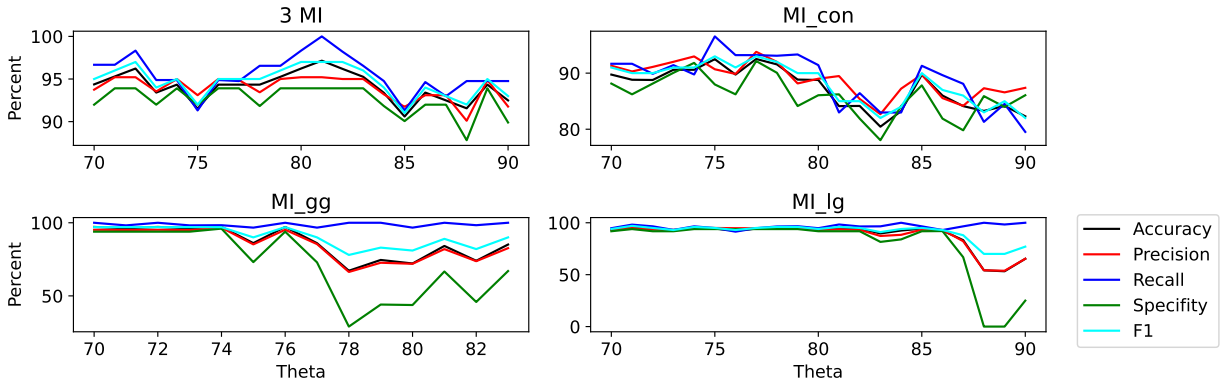


Figure 6.1: Contribution of different MI(s) filtering to the performance on microarray data of lung cancer with 22283 genes [3]

6.2 MI-based Neural Network Sparsification

To verify the effectiveness of neural network sparsification, we develop a prototype that trains a model for 3 epochs, then freezes the weights. We identify which connection weights should be removed and fix them to zero based on the mutual information between the output of incoming and outgoing neurons. Then we define an empty model with exact architecture of the previous one and load customized weights into the model. The same process is repeated with 14 different thresholds sharing one baseline model. Then we continue training the baseline along with sparsified models for 1024 more epochs and compare their performance. We repeat these steps 5 times and average all the metrics to avoid unintentional bias.

We provide experimental results from a sparse neural network (that only keeps connections with $N\%$ top MI, the rest's weights are zero) performance compared to a fully connected network. Accuracy, precision, recall, specificity, and f1-measure (explained in 2.2.3) are reported. The results are shown in figures 6.2, 6.3, 6.4, 6.5.

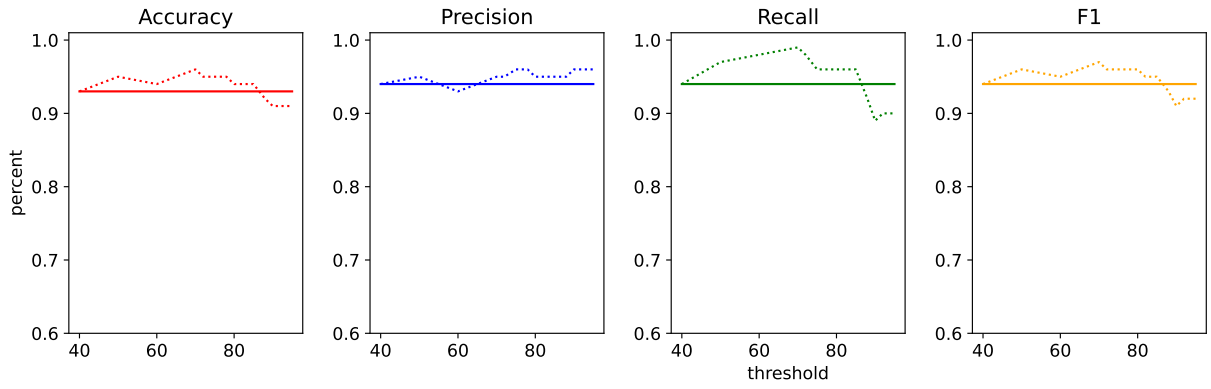


Figure 6.2: Sparsified NN vs fully connected NN with microarray breast cancer dataset [2] with 22283 genes. The solid line represents the baseline model and the dotted line is the sparsified models at different thresholds.

To explore if the network optimization along with mutual information (MI) based sparsification can filter out important (relevant) genes; we implemented sparsification between input genes and the first hidden layer. In this setup, the rest of the network was left untouched to train as normal, but the

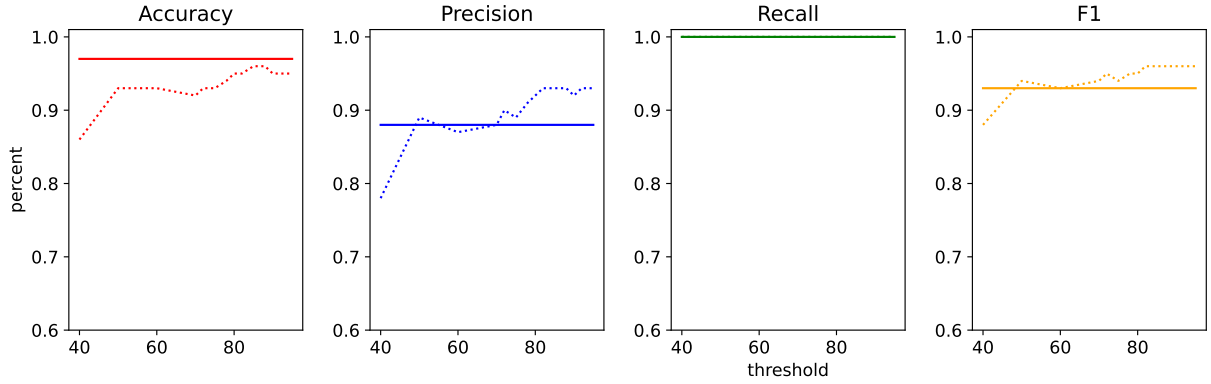


Figure 6.3: Sparsified NN vs fully connected NN with microarray lung cancer dataset [3] with 22283 genes. The solid line represents the baseline model and the dotted line is the sparsified models at different thresholds.

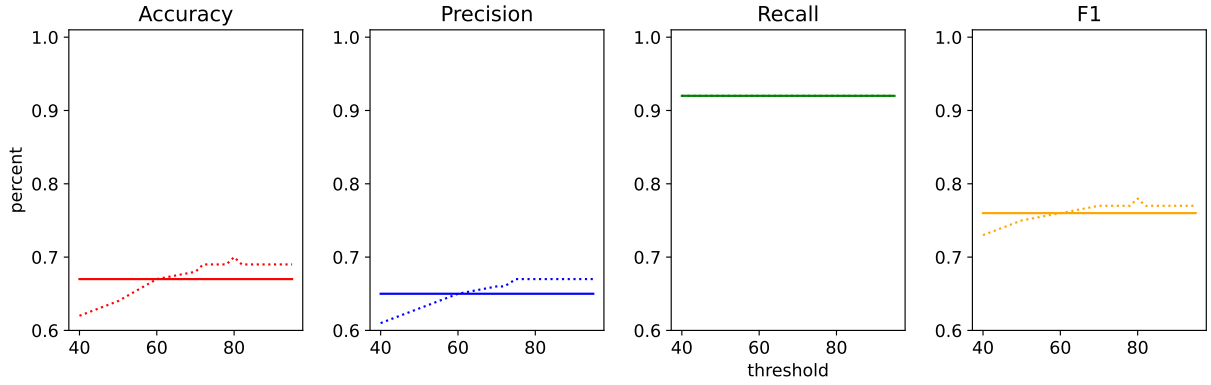


Figure 6.4: Sparsified NN vs fully connected NN with microarray of colon cancer dataset [1] with 2000 genes. The solid line represents the baseline model and the dotted line is the sparsified models at different thresholds.

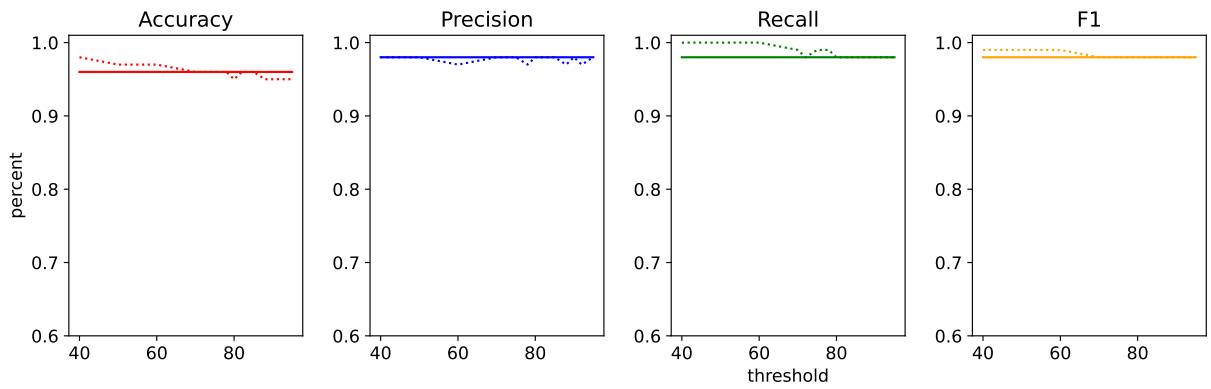


Figure 6.5: Sparsified NN vs fully connected NN with microarray of lung cancer dataset [58] with 12600 genes. The solid line represents the baseline model and the dotted line is the sparsified models at different thresholds.

connections between the inputs and the first hidden layer were sparsified. To find the importance of a gene, we calculate the sum of the absolute value of weights for all the outgoing connections from that gene. We observe $70\% - 75\%$ of genes from the input layer have below threshold MIs and hence can be removed. Table 6.5 shows how many genes were ignored directly when feeding into the first hidden layer. This experiment proves, neural network sparsification based on MI, can highlight the important genes in an explainable manner even when only one layer in a larger neural network is considered.

Table 6.5: Relations of Sparsified NN with filtering genes in different datasets

Dataset	#Genes	#Genes after filtering	#Genes in first layer of SNN	Overlap
Microarray colon cancer dataset [1]	2000	120	500	87 (72.5%)
Microarray breast cancer dataset [2]	22283	243	6000	186 (76%)
Microarray lung cancer dataset [3]	22283	224	5000	211 (94%)

Chapter 7

Discussion and Conclusion

7.1 Gene filtering

One of the highest mortality rates belongs to cancer diseases, and an early while an accurate diagnosis of cancer could save many lives and resources since it might affect treatment effectiveness. Most modern artificial intelligence models are not explainable, which limits their applicability, especially around healthcare applications. Identifying essential genes lets us avoid unnecessary computation of thousands of genes in the prediction. As a result, the gene selection process could lead to performance improvement while saving computing resources. However, the most significant attribute of these reductions is that they are fully explainable and backed by biological expertise. Our filtering approach makes the diagnosis process faster and could decrease the mortality rate of cancer disease. Comparison with the state-of-the-art techniques shows more reliability using our method, which is considered critical in predicting cancer diseases.

In this dissertation, we used different types of MI (gene-gene MI (gg_mi), gene-label MI (lg_mi), and conditional MI (con_mi)) to find the most relevant genes. Although in our current experiments, we aim for binary classification, the concept could be fully expanded to classify multi-class datasets like classifying different stages of cancer disease.

The comparison shows our filtering method works much stronger than the one they used in a recent study that utilized ANOVA and has neural network as the classifier [43].

On the lung cancer dataset [3], both our filtering method and the one used in [43] by ANOVA extracted 222 genes. Our filtering method shows better performance in accuracy, precision, specificity, and f1-score as the result are shown in table 6.1.

On the breast cancer dataset [2], both our filtering method and the one used in [43] by ANOVA extracted 243 genes. Our filtering method shows better performance in accuracy, precision, specificity, and f1-score as the result is shown in table 6.2.

On the colon cancer dataset [1], both our filtering method and the one used in [43] by ANOVA extracted 120 genes from 1200. Our filtering method shows better performance in accuracy, precision, recall specificity, and f1-score as the result are shown in table 6.3.

Getting biological confirmation through Pathway Browser [5] gives 95%-100% endorsement of chosen gene subset by our method.

The experiments also demonstrate that filtered genes allow others ML methods to achieve higher performance than unfiltered datasets.

In the comparison of the effect of different MIs' (gene-gene MI (mi_gg), gene-label MI (mi_lg) and conditional MI (mi_con)) contribution to the performance, results show that gene-gene MI (mi_gg) is the most influential; however, it could be inferred that the other two MIs provide complementary information thus the combination of three has the best outcome (as the plot of 3-MI presents). This justifies that our 3-MI method is effective.

7.2 Sparse Neural network

As the average trend shows, sparsifying the network by removing connections with lower MI values improves the performance in classification sometimes, while it doesn't cause much loss anyway. This means neural networks prioritize connections where the input and output neurons have higher mutual information, meaning those neurons carry more data. This approach definitely increases relevancy while decreasing redundancy in a classification problem (Figures 6.2, 6.4, 6.3,6.5).

Although our experiments were mainly carried out with microarray data, based on the performance comparison, we conclude that this approach of neural network sparsification would be an efficient measure in pruning models, especially when the input data has too many features and only a sufficient subset of features offers a reasonable model.

We observe that about $75\% - 80\%$ of genes are completely ignored just at the beginning due to their low MI. Table 6.5 shows how many genes were ignored directly when feeding into the first layer.

7.3 Conclusion

We proposed two explainable information theory-based research approaches for dealing with high-dimension data in the supervised classification task. We utilized microarray cancer data to show the functionality of our methods. Microarray cancer data are matrix-like data $M \times N$ where M denotes the number of genes, N is the number of patients, and each element of the matrix shows gene expression of a specific patient associated with a known gene (the structure of microarray cancer data is shown in 5.1).

The first approach we propose is based on 3 different kinds of mutual information among provided attributes. It filters out the most significant attributes based on 3 types of MI. That would be gene-gene,

gene-label, and gene-gene-label MIs for microarray cancer datasets.

The second approach focuses on optimizing neural network architecture (as the classifier) and pruning connectivity inside the network based on Mi between neuron outputs. We also achieved improvements on some metrics for certain datasets. The results reveal a trade-off and, depending on the task sensitivity needs to adjusted.

As our experiments show, the 3-MI method is very effective and offers considerable improvement for the microarray cancer data classification task. Due to the inherent properties of mutual information, it well navigates the classifier to include relevant attributes and ignore redundant ones. Microarray cancer data originally provides thousands of gene information which, not all of them are informative for the classifier, and our 3-MI method filters the most significant gene subset to do the classification task. Gene filtering shows performance improvement, which could be biologically explained.

On the sparsified neural network, leveraging the power of mutual information enables us to optimize connectivity inside the neural network to improve the outcome. In the experiments on microarray cancer data, up to 90% of neural network connections were removed while improving on some metrics on certain datasets, sacrificing other measures. While the main concept of our proposal is valid, this would be a task-related or data-dependent issue that needs to be addressed by customizing the methodology.

To further expand on these directions, the following research areas can be explored:

- Apply 3-MI to non-binary classification where more than two classes are present.
- Apply 3-MI to data other than cancer research like a social network.
- Apply the current approach to further study the more pertinent relationship between MI and NN sparsification.

Bibliography

[1] Gene expression of colon tissues. 1999. URL: http://genomics-pubs.princeton.edu/oncology/affydata/index.html.

[2] Expression data from human breast tumors and their paired normal tissues. 2009. URL: https://www.ncbi.nlm.nih.gov/geo/query/acc.cgi?acc=GSE15852.

[3] Gene expression of cigarette smoking and its role in lung adenocarcinoma development and survival. 2018. URL: https://www.ncbi.nlm.nih.gov/geo/query/acc.cgi?acc=GSE10072.

[4] Study by Sixiang Zhang in RNA informatics lab of CS department at University of Georgia under supervision of Dr. Liming Cai. 2022.

[5] Pathway Browser website. URL: https://reactome.org/PathwayBrowser/.

[6] TaeJin Ahn et al. "Deep learning-based identification of cancer or normal tissue using gene expression data". In: *2018 IEEE international conference on bioinformatics and biomedicine (BIBM)*. IEEE. 2018, pp. 1748–1752.

[7] U. Alon et al. "Broad patterns of gene expression revealed by clustering analysis of tumor and normal colon tissues probed by oligonucleotide arrays". In: *Proceedings of the National Academy of Sciences* 96.12 (1999), pp. 6745–6750. DOI: 10.1073/pnas.96.12.6745. eprint: https://www.pnas.org/doi/pdf/10.1073/pnas.96.12.6745. URL: https://www.pnas.org/doi/abs/10.1073/pnas.96.12.6745.

[8] Ahmad Alwosheel, Sander van Cranenburgh, and Caspar G Chorus. "Is your dataset big enough? Sample size requirements when using artificial neural networks for discrete choice analysis". In: *Journal of choice modelling* 28 (2018), pp. 167–182.

[9] Julia Amann et al. "Explainability for artificial intelligence in healthcare: a multidisciplinary perspective". In: *BMC Medical Informatics and Decision Making* 20.1 (2020), pp. 1–9.

[10] M Augasta and Thangairulappan Kathirvalavakumar. "Pruning algorithms of neural networks—a comparative study". In: *Open Computer Science* 3.3 (2013), pp. 105–115.

[11] Roberto Battiti. "Using mutual information for selecting features in supervised neural net learning". In: *IEEE Transactions on neural networks* 5.4 (1994), pp. 537–550.

[12] William G Baxt. "Application of artificial neural networks to clinical medicine". In: *The lancet* 346.8983 (1995), pp. 1135–1138.

[13] David G Beer et al. "Gene-expression profiles predict survival of patients with lung adenocarcinoma". In: *Nature medicine* 8.8 (2002), pp. 816–824.

[14] Peter N. Belhumeur, Joao P Hespanha, and David J. Kriegman. "Eigenfaces vs. fisherfaces: Recognition using class specific linear projection". In: *IEEE Transactions on pattern analysis and machine intelligence* 19.7 (1997), pp. 711–720.

[15] Mohamed Bennasar, Yulia Hicks, and Rossitza Setchi. "Feature selection using joint mutual information maximisation". In: *Expert Systems with Applications* 42.22 (2015), pp. 8520–8532.

[16] Rajendra Rana Bhat, Vivek Viswanath, and Xiaolin Li. "DeepCancer: detecting cancer through gene expressions via deep generative learning". In: *arXiv preprint arXiv:1612.03211* (2016).

[17] Davis Blalock et al. "What is the state of neural network pruning?" In: *Proceedings of machine learning and systems* 2 (2020), pp. 129–146.

[18] Maurizio Capra et al. "Hardware and software optimizations for accelerating deep neural networks: Survey of current trends, challenges, and the road ahead". In: *IEEE Access* 8 (2020), pp. 225134–225180.

[19] Giovanna Castellano, Anna Maria Fanelli, and Marcello Pelillo. "An iterative pruning algorithm for feedforward neural networks". In: *IEEE transactions on Neural networks* 8.3 (1997), pp. 519–531.

[20] Jie Chen et al. "Mutual information-based dropout: Learning deep relevant feature representation architectures". In: *Neurocomputing* 361 (2019), pp. 173–184.

[21] Zheng Chen et al. "Feature selection may improve deep neural networks for the bioinformatics problems". In: *Bioinformatics* 36.5 (2020), pp. 1542–1552.

[22] Scott Davies and Stuart Russell. "NP-completeness of searches for smallest possible feature sets". In: *AAAI Symposium on Intelligent Relevance*. AAAI Press Menlo Park. 1994, pp. 37–39.

[23] Raunak Dey and Yi Hong. "CompNet: Complementary segmentation network for brain MRI extraction". In: *International Conference on Medical Image Computing and Computer-Assisted Intervention*. Springer. 2018, pp. 628–636.

[24] Raunak Dey, Zhongjie Lu, and Yi Hong. "Diagnostic classification of lung nodules using 3D neural networks". In: *2018 IEEE 15th international symposium on biomedical imaging (ISBI 2018)*. IEEE. 2018, pp. 774–778.

[25] Andries P Engelbrecht. "A new pruning heuristic based on variance analysis of sensitivity information". In: *IEEE transactions on Neural Networks* 12.6 (2001), pp. 1386–1399.

[26] Chun Fan et al. "Layer-wise model pruning based on mutual information". In: *arXiv preprint arXiv:2108.12594* (2021).

[27] Madan Ravi Ganesh, Jason J Corso, and Salimeh Yasaei Sekeh. "Mint: Deep network compression via mutual information-based neuron trimming". In: *2020 25th International Conference on Pattern Recognition (ICPR)*. IEEE. 2021, pp. 8251–8258.

[28] Madan Ravi Ganesh et al. "Slimming neural networks using adaptive connectivity scores". In: *IEEE Transactions on Neural Networks and Learning Systems* (2022).

[29] Iffat A Gheyas and Leslie S Smith. "Feature subset selection in large dimensionality domains". In: *Pattern recognition* 43.1 (2010), pp. 5–13.

[30] Todd R Golub et al. "Molecular classification of cancer: class discovery and class prediction by gene expression monitoring". In: *science* 286.5439 (1999), pp. 531–537.

[31] Xiangyuan Gu et al. "A Feature Selection Algorithm Based on Equal Interval Division and Conditional Mutual Information". In: *Neural Processing Letters* 54.3 (2022), pp. 2079–2105.

[32] Moshood A Hambali, Tinuke O Oladele, and Kayode S Adewole. "Microarray cancer feature selection: review, challenges and research directions". In: *International Journal of Cognitive Computing in Engineering* 1 (2020), pp. 78–97.

[33] Judith N Haslett et al. "Gene expression profiling of Duchenne muscular dystrophy skeletal muscle". In: *Neurogenetics* 4.4 (2003), pp. 163–171.

[34] Xiuzhen Huang. *personal communication*. June 22, 2022. URL: `http://myweb.astate.edu/xhuang/`.

[35] Weikuan Jia et al. "Feature dimensionality reduction: a review". In: *Complex & Intelligent Systems* (2022), pp. 1–31.

[36] Asif Iqbal Khan, Junaid Latief Shah, and Mohammad Mudasir Bhat. "CoroNet: A deep neural network for detection and diagnosis of COVID-19 from chest x-ray images". In: *Computer methods and programs in biomedicine* 196 (2020), p. 105581.

[37] Solomon Kullback and Richard A Leibler. "On information and sufficiency". In: *The annals of mathematical statistics* 22.1 (1951), pp. 79–86.

[38] Philippe Lauret, Eric Fock, and Thierry Alex Mara. "A node pruning algorithm based on a Fourier amplitude sensitivity test method". In: *IEEE transactions on neural networks* 17.2 (2006), pp. 273–293.

[39] Zejun Li et al. "Semi-supervised maximum discriminative local margin for gene selection". In: *Scientific reports* 8.1 (2018), pp. 1–11.

[40] Tailin Liang et al. "Pruning and quantization for deep neural network acceleration: A survey". In: *Neurocomputing* 461 (2021), pp. 370–403.

[41] Paulo J Lisboa and Azzam FG Taktak. "The use of artificial neural networks in decision support in cancer: a systematic review". In: *Neural networks* 19.4 (2006), pp. 408–415.

[42] David JC MacKay, David JC Mac Kay, et al. *Information theory, inference and learning algorithms*. Cambridge university press, 2003.

[43] Subhasree Majumder et al. "Performance Analysis of Deep Learning Models for Binary Classification of Cancer Gene Expression Data". In: *Journal of Healthcare Engineering* 2022 (2022).

[44] Tom M Mitchell and Tom M Mitchell. *Machine learning*. Vol. 1. 9. McGraw-hill New York, 1997.

[45] Charles I Mosier. "I. Problems and designs of cross-validation 1". In: *Educational and Psychological Measurement* 11.1 (1951), pp. 5–11.

[46] Ivyna Bong Pau Ni et al. "Gene expression patterns distinguish breast carcinomas from normal breast tissues: the Malaysian context". In: *Pathology-Research and Practice* 206.4 (2010), pp. 223–228.

[47] Hanchuan Peng, Fuhui Long, and C. Ding. "Feature selection based on mutual information criteria of max-dependency, max-relevance, and min-redundancy". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 27.8 (2005), pp. 1226–1238. DOI: 10.1109/TPAMI.2005.159.

[48] N Pradhan, PK Sadasivan, and GR Arunodaya. "Detection of seizure activity in EEG by an artificial neural network: A preliminary study". In: *Computers and Biomedical Research* 29.4 (1996), pp. 303–313.

[49] Russell Reed. "Pruning algorithms-a survey". In: *IEEE transactions on Neural Networks* 4.5 (1993), pp. 740–747.

[50] Beatriz Remeseiro and Veronica Bolon-Canedo. "A review of feature selection methods in medical applications". In: *Computers in biology and medicine* 112 (2019), p. 103375.

[51] Stuart J Russell. *Artificial intelligence a modern approach*. Pearson Education, Inc., 2010.

[52] Hector Sanz et al. "SVM-RFE: selection and visualization of the most relevant features through non-linear kernels". In: *BMC bioinformatics* 19.1 (2018), pp. 1–18.

[53] Claude Elwood Shannon. "A mathematical theory of communication". In: *The Bell system technical journal* 27.3 (1948), pp. 379–423.

[54] Dinesh Singh et al. "Gene expression correlates of clinical prostate cancer behavior". In: *Cancer cell* 1.2 (2002), pp. 203–209.

[55] Alexander Statnikov et al. "GEMS: a system for automated cancer diagnosis and biomarker discovery from microarray gene expression data". In: *International journal of medical informatics* 74.7-8 (2005), pp. 491–503.

[56] Robin L Stears, Todd Martinsky, and Mark Schena. "Trends in microarray analysis". In: *Nature medicine* 9.1 (2003), pp. 140–145.

[57] Jian Tang and Shuigeng Zhou. "A new approach for feature selection from microarray data based on mutual information". In: *IEEE/ACM transactions on computational biology and bioinformatics* 13.6 (2016), pp. 1004–1015.

[58] *The Cancer Genome Atlas [TCGA] website*. Microarray Lung Cancer Data, preprocessed by Dr. Huang's Lab. URL: `https://www.cancer.gov/`.

[59] MTCAJ Thomas and A Thomas Joy. *Elements of information theory*. Wiley-Interscience, 2006.

[60] Jorge R Vergara and Pablo A Estévez. "A review of feature selection methods based on mutual information". In: *Neural computing and applications* 24.1 (2014), pp. 175–186.

[61] Ian Witten et al. *Data Mining: Practical Machine Learning Tools and Techniques*. Nov. 2016, p. 296.

[62] Svante Wold, Kim Esbensen, and Paul Geladi. "Principal component analysis". In: *Chemometrics and intelligent laboratory systems* 2.1-3 (1987), pp. 37–52.

[63] Hong-Jie Xing and Bao-Gang Hu. "Two-phase construction of multilayer perceptrons using information theory". In: *IEEE Transactions on Neural Networks* 20.4 (2009), pp. 715–721.

[64] Sheng Xu et al. "Convolutional neural network pruning: A survey". In: *2020 39th Chinese Control Conference (CCC)*. IEEE. 2020, pp. 7458–7463.

[65] Xiaoqin Zeng and Daniel S Yeung. "Hidden neuron pruning of multilayer perceptrons using a quantified sensitivity measure". In: *Neurocomputing* 69.7-9 (2006), pp. 825–837.

[66] Guoqiang Peter Zhang. "Neural networks for classification: a survey". In: *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)* 30.4 (2000), pp. 451–462.

[67] Zhenyu Zhao, Radhika Anand, and Mallory Wang. "Maximum relevance and minimum redundancy feature selection methods for a marketing machine learning platform". In: *2019 IEEE international conference on data science and advanced analytics (DSAA)*. IEEE. 2019, pp. 442–452.

[68] Wang Zhongxin et al. "Feature selection algorithm based on mutual information and Lasso for microarray data". In: *The Open Biotechnology Journal* 10.1 (2016).