# Gene Expression Analysis Using Machine Learning

by

Nafis Mostafa
20241055

A thesis submitted to the Department of Computer Science and Engineering
in partial fulfillment of the requirements for the degree of
B.Sc. in Computer Science

Department of Computer Science and Engineering
Brac University
January 2021

# Declaration

It is hereby declared that

1. The thesis submitted is my own original work while completing degree at Brac University.

2. The thesis does not withhold any materials previously published or written by a third party, except where this is appropriately cited through full and accurate referencing.

3. This thesis does not withhold any materials which has been accepted, or submitted, for any other degree or diploma at a university or other institution.

4. We have acknowledged all main sources of help.

**Student's Full Name & Signature:**

Nafis Mostafa
20241055

# Approval

The thesis/project titled "Gene Expression Analysis Using Machine Learning" submitted by

1. Nafis Mostafa (20241055)

Of Fall, 2020 has been accepted as satisfactory in partial fulfillment of the requirement for the degree of B.Sc. in Computer Science on January 11, 2021.

**Examining Committee:**

Supervisor:
(Member)

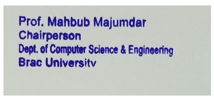
Rasif Ajwad

Rasif Ajwad
Lecturer
Computer Science Engineering
BRAC University

Program Coordinator:
(Member)

Dr. Md. Golam Rabiul Alam
Associate Professor
Computer Science Engineering
Brac University

Head of Department:
(Chair)




Mahbubul Alam Majumdar
Professor
Department of Computer Science and Engineering
Brac University

## Abstract

Cancer is a multifactorial disorder that occurs due to the complex interaction between the environment and gene. The susceptibility of a person to cancer depends on his genetic build-up. Recently, the study of genomes in discovering the interaction between disease and genes and how their interaction leads to specific phenotype, has grown exponentially. To analyze the expression of thousands of genes, one of the most important and revolutionary techniques used in genomics and systems biology is high-throughput microarray technology. To produce an accurate prognosis from such high-dimensional gene expressional data, machine learning can be an ideal choice. In this paper, we have tried to apply principal component analysis (PCA) and autoencoder on a brain cancer gene expression data retrieved from CuMiDa database and make an analysis of which technique produce better and more accurate reduced dimensional vectors and how different classical machine learning algorithms performs on these newly generated datasets. Finally, we also discussed how to improve these current techniques and how it can lead to better and sophisticated outcomes.

**Keywords:** Gene expression, PCA, Autoencoder, CuMiDa database

# Acknowledgement

Firstly, all praises goes to Almighty Allah Who has honored man and has created him in the best of forms and without Whom, my thesis would not have been completed

After that, I am very much grateful to my supervisor, Rasif Ajwad sir, and my Co-supervisor, Ismail Hossain, who have always been a constant support. They have always motivated me to think creative and help me with the understanding of concepts of Bioinformatics.

# Table of Contents

# List of Figures

# List of Tables

# Chapter 1

# Introduction

## 1.1 Human Genome Project

The importance of heredity and its principles of crossbreeding has been demonstrated by the Czech Republican scientist, Gregor Johann Mendel. He implemented its concepts for the quality improvement of crops and domestic animals. He was the one who discovered the governing laws of inheritance by studying pea plants. His work pushed the scientific community to go through intensive research to uncover the mysteries of genetics and the hidden world laid inside the chromosome until the groundbreaking discovery in 1953 when James Watson and Francis Crick discovered the DNA and its structure. Then in 1988, the Human Genome Project had commenced with three main goals: identifying all the bases in our genome's DNA, producing maps for major sections of all our chromosomes showing the location of genes, and last but not least, generating linkage maps by whose aid inherited traits, such as genetic disease, could be tracked over generation [2]. In short, generating a human genetic map, then a human genome physical map, and finally a sequence map. The word genome means that it is an organism's complete set of DNA with the inclusion of all its genes and it is comprised of more than 3 billion DNA base pairs. The Human Genome Project has revolutionized the field of biology and is now propelling the transformation of the molecular medicine industry [4], [11]. It has contributed to a more sophisticated diagnosis of diseases, early detection of some cancer gene therapy, organ cloning, and control system for drugs.

## 1.2 Basics of DNA

The basic building block of almost every living organism is Deoxyribonucleic Acid (DNA). It is mostly found inside the nucleus of a cell coiled up around a protein called histones in the form of chromosomes, while a small amount of it can also be found inside the mitochondria of a cell. A DNA is composed of 4 chemical bases, Adenine (A), Guanine (G), Cytosine (C), and Thymine (T).

The sequence of these bases along the backbones provides instructions for assembling protein and RNA molecules, Figure (1.1). The physical and functional nature of a species is determined by the fragments of DNA known as genes. Genes can be made of hundreds to millions of DNA bases in length. All human beings have similar

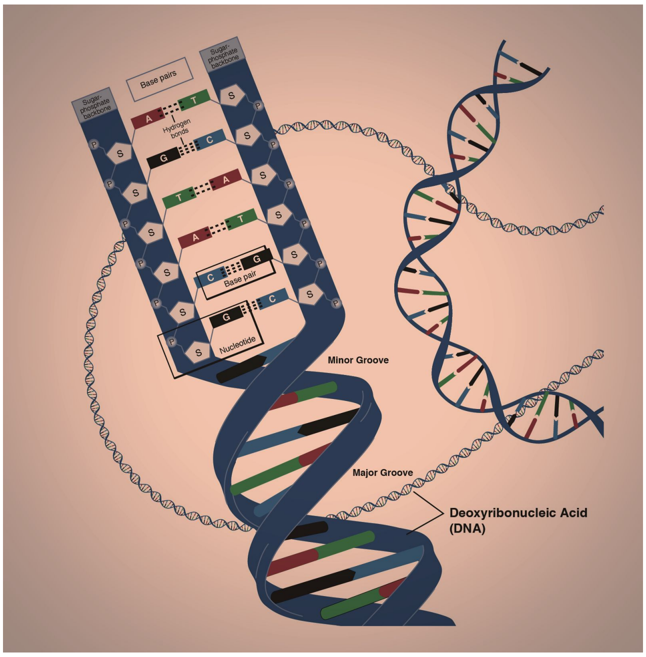



Figure 1.1: Constituent of a DNA molecule)

genes, the only little variations take place during meiosis and during the formation of the zygote which alters the genotype and thus resulting in different phenotypes.

## 1.3 Gene Expression Data

A microarray is an instrument that is used to identify the expression of thousands of genes simultaneously. The DNA chips or gene chips are tiny spots located on the microscopic slides where each probe consists of a known sequence of DNA or gene. The DNA molecules are strapped to each slide serve and act as probes. These probes are also known as the transcriptome or the messenger RNA (mRNA) transcripts [brazma2000gene]. For performing microarray analysis,mRNA molecules from both the experimental sample and a reference sample are collected. The reference sample is collected from a healthy individual, and the experimental sample is obtained from a patient suffering from diseases such as cancer. After transforming both the sample into complementary DNA, the samples are allowed to mix and 'attach to the sides of DNA probes by a binding process known as hybridization. The microarray is scanned following the hybridization to evaluate the expression of each gene printed on the slide. The data obtained from the microarrays can be used to generate profiles of gene expression, which demonstrates that in response to a particular condition or treatment, there is a simultaneous altering in the expression of various genes. A sample figure of a gene expression is shown in Figure 1.2.

With the advancement of technologies, the size of these biological data, produced by this sector, is extremely large and this needs to be transformed into useful information. Machine learning is the study of the algorithms which could learn from experience and then make predictions. Statistics and computer science are theoretical elements of machine learning, however, computational considerations are also important. Due to the complexity of the biological, machine learning plays an

| | Exp1 | Exp2 | Exp3 | Exp4 | Exp5 | Exp6 |
|---|---|---|---|---|---|---|
| Gene 1 | -1.2 | -2.1 | -3 | -1.5 | 1.8 | 2.9 |
| Gene 2 | 2.7 | 0.2 | -1.1 | 1.6 | -2.2 | -1.7 |
| Gene 3 | -2.5 | 1.5 | -0.1 | -1.1 | -1 | 0.1 |
| Gene 4 | 2.9 | 2.6 | 2.5 | -2.3 | -0.1 | -2.3 |
| Gene 5 | 0.1 | | 2.6 | 2.2 | 2.7 | -2.1 |
| Gene 6 | -2.9 | -1.9 | -2.4 | -0.1 | -1.9 | 2.9 |

Figure 1.2: Gene Expression data

important role in the analysis step.

## 1.4 Research Objectives

In this paper, our main objectives are as follows:

- To analyze gene expression data of brain cancer
- Use of different feature Extraction technique
- Validate the feature extraction techniques by running different machine learning models
- Analyze the models and evaluate which performs better in classifying different tumor subgroups

# Chapter 2

# Literature Review

The complexity of mining massive genomic data with the aid of only a visual investigation of pairwise correlations is quite an enormous challenge. Analytical tools are necessary to discover the unanticipated relationship, derive novel hypotheses and models, and make predictions [7], [14]. Some algorithms require hardcoding of domain expertise and assumptions, but unlike them, machine-learning algorithms are constructed to detect patterns automatically. Therefore, machine learning algorithms are suitable for genomics. [3], [18]. However, the representation of data and the computation of features strongly determine machine learning algorithms' performances. A preprocessing algorithm could detect cells, identify their type, and generate a list of counts for each cell type to classify a tumor as malignant or benign from a fluorescent microscopy image. A machine-learning algorithm then uses the estimated cell counts as input features to classify the tumor type. These machine learning algorithms' performance for classification strongly depends upon the relevance and quality of these features. For example, cell morphology, distances between cells, or position within an organ have not been considered in cell counts, and these inaccurate data representations result in a reduction in the precision of the classification. Deep learning solves this problem by incorporating feature computation into the machine learning algorithm itself in order to establish end-to-end models[21]. This yielding has come to light with machine learning and deep neural networks' progression, which include successive elementary operations, taking the effects of previous operations as input, and determining more complex features. The improvement in the prediction accuracy with the discovery of high complexity relevant features, such as the spatial organization of cells and cell morphology, can be achieved by deep neural networks.

The explosion of data has facilitated the creation and training of deep neural networks, algorithmic advances, and a significant rise in computational power, in particular, through the use of Graphic processing units (GPUs) [22]. Deep neural networks have contributed to numerous breakthroughs in speech recognition, machine translation, and computer vision[13], [16], [19]. Since the demonstration of the applicability of deep neural networks to DNA sequence data [17], [20] in 2015 seminal studies, the number of articles related to the application of deep neural networks to genomics has grown exponentially.

In some of the recent studies, [6], [12], [15], the proposition of many feature selection methods has been made. The authors in [6] have tried to differentiate samples using the relief-f filtering feature selection method and use an SVM classifier. To enhance the classification accuracy and feature selection stability, authors have used an ensemble of feature selection methods [10]. Some early cancer prognosis attempts have been made to construct models using either the genes expression data [9], clinical tumor and patient data [8], or some cellular features of tissue slides. These studies have made a comparison to demonstrate the similar performance acquirement of Neural Network's success output to Cox-PH and Kaplan Meier methods. The deep neural network model, DeepSurv, built by Katzman et al., has outperformed the CoxPH model. The model uses the patient's clinical data as input and incorporates regularization, dropout, and learning rate decay, to optimize for different dataset [23]. Huang et al. had collected five omics data and then performed feature extraction from these data before creating a deep learning model to predict the survival of patients with breast cancer[26].In the multi-omics NN (or SALMON) model, the authors have performed feature extraction using a local maximum Quasi-Clique Merger (lmQCM) spectral clustering algorithm from mRNA and miRNA data [26]. Another study had devised a support vector machine (SVM) classifier to identify biomarker genes related to prostate cancer progression using next-generation sequencing data. They were able to distinguish successive prostate cancer stages with relative high performances [24].

# Chapter 3

# Methodology

## 3.1 Model

### 3.1.1 Principal Component Analysis (PCA)

Principal component analysis (PCA) is an orthogonal linear transformation technique which is well-known procedure for feature extraction and dimensionality reduction. The goal of PCA is to map high-dimensional data into a lower-dimensional space such that maximum variance from the original data can be preserved, while minimising the total squared error. This also causes there is a reduction of space and time complexities as well. The method is mostly beneficial for differentiating signals from various sources. If the number of independent components is aware ahead of time then the technique is easier to carry out as with standard clustering methods. The process of working with principal components is theoretically rather straightforward. Firstly, for the complete dataset, the $xd$ dimensional mean vector $\mu$and $x \times x$ covariance matrix is computationally calculated. Consequently, in terms of decreasing eigenvalue, the eigenvectors and eigenvalues are computed and organized accordingly [5].Thereafter, the largest $k$such eigenvectors are selected by examining the array of eigenvectors. The rest of the dimensions are noise. A $k \times k$ matrix A is formed whose columns consist of k eigenvectors. The data is pre-processed according to $x = A^t(x - \mu)$.

### 3.1.2 Autoencoders

An autoencoder is a neural network which, other than being helpful for a lot of tasks, is also an entry point to learn more complex concepts in machine learning. In deep autoencoding, autoencoder focuses on deriving complex transformations from simple ones and performs autonomous learning to identify hierarchies of features and thus fragmenting the data to generate features [1]. In this procedure, an addition of one more autoencoder layer means an addition of inputs with abstract representations. The use of an autoencoder can overcome the limitations of neural networks with randomly initialized weight values, meaning that the input features are independent of each other. However, if there exists a correlation between input features, the association can be learned and simultaneously exploited when the input is passed through the neural network's bottleneck.

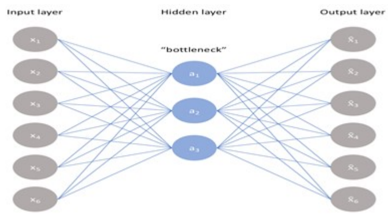



Figure 3.1: Bottleneck Linear Autoencoder

As seen in the above figure 3.1, an unlabeled dataset is taken and is made to output $\hat{x}$, a reconstruction of the original input $x$, given that the unlabeled dataset is framed as a supervised learning problem. By shrinking the reconstruction error, $L(x, \hat{x})$, the network can be trained and thus measure the differences between the original input and the subsequent reconstruction. The bottleneck being the key attribute of the network design, constrains the amount of information that can flow through the whole network thus constraining a learned compression of the input data, otherwise, the input values could easily traverse through the network by simply memorizing the input values as inferred in the figure 3.2 below.

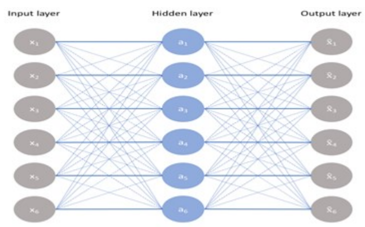



Figure 3.2: Linear Autoencoder

The autoencoder could produce a result similar to the dimensionality reduction as seen in PCA if the constructed network was linear, i.e., at each layer, nonlinear activation functions were omitted. The two balancing properties of an ideal autoencoder are the model's sensitivity to the inputs during the actual rebuilding of the model and its insensitivity to the model such that the model does not overfit or memorize the training data. Such a balance forces the model to retain only the differences in the data used to rebuild the input, thus disregarding any repetitions within the input. Almost in all cases, a loss function is built where one term stimulates the model to be reactive to the input, that is, reconstruction loss $L(x, \hat{x})$ and the consequent term dissuades memorization/overfitting, which in this case is an added regularizer.

$$L(x, \hat{x}) + regularizer$$

To adjust the trade-off among the two objectives, a scaling parameter is also added in front of the regularization term.

### 3.1.3 Supervised Models

The main goal of supervised machine learning models is to make a prediction for a target output by building a model which takes features as input. In order to construct a model, the model is first needs to be trained using the features vectors.

The training of a machine learning model infers to learning its parameters, and this requires minimizing of the loss function on training data with the goal of achieving accurate prediction on unseen data.

## 3.2 Proposed methodology

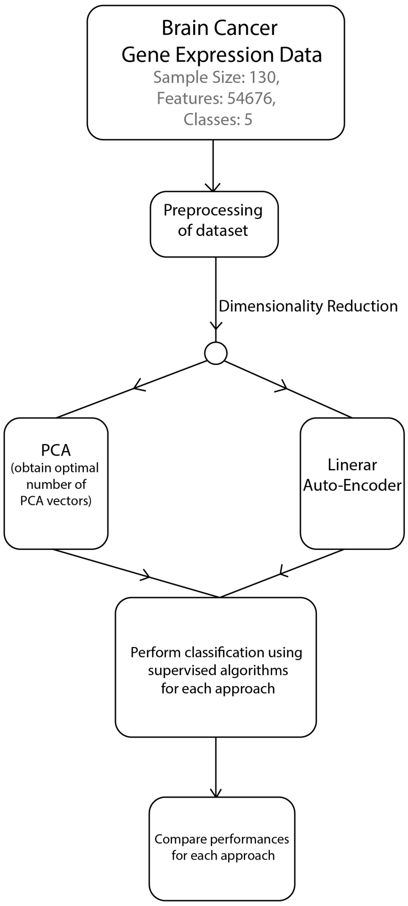



Figure 3.3: Workflow

### 3.2.1 Dataset

In this paper, we have the brain cancer gene expression dataset retrieved from Curated Microarray Database [25]. The microarray dataset found in CuMida have been extensively curated from the 30000 studies from Gene Expression Omnibus (GEO), dedicated for machine learning. The dataset have already preprocess and free from null values, reading from unwanted probe, and normalized. To ensure the validity of the dataset, 3-fold cross validation have been performed on it. The dataset is retrieved from platform GSE50161 and contains the gene expression levels of 54676 genes (columns) from 130 samples (rows). There are in total 5 classes, where 4 of them are different types of brain cancer (ependymoma, glioblastoma, medulloblastoma, pilocytic astrocytoma) and the last one is normal healthy human tissue.

### 3.2.2 Preprocessing

First of all, we have categorize the output classes where normal = 0, ependymoma = 1, glioblastoma = 2,'medulloblastoma = 3, pilocytic astrocytoma = 4 have been labelled. Then we have split the dataset into a 80:20 ratio where 80% are split into training set and rest as testing set. Then we have used "Standardscaler" of sklearn library.

```
# Feature Scaling
from sklearn.preprocessing import StandardScaler
sc = StandardScaler()
X_train_scl = sc.fit_transform(X_train)
X_test_scl = sc.transform(X_test)
```

Figure 3.4: Feature Scaling)

### 3.2.3 Dimensionality Reduction

After the preprocessing, we have first applied Principal component analysis (PCA) on the dataset. We have checked for which value of k, the variance is greater than 95%. Here variance means 95% of the original components can represented by the k number of vectors. Then using that value of k, we reduced the number of features of the original dataset. As for the auto-encoder, we have tried a lot of combination for instance, decreasing the number of neurons by a factor of 2 in the encoded section of the network. But the main problem was that the current configuration of our hardware wasn't powerful enough to support our approach. First, we ran our program on a desktop with a configuration of Core i5 9th Generation 2.4GHz processor, 16GB RAM, and NVIDEA GTX GeForce 1650 8GB graphics card. The entire 16GB RAM wasn't being used since some of it were being used to run OS and other background software. So, we decided to run to our program on Kaggle platform which allow us to have a dedicated 16 GB RAM to run the program. But unfortunately, our algorithm failed to run in this configuration as well. So, we have decided to change the neural architecture and the final version of it is shown below in figure 3.3.

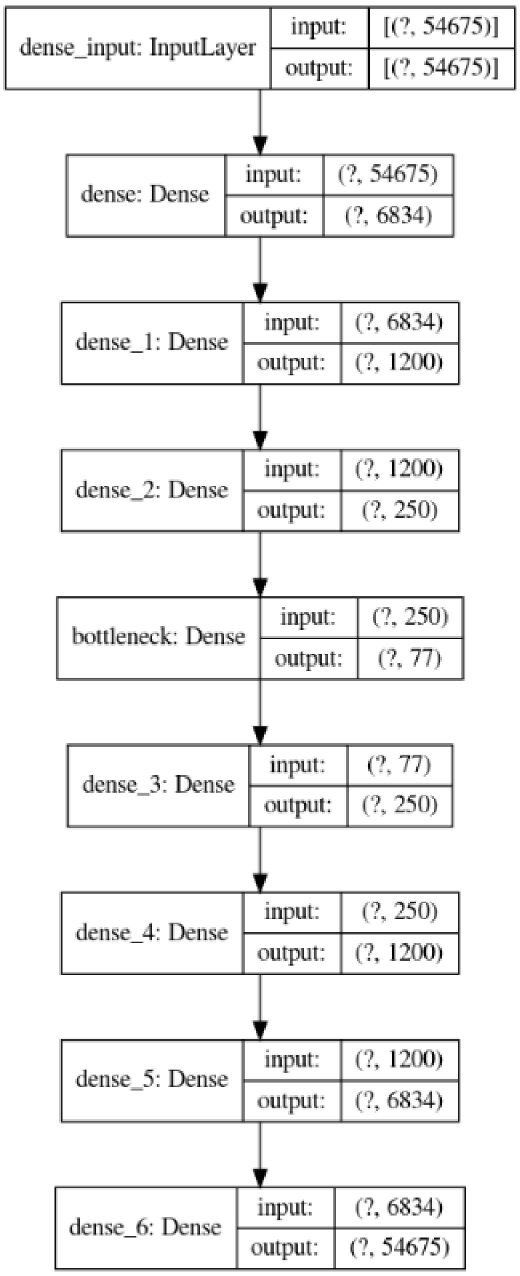



Figure 3.5: Proposed Autoencoder Architecture

In our linear autoencoder architecture, we have used 'relu' activation function for all the layers except the final layer in which we have incorporated softmax. Moreover, we have included 'mean squared logarithmic error loss function and 'adamax' optimizer in the final output layer. We have trained our model for 20 epochs with a batch size of 6.

### 3.2.4 Classification

For classification between different classes of brain cancer type, we have applied six different supervised algorithms including, Gaussian Naïve Bayes, Support Vector Machine (SVM), Decision Tree, Bagging, Random Forest and Ada Boost.

# Chapter 4

# Result

At first we have tried to determine for which value of k, the PCA algorithm can represent 97% of the feature vector. Figure 4.1 shows that, for k = 79, we get our desired result. After determination of our desired k value, we have used it to reduce

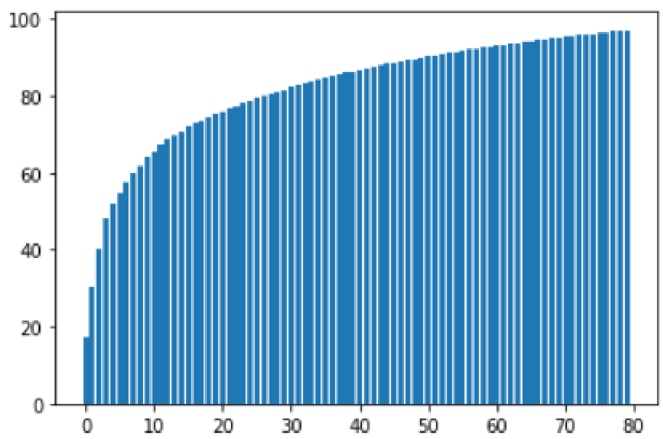



Figure 4.1: Proposed Autoencoder Architecture

our features from 54575 to 79 components for both the training and the test set. Then using the reduced dataset, we train our supervised machine learning models.

Table 4.1: Accuracy achieved while using PCA-generated dataset

| Algorithm | Accuracy |
|---|---|
| Random Forest | 92.31% |
| DecisionTree | 84.61% |
| Naive Bayes | 92.31% |
| SVMK | 92.31% |
| Bagging | 92.31% |
| Ada Boosting | 88.46% |

For the autoencoder, we have trained the dataset for 20 epochs with a batch size of 6. For running each epoch, it took approximately 100 seconds on an average and in total, the network took around 34 minute to get trained completely with a final minimal loss of around 2.33%. The loss of both the training and testing data is shown below.

Table 4.2: Accuracy achieved while using Autoencoder-generated dataset

| Algorithm | Accuracy |
|---|---|
| Random Forest | 80.77% |
| DecisionTree | 73.08% |
| Naive Bayes | 80.77% |
| SVMK | 80.77% |
| Bagging | 76.92% |
| Ada Boosting | 61.54% |

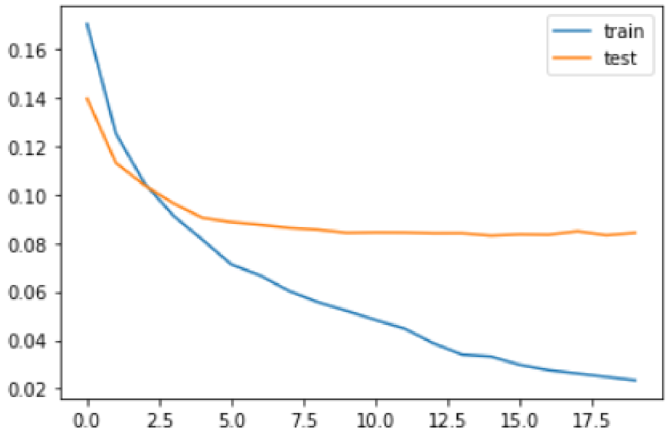



Figure 4.2: Proposed Autoencoder Architecture

The encoder section bottlenecks with 77 neurons meaning that this layer will act as the output for the feature vectors. After this, using the encoder part from the trained architecture, we have generated new datasets for training and testing with reduced features. Using these, the above mentioned classifiers are trained to obtain the following results on test datasets:

# Chapter 5

# Discussion

First of all, the importance of feature reduction is important for preparing the data to be used for training classifiers. When using PCA, we have seen that it can compact more 97% of the features within 79 feature vectors and the training time took approximately 15 seconds whereas in training the linear autoencoder, it took almost 34 minutes. The difference in training is massive. Moreover, on comparing the predictions made by the classifiers from both the dataset, we have seen that models trained on PCA-generated dataset outperforms the models trained on autoencoder-generated dataset. The accuracy scores of Gaussian Naïve Bayes, Decision Tree, Support Vector Machine, Bagging, Random Forest, and Ada Boost on PCA-generated datasets are 92.31%, 84.62%, 92.31%, 92.31%, 92.31% and 88.46% respectively whereas for autoencoder-generated dataset the scores are 80.77%, 73.08%, 76.92%, 73.08%, 80.77% 61.54%. So the maximum score on autoencoder-generated dataset is 80.77% and on PCA-generated dataset it is 92.31%. On a first look, it may look like that PCA has better performances in reducing dimensions than autoencoder. But the main problem with the performances of autoencoder is the limitation of hardware performances. During its training, the loss on the training set has diminished to around 2.33% where in the test set its around 14%. On observing the curve, we can make an assumption that the model is probably overfitted which can be due to limited number of training samples. But due to hardware limitation, the number of neurons in each layer had to be reduced, for instance in the second layer the number of neurons has decreased from 54675 to 6834. This has caused a loss of transferring of information onto the second layer. Furthermore, this has contributed to the reduced number of layers till the bottleneck of the encoder section, so thus the weight matrices of both the encoder and decoder has been wrongly constructed. Another important observation is that the loss function was over 200% when the autoencoder model was being trained on standardized dataset. So it can be concluded that the feature extraction done on an unscaled dataset using autoencoder on gene expression, generates better feature vectors, although the dataset, which was used from CuMida database [25], was already normalized using log transformation. So we can conclude that reducing the differences of neurons between two layers in the encoder and decoder section would have probably enhance quality of feature vectors and thus leading to better performances of classification models.

# Chapter 6

# Conclusion and Future Prospects

Overall, it is quite clear that reducing the number of features of dataset leads to better performances of models on datasets and PCA is a better choice. This is very important since our dataset contains expression levels of only 54676 genes whereas on a larger scale there can be millions of genes. It is also clear that the autoencoder can also play a vital role on a large-scaled data given that there is sufficient RAM, CPU and GPU requirements. But the main problem with the gene expression data is that there are less number of samples which is why the neural network do not get to train properly. This is where General Advarsarial Network (GAN) come into play. Addition of synthetic data to the original set can really boost up the performances of the neural network and in this paper we have already proved that the models make better predictions if the gene expression data are not scaled. We are planning to continue our works in this field to verify our hypothesis as in the future, the genomic based prediction will play a major role in understanding how the effects of diseases vary from person to person, since these cause an alteration of gene expression levels.

# Bibliography


[1] M. A. Kramer, “Nonlinear principal component analysis using autoassociative neural networks,” *AIChE journal*, vol. 37, no. 2, pp. 233–243, 1991.

[2] F. S. Collins and L. Fink, “The human genome project,” *Alcohol Health and Research World*, vol. 19, no. 3, p. 190, 1995.

[3] R. Durbin, S. R. Eddy, A. Krogh, and G. Mitchison, *Biological sequence analysis: probabilistic models of proteins and nucleic acids*. Cambridge university press, 1998.

[4] F. S. Collins and V. A. McKusick, “Implications of the human genome project for medical science,” *Jama*, vol. 285, no. 5, pp. 540–544, 2001.

[5] I. T. Jolliffe, “Springer series in statistics,” *Principal component analysis*, vol. 29, 2002.

[6] Y. Wang and F. Makedon, “Application of relief-f feature filtering algorithm to selecting informative genes for cancer classification using microarray data,” in *Proceedings. 2004 IEEE Computational Systems Bioinformatics Conference, 2004. CSB 2004.*, IEEE, 2004, pp. 497–498.

[7] C. M. Bishop, *Pattern recognition and machine learning*. springer, 2006.

[8] R. Joshi and C. Reeves, “Beyond the cox model: Artificial neural networks for survival analysis part ii,” in *Proceedings of the eighteenth international conference on systems engineering*, 2006, pp. 179–184.

[9] L. P. Petalidis, A. Oulas, M. Backlund, M. T. Wayland, L. Liu, K. Plant, L. Happerfield, T. C. Freeman, P. Poirazi, and V. P. Collins, “Improved grading and survival prediction of human astrocytic brain tumors by artificial neural network analysis of gene expression microarray data,” *Molecular cancer therapeutics*, vol. 7, no. 5, pp. 1013–1024, 2008.

[10] T. Abeel, T. Helleputte, Y. Van de Peer, P. Dupont, and Y. Saeys, “Robust biomarker identification for cancer diagnosis with ensemble feature selection methods,” *Bioinformatics*, vol. 26, no. 3, pp. 392–398, 2010.

[11] E. D. Green and M. S. Guyer, “Charting a course for genomic medicine from base pairs to bedside,” *Nature*, vol. 470, no. 7333, pp. 204–213, 2011.

[12] A. Sharma, S. Imoto, and S. Miyano, “A top-r feature selection algorithm for microarray gene expression data,” *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, vol. 9, no. 3, pp. 754–764, 2011.

[13] P. Bartlett, F. Pereira, C. Burges, L. Bottou, and K. Weinberger, *Advances in neural information processing systems 25 (nips 2012): 26th annual conference on neural information processing systems 2012*, 2012.

[14] K. P. Murphy, *Machine learning: a probabilistic perspective.* MIT press, 2012.

[15] R. Fakoor, F. Ladhak, A. Nazi, and M. Huber, "Using deep learning to enhance cancer diagnosis and classification," in *Proceedings of the international conference on machine learning*, ACM New York, USA, vol. 28, 2013.

[16] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2014, pp. 580–587.

[17] B. Alipanahi, A. Delong, M. T. Weirauch, and B. J. Frey, "Predicting the sequence specificities of dna-and rna-binding proteins by deep learning," *Nature biotechnology*, vol. 33, no. 8, pp. 831–838, 2015.

[18] M. W. Libbrecht and W. S. Noble, "Machine learning applications in genetics and genomics," *Nature Reviews Genetics*, vol. 16, no. 6, pp. 321–332, 2015.

[19] J. Long, E. Shelhamer, and T. Darrell, "Fully convolutional networks for semantic segmentation," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 3431–3440.

[20] J. Zhou and O. G. Troyanskaya, "Predicting effects of noncoding variants with deep learning–based sequence model," *Nature methods*, vol. 12, no. 10, pp. 931–934, 2015.

[21] I. Goodfellow, Y. Bengio, and A. Courville, "Deep learning book," *MIT Press*, vol. 521, no. 7553, p. 800, 2016.

[22] S. Shi, Q. Wang, P. Xu, and X. Chu, "Benchmarking state-of-the-art deep learning software tools," in *2016 7th International Conference on Cloud Computing and Big Data (CCBD)*, IEEE, 2016, pp. 99–104.

[23] J. L. Katzman, U. Shaham, A. Cloninger, J. Bates, T. Jiang, and Y. Kluger, "Deepsurv: Personalized treatment recommender system using a cox proportional hazards deep neural network," *BMC medical research methodology*, vol. 18, no. 1, p. 24, 2018.

[24] A. Alkhateeb, I. Rezaeian, S. Singireddy, D. Cavallo-Medved, L. A. Porter, and L. Rueda, "Transcriptomics signature from next-generation sequencing data reveals new transcriptomic biomarkers related to prostate cancer," *Cancer informatics*, vol. 18, p. 1 176 935 119 835 522, 2019.

[25] B. C. Feltes, E. B. Chandelier, B. I. Grisci, and M. Dorn, "Cumida: An extensively curated microarray database for benchmarking and testing of machine learning approaches in cancer research," *Journal of Computational Biology*, vol. 26, no. 4, pp. 376–386, 2019.

[26] Z. Huang, X. Zhan, S. Xiang, T. S. Johnson, B. Helm, C. Y. Yu, J. Zhang, P. Salama, M. Rizkalla, Z. Han, *et al.*, "Salmon: Survival analysis learning with multi-omics neural networks on breast cancer," *Frontiers in genetics*, vol. 10, p. 166, 2019.