Islamic University of Technology (IUT)

Blood Cancer Prediction using Leukemia Microarray Gene Data and Deep Learning

Authors

MD Mehdad Hossain, 180041232

MD Abul Kalam Siddiquee, 180041216

Muhammad Yeasin Hossain, 180041125

Supervisor

Tareque Mohmud Chowdhury

Assistant Professor

Dept. of CSE, IUT

Co-Supervisor

Tasnim Ahmed

Lecturer

Dept. of CSE, IUT

A thesis submitted in partial fulfillment of the requirements

for the degree of B. Sc. Engineering in Computer Science and Engineering

Academic Year: 2021-2022

Department of Computer Science and Engineering (CSE)

Islamic University of Technology (IUT)

A Subsidiary Organ of the Organization of Islamic Cooperation (OIC)

Dhaka, Bangladesh

June 6, 2023

Declaration of Authorship

This report is to certify that the work submitted in this thesis is the outcome of the analysis and experiments carried out under the supervision of Tareque Mohmud Chowdhury, Assistant Professor, Department of Computer Science and Engineering (CSE), Islamic University of Technology (IUT), Gazipur, Bangladesh. It is further declared that no portion of this thesis or any other portion has ever been presented anywhere for a degree or diploma. A list of references is provided, and information taken from both published and unpublished works of other party is recognized in the document.

Authors:

MD Mehdad Hossain
Student ID - 180041232

MD Abul Kalam Siddiquee
Student ID - 180041216

Muhammad Yeasin Hossain
Student ID - 180041125

Approved By:

Supervisor:

Tareque Mohmud Chowdhury
Assistant Professor
Department of Computer Science and Engineering (CSE)
Islamic University of Technology (IUT), OIC

Co-Supervisor:

Tasnim Ahmed
Lecturer
Department of Computer Science and Engineering (CSE)
Islamic University of Technology (IUT), OIC

Acknowledgement

We want to thank Tareque Mohmud Chowdhury, Assistant Professor, Department of Computer Science and Engineering, Islamic University of Technology, for serving as our mentor and advisor. His motivation, suggestions, and ideas have been extremely helpful for this project. This research would not have been accomplished without his assistance and right direction. His valuable judgement, effort, and time were supplied throughout the work, from the first phase of thesis themes introduction to the subject selection, proposing algorithm, and modification, to the project implementation and finalization, which has assisted us to properly complete our work. He has our sincere gratitude.

We also express our gratitude to Tasnim Ahmed, Lecturer, Department of Computer Science and Engineering, Islamic University of Technology, for his valuable inspection and suggestions on our proposal for Deep Learning models for the microarray gene dataset. His extraordinary knowledge, tolerance, and enthusiasm have been crucial in guiding our research and encouraging our development as aspiring researchers. His insightful criticism and guidance have greatly improved the caliber and scope of our work. We are very appreciative of his never-ending support and faith in our competence. We are grateful to have had the opportunity to work under their direction because of the deep contributions he made, which were essential to the achievement of this thesis.

Abstract

The diagnosis of blood cancer with the use of any leukemia microarray gene sequence data and a machine learning approach is one of the most important fields of medical research. More advancements are needed to obtain the requisite accuracy and efficiency notwithstanding research efforts.

Our work's major goal is to present a method that, using microarray gene data, can accurately predict blood cancer. By increasing the classification accuracy for automated analysis of microarray data analysis, our research seeks to suggest a deep learning model to identify and categorize various types of leukemia.

We will use the Leukemia_GSE28497 dataset for training our model which contains 281 samples consisting of 22,285 genes (features) of 7 target classes. We preprocess the dataset by deleting null items before training our models. For the prediction of the blood cancer classes, we investigate three classification algorithms: logistic regression, single-layer neural networks, and TabNet. We use a variety of metrics, such as model accuracy, model loss, confusion matrix, train value accuracy, train value loss, and ROC curve, to measure the performance of our models. The outcomes of our studies analyze the effectiveness of deep learning models for classifying different forms of blood cancer from microarray gene data.

Contents

1 Introduction

1.1 Overview

Leukemia is one type of blood cancer that is distinguished by the prompt increase of abnormal blood cells. This uncontrolled growth happens in bone marrow, which is where the majority of the blood in a body is made. Leukemia cells are often young or still under development white blood cells. Leukemia cells divide quickly, causing the disease to spread quickly. If a person has acute leukemia, he will start to feel sick within several weeks of the leukemia cells forming. It is a dangerous illness that requires prompt treatment. It is also the most prominent type of malignancy in children. Acute lymphocytic leukemia (ALL) is the most common kind of leukemia in kids, teenagers, and young adults up to age 39. ALL can have an effect on adults of any age. Acute myelogenous leukemia (AML) is the most common kind of acute leukemia in adults. Over 65s are more prone to it than younger people. Young people can also be affected by AML.

Bioinformatics is the area of computer science that is most pertinent to our study. Bioinformatics is the study of the methods used to collect biological data, transform it into useful formats, and then analyze the collected data. The diagnosis of cancer is a major area of bioinformatics research. One of the cutting-edge technologies used to assess the amount of gene expression in a large number of genes is the DNA microarray. This method can establish whether a person's DNA contains a gene for mutation or not. Numerous forms of leukemia can be analyzed and predicted using microarray technology. Hospitals also produce a significant amount of DNA expression data thanks to microarray technologies.

1.2 What is Microarray Gene Expression

The fundamental structural and functional component of heredity is a gene. DNA is a component of genes. Some genes function as blueprints for producing proteins. Many genes, however, do not code for proteins. Several hundreds of DNA

nucleotide to over 2 million bases can make up a human gene. Humans have 20,000 to 25,000 genes, according to the Human Genome Project, an international research endeavor to comprehend the human genome's sequence and catalog its genes.

Gene expression is the process through which a gene's information is converted into a function. The main cause of this is transcription of RNA molecules that code for proteins or non-coding RNA molecules that have other functions. Gene expression is analogous to a "on/off switch" that regulates both the location and timing of protein and RNA synthesis as well as a "volume control" that regulates the relative quantities of each. Environment-related variables and cell type have a big impact on how genes are expressed. The RNA and proteins that regulate the expression of other genes are made by several genes.

An experimental method for concurrently detecting the expression levels of thousands of genes is called a microarray. The tens of thousands of tiny dots on the microscope slide each stand for a recognized gene or DNA sequence. We call them DNA microarrays. DNA or gene chips are common names for these slides. The messenger RNA (mRNA) transcriptome, also known as the collection of messenger RNA (mRNA) transcripts expressed by a set of genes, is detected by the DNA molecules linked to each slide as probes.

1.3 Microarray Gene Expression Data for Leukemia Detection and Classification

The need for early detection and appropriate treatment for blood cancer has increased during the past ten years. The diagnosing procedure is expensive, time-consuming, and involves a number of tests and medical professionals. Therefore, an automatic diagnosis system is crucial for making an accurate forecast. Today, one of the most significant areas of medical study is the diagnosis of blood cancer combining leukemia microarray gene data and a machine learning approach.

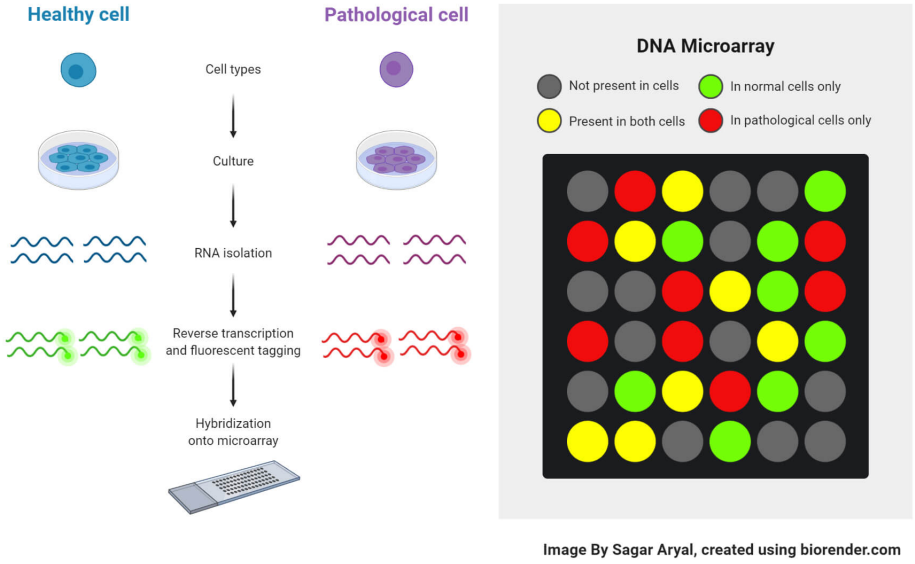

Figure 1: Steps involved in cDNA-based microarray

Despite research efforts, more improvements are required to achieve the desired accuracy and efficiency.

Up until the advent of Next Generation Sequencing (NGS) methods, microarray technology was the most frequently utilized gene quantification technology in the last 20 years. Due to the inexpensive cost of this technology in comparison to RNA-seq, Microarray is still in use today. The first technology to measure all genes' levels of expression concurrently was the microarray. Numerous Microarray systems or manufacturers can also be considered.

Aggressive cancer detection techniques, including bone marrow biopsy, which is employed in the precise diagnosis of acute myeloid leukemia, can take the role of microarray. The invasive, uncomfortable, and potentially dangerous procedure known as a bone marrow biopsy can result in bleeding and infections. [12] Because they provide very accurate diagnostic techniques, microarrays improve clinical diagnosis. It can serve as a gold standard for diagnosis, advanced therapy research, and cell biology comprehension, particularly in oncology studies. [16] Automated gene data classification based on machine learning is crucial for diagnosing any

genetic disorders and diseases. The need for a good classifier that can handle massive amounts of data is critical given the quantity of the data. One of the cutting-edge machine learning methods to address these issues is deep learning. Because there are more hidden layers, it can process large amounts of data with ease.

1.4 Motivation and Problem Statement

Blood cancer cases have increased over the past ten years, and early detection is essential for successful treatment to start. The diagnosing process is pricy, drawn out, and requires a variety of tests and medical experts. Consequently, a system for autonomous diagnosis is essential for producing precise forecasts. One of the most important fields of medical research nowadays is the diagnosis of blood cancer using a machine learning approach using leukemia microarray gene data. More advancements are needed to obtain the requisite accuracy and efficiency notwithstanding research efforts.

In the modern medical world, early leukemia prediction is a difficult task that can be accomplished by implementing computer-aided automated illness diagnosis systems. To create a smart diagnosis method, many machine learning algorithms had been used for medical datasets. Huge amounts of data are being generated from the medical sectors as a result of the digital revolution and improvements in information technologies. Machine learning algorithms are highly adapted for the processing of these enormous amounts of data, and numerous techniques have also been used for disease diagnosis. [8]

The main objective of our work is to put forth a better deep learning model that can more accurately predict blood cancer utilizing larger microarray gene sequence datasets. Our research aims to develop a deep learning model to diagnose and classify different types of leukemia by improving the classification accuracy for automated analysis of microarray data analysis.

2 Literature Review

Due to the significance of the healthcare industry, a number of research studies on leukemia cancer prediction employing machine learning, deep learning, and image processing can be found in the literature. A class of machine learning algorithms known as "deep learning" is capable of accurately analyzing data with many features and samples as well as identifying patterns with high complexity in huge datasets.

2.1 Machine Learning Approaches for Leukemia Classification

The 22,283 gene leukemia microarray gene dataset is used in the research of Rupapara et al. [13] Problems with imbalanced and high-dimensional datasets are solved using ADASYN resampling and Chi-squared (Chi2) features selection methods. To balance the dataset for each target class, ADASYN creates synthetic data, then Chi2 chooses the top 22,283 characteristics to train learning models on. A hybrid logistics vector trees classifier (LVTrees) that combines logistic regression, support vector classifiers and additional tree classifiers is proposed for classification. For assessing the significance of the suggested methodology, extensive experiments on the datasets and performance comparisons with cutting-edge methods have been done. With significant 100% accuracy, LVTrees exceed all other models using ADASYN and Chi2 methods. A statistical significance T-test is also run to demonstrate the effectiveness of the suggested strategy. Results from k-fold cross-validation show that the suggested model is superior. On the basis of microarray gene data, the effectiveness of well-known machine learning methods is examined. These algorithms include RF, logistic regression (LR), support vector classifier (SVC), KNN, Naive Bayes (NB), extra tree classifier (ETC), DT, and Adaboost classifier (ADA). LVTrees, a hybrid model that makes use of RL, SVC, and ETC through majority voting, is suggested. Chi2 is used to choose the ideal collection of characteristics for classification while the influence of ADASYN is examined for

data balance. Extensive tests to determine the effectiveness of the suggested strategy. Modern techniques are contrasted with the suggested strategy. The proposed approach's validity is examined using the statistical significance test. Utilizing k-fold cross-validation, results are further validated.

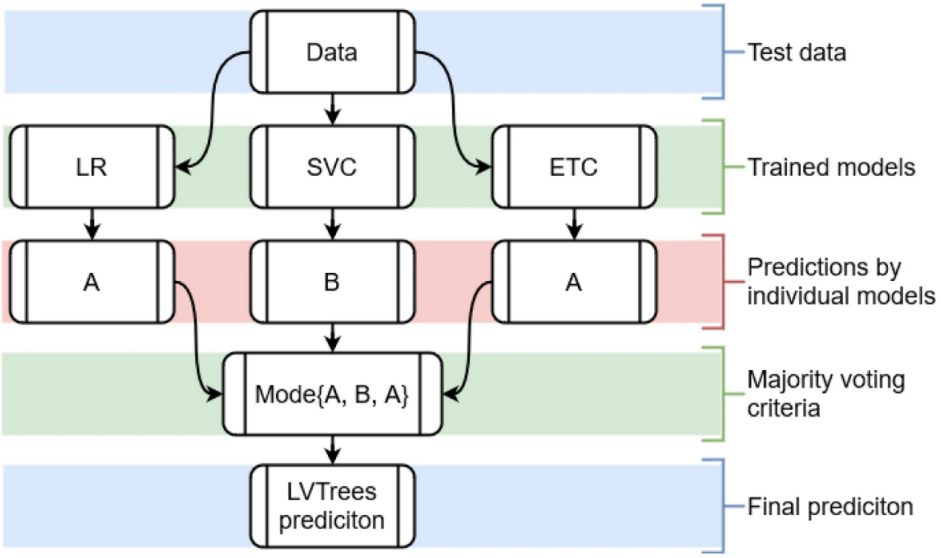

Figure 2: The Methodology Applied for the Rapapura et al. Study

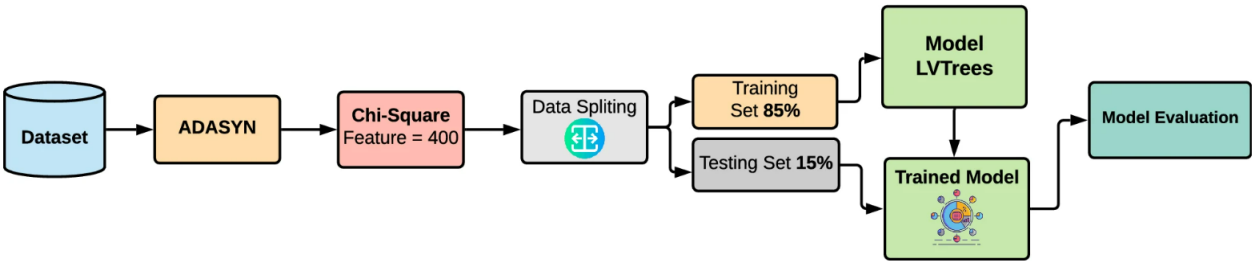

Figure 3: The Architecture of the Hybrid LVTrees Model

The study by Abdullah et al. [2] provides a method for predicting leukemia subtypes from microarray gene expression data using an ensemble classifier called EODClassifier, which has just been published. A classification accuracy of over 96% is established over numerous cross-validation studies, demonstrating consistency and robustness.

In the work of Castillo et al. [3], several Microarray and RNA-seq platforms have

been integrated, and this has allowed for the construction of a multiclass study using samples from the main four kinds of leukemia to measure gene expression. Then, a novel parameter known as coverage is introduced here in order to identify a set of differentially expressed genes with the best ability to distinguish between various forms of leukemia. This measure enables evaluation of the variety of disorders that a certain generation can identify. It has been assessed along with other well-known criteria using an ANOVA statistical test, which confirmed its filtering ability when the identified genes are put through a multiclass level machine learning process.

42 highly relevant expressed genes were chosen as a result of the statistical test's statistical evaluation of the criteria for gene extraction. These genes were reordered and evaluated using four different classification strategies using the minimum-Redundancy Maximum-Relevance (mRMR) feature selection algorithm. Excluding all other factors and considering only the top ten genes in the ranking led to exceptional results. After consulting specialized literature, it was discovered that almost all of the genes in this last subset were involved in biological processes connected to leukemia. These findings highlight the importance of taking into account a new criterion that makes it easier to identify highly valid expressed genes for simultaneously differentiating between numerous forms of leukemia.

In the Kilicarslan et al. study [7], a hybrid approach is proposed that combines support vector machines (SVM) and convolutional neural networks (CNN) for classification with Relief and stacking autoencoders for dimension reduction. Three microarray datasets for the ovary, leukemia, and central nervous system (CNS) were utilized in the study. There are 60 samples, 7129 genes, and 2 classes in the CNS dataset, 253 samples, 15,154 genes, and 2 classes in the ovarian dataset. There are 72 samples, 7129 genes, and 2 classes in the leukemia dataset. SVM had the best classification accuracy without dimension reduction when applied to the three microarray datasets, with values of 96.14% for the ovarian dataset, 94.83% for the leukemia dataset, and 65% for the CNS dataset. In comparison to

existing methods, the suggested hybrid ReliefF + CNN method fared well. For the ovarian, leukemia, and CNS datasets, it provided classification accuracy of 98.6%, 99.86%, and 83.95%, respectively. Results indicate that dimension reduction techniques increased the classification precision of SVM and CNN techniques.

Human acute leukemia is used as a test case for a general approach to cancer categorization based on gene expression monitoring using DNA microarrays in the work of El-Nasser et al. [1] When these classes were known in advance, a class discovery technique automatically determined the distinctions between acute lymphoblastic leukemia (ALL) and acute myeloid leukemia (AML). The two primary goals of this research are to introduce the SMIG (Select Most Informative Genes) Algorithm and to create and implement an enhanced classification algorithm (ECA) system to improve the classification of leukemia cancer utilizing the SMIG module and ranking process. The proposed strategy and trials revealed that, when preprocessing and classification are completed using the suggested ECA system, an accuracy of 98% may be attained in 0.1 seconds, which is faster than previous methods in previously published studies.

In the study of Xiao et al. [15], we present a novel approach that integrates a variety of machine learning models into an ensemble approach using deep learning. Using differential gene expression studies, we provide valuable gene information to five different categorization models. After that, a deep learning technique is used to combine the results from the five classifiers. Three publicly available RNA-seq data sets representing three different cancer types—stomach adenocarcinoma, lung adenocarcinoma, and breast invasive carcinoma—were used to test the proposed deep learning-based multi-model ensemble technique. The test findings show that, in comparison to employing a single classifier or the majority voting algorithm, it improves the accuracy of cancer prediction for all the evaluated RNA-seq data sets.

2.2 Deep Learning Approaches for Leukemia Classification

To comprehend how deep neural networks train, Mallick et al. [10] have provided a categorization algorithm (DNN). The network is over-parameterized and the assumptions are that the inputs do not degrade. Additionally, the quantity of hidden neurons is sufficient. DNN was utilized by the authors of this study to categorize the gene expression data. The expressions for bine marrow of 72 patients with leukemia are included in the dataset used for the study. For acute lymphocyte (ALL) and acute myelocytic (AML) samples classification, a five-layer DNN classifier is created. 80% of the data is used to train the network, while the remaining 20% is used for validation. In comparison to existing classifiers, the proposed DNN classifier is producing results that are satisfactory. With 98.2% accuracy, 96.59% sensitivity, and 97.9% specificity, two kinds of leukemia are identified. Future generations of genetic and virology researchers may benefit from the various sorts of computer-based analyses of genes.

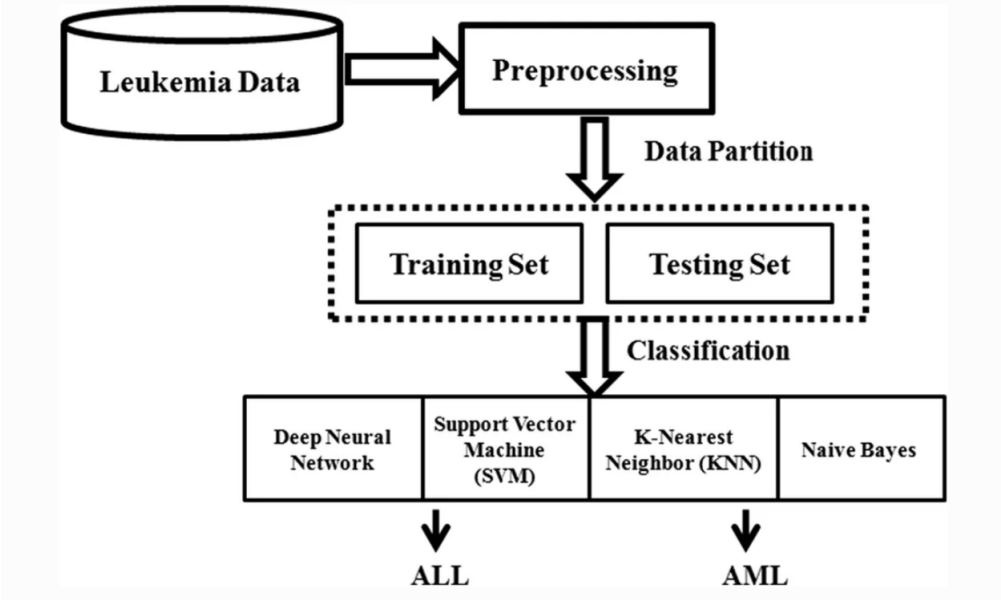

Figure 4: The Architecture of the Mallick et al. Study

The Gene Expression Omnibus repository's leukemia microarray gene data, which included 22283 genes, was taken for the study of Nazari et al. [11] Python's normalization test and main component analysis were used for the initial prepro-

cessing. Following the design and implementation of a DNN neural network using the data, classifiers cross-validated the results. Their results demonstrate that the PCA gene segregation capacity and cancer cells and healthy cells independence. Their normalization test was much significant (P¿0.05). Their results accuracy for deep neural learning network with three hidden layers and a single-layer neural network are 63.33 and 96.67, respectively.

The potential for the identification of hundreds of novel biomarkers may be constrained by Gupta et al.'s [5] investigation of the use of generative models to recapitulate and expanding scientific knowledge utilizing microarray data. On a dataset with a TB emphasis, they look at the potential of these generative models and assess the effectiveness of generative adversarial networks (GAN), gaussian mixture models (GMM), and variational autoencoders (VAE). They further investigate whether previously referred to axes genes can be used as an efficient method to use domain knowledge while designing those generative models, in order to further reduce biological noise and enhance signals that can be verified by conventional enrichment techniques or functional experiments. The likelihood of identifying hundreds of novel biomarkers may be constrained by Gupta et al.'s investigation on the ability for generative models to replicate and improve scientific knowledge utilizing microarray data. We examine the possibilities of generative models and evaluate the performance of generative adversarial networks (GAN), gaussian mixture models (GMM), and variational autoencoders (VAE) on a dataset with a TB emphasis. To reduce biological noises and enhance signals that can be verified by conventional enrichment techniques or functional an experiments, we investigate whether previously referred to as axes genes can be used as an effective strategy to employ domain knowledge while designing those generative models.

Shen et al. [14] give a succinct overview of several deep learning applications to genomic research in this review paper. As a group of machine learning tech-

niques, deep learning can be divided into supervised learning and unsupervised learning. We begin by outlining the fundamentals of supervised, unsupervised, and semi-supervised learning. We then go over common deep learning techniques and how they are used in genomic research. The evaluation primarily focused on traditional deep learning techniques, particularly those with the potential to be employed for genomic data analysis because of the enormous number of deep learning techniques that are now accessible and the space constraints.

2.3 Image-based Approaches for Leukemia Classification

In order to classify the 33 tumor types, Lyu et al. [9] employed a convolutional neural network to integrate the high dimensional RNA-Seq data into 2-D pictures. 95.59% was the final accuracy we obtained. Additionally, using the concept of Guided Grad Cam, we created important heat maps for every gene according to each class. The success of our method was demonstrated by methodical analysis of the genes with high intensities in the heat maps, which confirmed that these top genes are associated with tumor-specific pathways and some of thepathways have already been used as biomarkers.

Using 85 microscopic blood pictures obtained from bone marrow sequence of Multiple Myeloma (MM) patients, and the suggested research by Kamma et al. [6] offers a reliable method for the disease's prediction among the many types of blood malignancies. The proposed work uses Convolutional Neural Networks (CNN) to eliminate the possibility of errors in the manual feature extraction process. This follows the training the model using Artificial Neural Networks (ANN), Support Vector Machines (SVM), and Random Forest algorithms for classification. We used Random Search Optimizer in conjunction with SVM and Random Forest to identify the optimal set of hyperparameters for improved outcomes. For the CNN-ANN model, the overall accuracy was noted to be 100%. As a result, the model can be deployed to determine the Multiple Myelomas from the images of cells.

3 Dataset

We have used the Leukemia_GSE28497 dataset that contains 281 samples consisting of 22,285 genes (features) of 7 target classes. The Leukemia GSE28497 dataset is a gene expression dataset that has been used widely in the field of cancer research, specifically for studying leukemia. This dataset is publicly available and can be accessed from various repositories, such as the Gene Expression Omnibus (GEO).

Here are some key points about the Leukemia GSE28497 dataset:

1. Dataset Origin: The dataset was generated from gene expression profiling experiments using the microarray technology. It includes gene expression data from bone marrow samples of patients who are diagnosed with acute lymphoblastic leukemia (ALL) and acute myeloid leukemia (AML), as well as samples from healthy individuals serving as controls.

2. Dataset Size: The Leukemia GSE28497 dataset typically contains gene expression measurements for thousands of genes (features) across multiple samples. The exact number of samples and genes can vary, depending on the specific version and processing of the dataset.

3. Purpose: The dataset was collected to investigate the gene expression patterns associated with different types of leukemia. Researchers have used this dataset to identify differentially expressed genes, discover biomarkers, and gain insights into the molecular mechanisms underlying leukemia development and progression.

4. Clinical Information: Along with the gene expression data, the Leukemia GSE28497 dataset may also include additional clinical information about the patients, such as age, gender, subtype of leukemia, and treatment response. This information can be useful for exploring correlations between gene expression patterns and clinical outcomes.

5. Data Availability: The Leukemia GSE28497 dataset is publicly available, and researchers can access and download the data from repositories like GEO. It is

typically provided in a tabular format, with rows representing genes and columns representing samples, along with metadata providing additional information.

4 Proposed Methodology

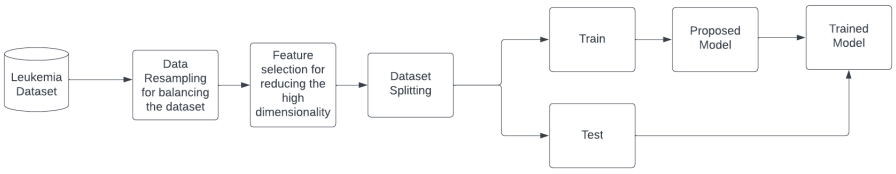

Figure 5: Our Proposed Methodology

- **Data Resampling:** Different resampling techniques are being tested for balancing the imbalanced dataset.
- **Feature Selection:** For reducing the very high dimensionality of the data.
- **Test & Train Data:** Splitting the preprocessed data into test and train set for feeding into the network.
- **Proposed Model:** Using a deep learning model to train and test the processed data. TabNet, 1DCNN, and LSTM are our models of preference.

4.1 Data Resampling

Machine learning algorithms do not take class distribution into account since they prefer to improve accuracy by decreasing the error. Examples of this issue are common in fraud detection, anomaly detection, facial recognition, etc.

Two of the common methods of Resampling we can use are –

- **Cross Validation**
- **Bootstrapping**

Cross-validation is a technique used to assess a model's performance by estimating the test error linked to it.

The most fundamental strategy is the validation set approach. The dataset is simply split into two portions at random: a training set and a validation set or hold-out set. On the training set, the model is fitted, and on the validation set, predictions are made using the fitted model.

The validation set approach is inferior to leave-one-out-cross-validation (LOOCV). One observation is utilized for validation, while the remaining observations are used to fit the model, rather than splitting the entire dataset in half.

The set of observations is randomly divided into k folds of almost similar size in the k-fold cross-validation procedure. The model is fitted on the remaining folds using the initial fold as a validation set. The process is then carried out k times, with a different group being used as the validation set each time.

4.2 Feature Selection

A machine learning model can only be created using a small subset of the dataset's variables; the remainder are either unnecessary or unimportant. If all this useless and redundant data is included in the dataset, the overall performance and accuracy of the model may deteriorate. It is essential to identify and pick the most appropriate features from the data in order to remove the superfluous or less important information, which is performed with feature selection in machine learning.

Feature selection, which can be carried out manually or automatically, is the process of choosing the subset of the most acceptable and pertinent features to be used in model creation. The original features of the dataset are kept, and either new, significant features are added, or old, unimportant features are removed.

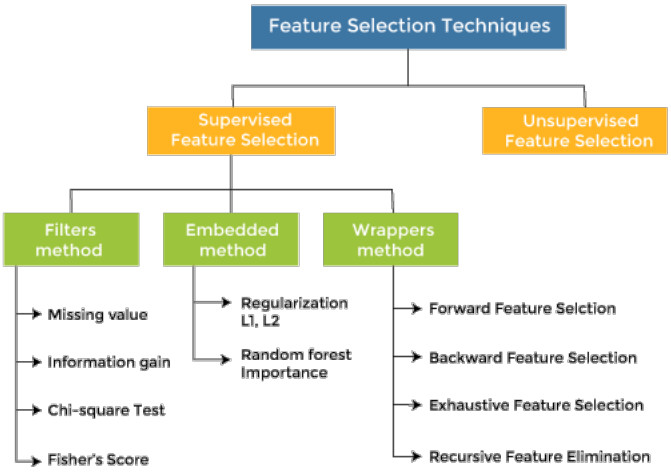

Figure 6: Feature Selection Methods

Techniques for supervised feature selection that take into account the target variable can be used for the labeled dataset. Unsupervised feature selection methods may be used to unlabeled datasets and ignore the target variable.

The filter method's features are selected using statistical measures. In this approach, the features are chosen during a pre-processing stage that is separate from the learning algorithm. The filter approach uses various metrics through ranking to remove unnecessary columns and features from the model. Filter methods have the benefit of requiring less calculation time and not overfitting the data.

1. Missing Value: When there is no data for one or more features in a dataset, it is said that there are "missing values." Because missing values can add bias and affect how well machine learning models perform, handling them properly is essential. Examining the connection between missing values and the target variable is one strategy for feature selection. Even if a feature has missing values, if it has a strong correlation with the objective variable, it might still be valuable and kept for further study. On the other hand, a feature might be dropped if its missing values have no bearing on the target variable.

2. Information Gain: The amount of information gained about the target

variable by knowing the value of a specific attribute is measured by the information gain concept from information theory. It is frequently employed in decision tree-based feature selection techniques. The difference between the entropy (a measurement of impurity or disorder) before and after the dataset was divided based on a specific attribute is calculated as information gain. Since they provide a greater contribution to lowering the uncertainty in predicting the target variable and are therefore more likely to be chosen, features with higher information gain are seen as being more informative.

3. Chi-Square Test: A statistical technique called the chi-square test is used to assess whether two categorical variables are independent or related. The chi-square test can be used to evaluate the association between each feature and the target variable in feature selection. It determines the chi-square statistic, which gauges the discrepancy between the frequency of the feature's values as seen and expected. The relevance of the feature for prediction is increased by a larger chi-square value, which denotes a greater relationship between the feature and the target variable. As a result, features with higher chi-square values are frequently chosen for further study since they are thought to be more significant.

4. Fisher's Score: A feature selection technique that is frequently used in classification problems is Fisher's score, also referred to as the Fisher criterion or Fisher's discriminant ratio. It seeks to identify traits that show a large mean difference between classes and a little variance within each class. By dividing the between-class variance by the within-class variance, the Fisher score is determined. Higher Fisher scores for features are thought to be more useful for classification because they help distinguish between various classes more. As a result, features with high Fisher scores are frequently chosen to be part of the model.

Embedded approaches combined the benefits of the filter methods and wrapper methods by considering feature interaction and low processing costs. Similar to

the filter method but faster, these quick processing methods are more precise.

5. L1 Regularization (Lasso): L1 regularization, also known as Lasso regularization, is a technique used to penalize the model for having too many irrelevant features. It adds a penalty term to the loss function, which encourages the model to minimize the absolute values of the feature weights. As a result, L1 regularization tends to drive the weights of irrelevant features to zero, effectively eliminating them from the model. This property makes L1 regularization useful for feature selection, as it helps identify the most important features.

6. L2 Regularization (Ridge): Ridge regularization, commonly referred to as L2 regularization, is another method for reducing overfitting and enhancing model generalization. Like L1 regularization, L2 regularization adds a penalty term that encourages the model to minimize the squared values of the feature weights rather than penalizing the absolute values of the weights. In contrast to L1 regularization, L2 regularization reduces the weights of less significant features toward zero but does not require them to be exactly zero. As a result, L2 regularization does not explicitly perform feature selection, but it nevertheless aids in minimizing the influence of unimportant characteristics on the model.

Several machine learning methods, including linear regression, logistic regression, and support vector machines can be regularized using both L1 and L2 techniques. The exact situation at hand and the desired result determine which L1 or L2 regularization should be used. While L2 regularization reduces all feature weights toward zero but typically does not make them completely zero, L1 regularization tries to yield sparse models by setting some feature weights to zero.

7. Random Forest Importance: The ensemble learning technique Random Forest mixes several decision trees to produce predictions. A feature selection method called Random Forest Importance uses the Random Forest algorithm to

assess the significance of each feature. Random Forest determines the impurity reduction that each feature achieves when it is utilized to split nodes in the trees during the training phase. The average impurity decrease across all of the trees in the Random Forest is then used to calculate a feature's relevance. The predictions of the model are thought to be more influenced by features with higher significance scores.

Given the interactions and connections between features in a dataset, Random Forest Importance offers a reliable indicator of feature importance. It may be used for both of the classification tasks and regression tasks and can handle a variety of data sources, including categorical and numerical data. It enables efficient feature selection by ranking the features according to their relevance ratings, making it possible to find the features that are most pertinent to the model.

The wrapper methodology treats the selection of characteristics as a search issue, where numerous combinations are made, evaluated, and compared to other choices. The algorithm is trained by using the subset of features repeatedly.

8. Forward Feature Selection: Starting with an empty set of features, forward feature selection adds features one at a time based on a predefined criterion. The algorithm chooses the feature that best improves the model at each iteration by assessing how well the model performs with the features that are currently picked. This procedure keeps going until a predetermined stopping criterion has been reached, such as when the target number of features is attained or performance starts to decline. A greedy technique called forward feature selection continuously increases the feature set while prioritizing the most promising characteristics.

9. Backward feature selection: In each iteration, one feature is eliminated at a time after starting with the entire collection of features. A criterion is used to assess the performance of the model after each feature has been removed, similar

to forward feature selection. Based on a criterion that minimizes performance loss, the algorithm decides which feature to remove. This procedure continues until a stopping criterion is attained, such as when the target number of features is attained or when performance starts to noticeably deteriorate. In a greedy algorithm, backward feature selection operates in the opposite manner from forward selection.

10. Exhaustive Feature Selection: Exhaustive feature selection entails analyzing every conceivable feature subset from the provided dataset, as the name suggests. It looks through every possible feature combination and assesses the model's performance for each subset. The best subset of characteristics is chosen using the performance criterion, such as accuracy or error rate. Exhaustive feature selection can be computationally expensive and is often doable for small feature spaces due to the combinatorial nature of the process. It is exhaustive, but it ensures that the best feature subset will be found based on the given criterion.

11. Recursive Feature Elimination (RFE): Recursive feature elimination is a feature selection technique that iteratively removes features from the dataset to identify the best ones. It builds a model using the chosen features after starting with the entire collection of features. The feature(s) deemed to be of the least value are then removed after each characteristic has been evaluated for importance or weight. Up until the required performance level or predetermined number of features are met, the procedure iteratively continues. Different criteria, such as coefficient magnitudes in linear models or feature importances in tree-based models, can be used in recursive feature elimination to determine how important a feature is.

4.3 Proposed Models

By using matrix factorization, deep neural network (DNN) models can get past these issues. For the adaptability of the input layer of the network, DNNs may

add query characteristics and item features, that can later help identify each user's unique interests and increase the probability of relevant suggestions.

4.3.1 Logistic Regression

Using microarray gene datasets, the statistical modeling technique known as logistic regression is widely used to forecast the presence or the absence of blood cancer. An easy-to-understand method for examining the connection between gene expression levels and the likelihood of blood cancer is through the use of logistic regression.

A mathematical formulation that describes the link between the predictor variables (gene expression levels) and the probability of the binary outcome (the presence or the absence of blood cancer) makes up the logistic regression architecture. Let's investigate the logistic regression architecture using math:

1. Binary Outcome Representation:

In logistic regression, the binary outcome variable (presence or absence of blood cancer) is represented as Y, which takes on two values: 0 or 1. $Y = 1$ indicates the presence of blood cancer, while $Y = 0$ represents the absence of blood cancer.

2. Predictor Variables:

The microarray gene dataset provides a set of predictor variables, represented as $X = (X_1, X_2, ..., X_p)$, where p is the number of genes or gene expression levels. Each X corresponds to the expression level of a specific gene in the dataset.

3. Logistic Regression Model:

The logistic regression model gets to assume that the log-odds of the probability of Y being 1 can be expressed as a linear combination for the predictor variables. Mathematically, the logistic regression model can be represented as:

$$logit(P(Y = 1)) = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + ... + \beta_p X_p$$

In the equation above, P(Y=1) represents the probability of Y being 1, and

$logit(P(Y=1))$ is the natural logarithm of the odds ratio. The coefficients $\beta_0, \beta_1, \beta_2, ..., \beta_p$ are the parameters to be estimated by the logistic regression model. They represent the effect of each predictor variable (gene expression level) on the log-odds of the outcome variable.

4. Probability Estimation:

To convert the log-odds into a probability, the logistic function (also known as the sigmoid function) is applied. The logistic function is defined as:

$P(Y=1) = \frac{1}{1+e^{(-logit(P(Y=1)))}}$

The logistic function maps the range of the $log - odds(-\infty, +\infty)$ to a probability range between 0 and 1. When the log-odds are positive, the probability tends towards 1, indicating a higher likelihood of Y being 1 (presence of blood cancer). Conversely, when the log-odds are negative, the probability tends towards 0, indicating a higher likelihood of Y being 0 (absence of blood cancer).

5. Parameter Estimation:

The parameters $\beta_0, \beta_1, \beta_2, ..., \beta_p$ in the logistic regression model are estimated using the maximum likelihood estimation (MLE). The MLE seeks to find the values for the parameter that maximizes the very likelihood of observing the given data. This involves finding the values of $\beta_0, \beta_1, \beta_2, ..., \beta_p$ that maximizes the joint probability of observing the binary outcomes for the given predictor variables.

Different optimization algorithms, such as gradient descent or Newton's method, can be used to iteratively update the parameter values based on the observed data until convergence is reached in order to estimate the parameters.

6. Prediction:

Once the parameters are estimated, they can be used to predict the likelihood of the binary outcome (presence or absence of blood cancer) for new observations. A decision threshold can be set to classify the observations as either $Y = 1$ or

$Y = 0$, based on whether the predicted probability exceeds the threshold.
Based on microarray gene datasets, logistic regression gives a simple and understandable method for predicting blood cancer. Logistic regression can offer useful insights into the connection between particular genes and the chance of the disease by simulating the relationship between gene expression levels and the likelihood of blood cancer.

4.3.2 Neural Networks

In order to forecast blood cancer, neural networks have become potent models for assessing complicated datasets, such as microarray gene datasets. For the investigation of gene expression levels and their correlations with disease outcomes, neural networks are particularly good at capturing complex patterns and interactions within high-dimensional data.
A neural network can be created specifically for the goal of predicting the presence or absence of blood cancer by learning the underlying patterns in the microarray gene collection. The nodes (sometimes referred to as neurons) in the layers that make up the neural network architecture process and transform the input data.
A neural network for blood cancer prediction frequently has an input layer, one or more hidden layers, and an output layer. Each layer is composed of several neurons that are connected to the neurons in the layers below by weighted connections. For binary classification tasks, the number of neurons in the output layer is frequently set to 1, signifying the expected likelihood of blood cancer, whereas the number of neurons in the input layer is determined by the number of gene expression features in the microarray dataset.
Mathematically, the neural network architecture can be represented as follows:
Neurons in the input layer are the ones that receive the microarray dataset's gene expression levels. Each input layer neuron is associated with a particular aspect of gene expression, denoted by the symbol Xi, where i is a number between 1 and the total number of genes in the dataset.
The so-called "hidden layers" are situated between the input and output layers.

They are composed of neurons that alter the incoming data using activation processes. Each neuron in the hidden layer receives input from all the neurons in the layer above, calculates a weighted total of these inputs, and then applies an activation function.

A hidden layer neuron's weighted sum can be written as follows:

$$z_j = \sum \left(w_{ji} * X_i\right) + b_j$$

where z_j is the weighted sum, $w_(ji)$ is the very weight connecting the jth neuron in the hidden layer to the ith neuron in the previous layer, X_i is the input from the previous layer, and b_j is the bias term associated with the jth neuron.

The weighted sum is then passed through an activation function, such as the sigmoid function, ReLU (Rectified Linear Unit), or tanh (hyperbolic tangent), to introduce non-linearities into the neural network. The choice of activation function depends on the specific problem and desired properties of the model.

Output layer: The final prediction is produced by a single neuron in the output layer. After calculating a weighted sum of the inputs from the last hidden layer, an activation function is then used. Because it converts the weighted total to a probability value between 0 and 1, which indicates the chance of blood cancer, the sigmoid activation function is used in binary classification tasks like the prediction of blood cancer.

The weights and biases of the neural network are modified during training to minimize the disparity between the expected outputs and the actual labels in the training dataset. The weights and biases are iteratively updated throughout this procedure, called backpropagation, making the use of an optimization method like gradient descent or its variations.

Similar to other classification models, model performance may be assessed using measures like accuracy, sensitivity, specificity, and AUC-ROC. Additionally, methods like cross-validation can be used to evaluate the neural network's capacity for generalization on new data.

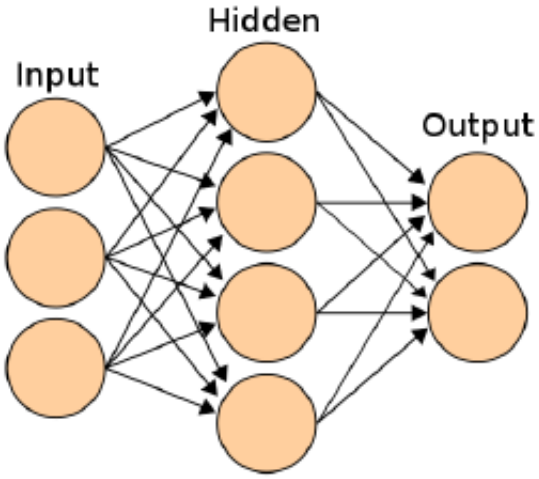

Figure 7: Neural Network

4.3.3 TabNet

Sequential attention is used by TabNet, a deep tabular data learning architecture, to choose which attributes to infer from at each step of the decision-making process. The TabNet encoder consists of a feature transformer, an attentive transformer, and feature masking. A split block is used to partition the processed representation into portions for the final output and the attentive transformer of the subsequent phase. By integrating the feature selection masks, which offer comprehensible details about how the model behaves for each step, the importance of a feature can be assessed worldwide. Blocks of the feature transformer make up the stages of the TabNet decoder.

In order to predict blood cancer, the deep learning model TabNet has attracted attention for its success in processing tabular data, including microarray gene datasets. TabNet is an effective tool for deriving critical insights from gene expression levels and forecasting blood cancer outcomes because it combines the benefits of deep learning and attention mechanisms to understand key features and correlations within the data.

The TabNet architecture consists of several key components, including the feature transformer, the decision step, and the aggregation step. Let's explore each component and its mathematical analysis:

1. Feature Transformer:

The feature transformer is responsible for processing and transforming the input features from the microarray gene dataset. It typically consists of multiple fully connected layers that learn hierarchical depictions of the input data.

Mathematically, the feature transformer can be represented as follows:

$$h(0) = X$$
$$h(t) = f(W(t)h(t-1) + B(t))$$

where h(t) represents the hidden representation at the time step t, X is the input feature vector, W(t) and B(t) are the weights and biases of the fully connected layers at the time step t, and f() is the activation function used to introduce non-linearities.

2. Decision Step:

The decision step incorporates the key idea of TabNet, which is the adaptive and interpretable attention mechanism. This step uses a sparse attention mask to select relevant features and suppress irrelevant ones.

Mathematically, the decision step can be represented as follows:

$$M(t) = g(S(t-1))$$
$$A(t) = softmax(M(t))$$

where M(t) represents the attention mask at time step t, S(t-1) is the shared transformation applied to the hidden representation at the previous time step, and g() is the Gated Linear Unit (GLU) activation function that applies a gating mechanism to the shared transformation.

The attention mask M(t) is obtained by applying the g() function to S(t-1), which allows the very model to automatically learn the importance of each feature for the prediction task. The softmax function is then applied to the attention mask to obtain A(t), which represents the normalized importance weights for each feature.

3. Aggregation Step:

The aggregation step combines the information from both the feature transformer and the decision step to produce the final prediction. It involves a weighted sum of the hidden representations, with the attention weights from the decision step acting as the weights.

Mathematically, the aggregation step can be written as follows:

$$T(t) = A(t) * h(t)$$
$$output = sum(T(t))$$

where T(t) represents the transformed features at time step t, output represents the final prediction, and sum() computes the sum of the transformed features weighted by the attention weights.

The aggregation step enables the model to concentrate on the most pertinent features while ignoring the less significant ones, enhancing the model's interpretability and performance.

To train the TabNet model, a loss function, such as binary cross-entropy, is optimized using methods like backpropagation and gradient descent. To reduce the discrepancy between the anticipated outputs and the true labels, the model learns to change the weights and biases in both the feature transformer and the decision phase.

TabNet has shown promising results in predicting blood cancer outcomes by effectively capturing the complex relationships within microarray gene datasets. Its attention mechanism enables the identification of crucial features and provides interpretability, allowing researchers to gain insights into the genetic mechanisms underlying blood cancer.

4.4 Evaluation Metrics

There are various methods for assessing a machine learning model's performance, and the best one will depend on your dataset's unique properties and the kind of model you're employing. Here are some typical evaluation techniques you might employ:

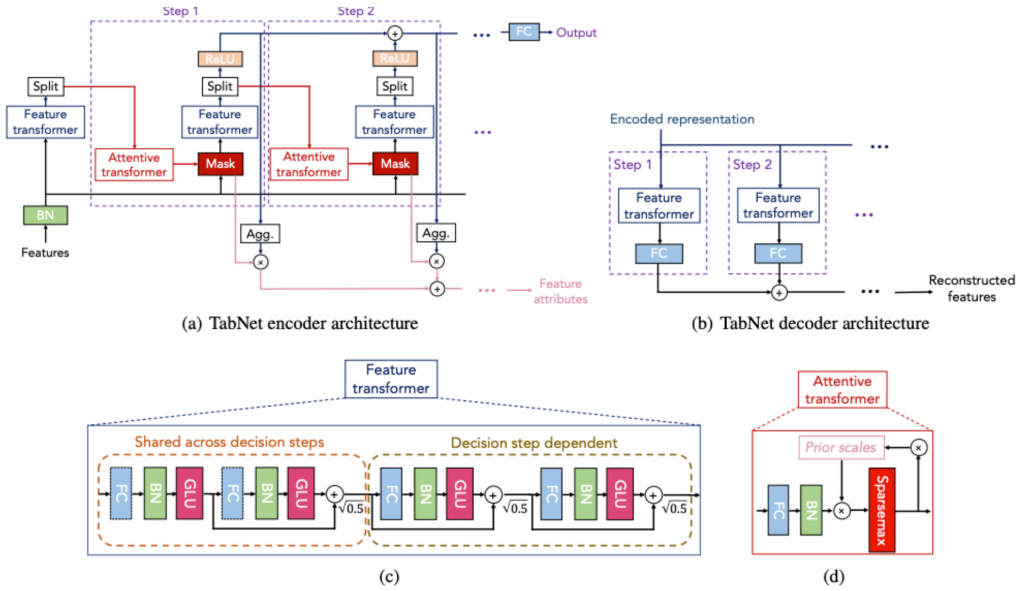

Figure 8: Tabnet Architechture

Training and validation accuracy: Comparing a model's training and validation accuracy is one method of evaluation. The model may be overfitting to the training data and not generalizing well to new cases if it has a high training accuracy but a low validation accuracy.

Training and validation loss: Another way to evaluate a model is to compare its training and validation loss. If the model has a low training loss but a high validation loss, it may be underfitting to the training data and not performing well on the validation set.

Confusion matrix: A confusion matrix can be used to evaluate the performance of a classifier model. A table called a confusion matrix lists the number of accurate positive, false positive, accurate negative, and false false predictions made by the model. Precision, recall, and the f1-score may all be calculated from the values in the confusion matrix.

ROC curve: If your model is a binary classifier, you can evaluate its effectiveness using a receiver operating characteristic (ROC) curve. A ROC curve contrasts the

true positive rate as well as the false positive rate for various categorization criteria. The area under the curve (AUC), which ranges from 0.5 to 1.0 and represents perfect accuracy, is used to assess the model's accuracy.

Cross-validation: Another technique for evaluating models is to use cross-validation. To do this, the dataset is partitioned into a number of folds, the model is trained on some of the folds, and it is then assessed on the other folds. To give a final evaluation of the model's performance, the evaluation scores acquired after several iterations with different folds are averaged.

These are just a few examples of evaluation techniques that we might apply to judge how well a machine learning model is performing. The properties of your dataset and the kind of model we're utilizing will determine the precise strategy we go for.

5 Research Challenges

Microarray statistical classifications are challenging due to overestimation and numerous linearity difficulties, making it challenging to assess microarray data having to use statistical methods to locate high-dimensional data (p¿n). [4]

The inability to be interpreted is one problem with deep learning. Finding and interpreting disease-associated genetic markers is of primary interest in genetic association research. Deep learning is still viewed as a "black box," which makes it difficult to use in genetic association studies.

Other significant issues include the data of imbalance and the large dimensionality of the data. The term "data imbalance" describes a situation where a dataset's sample distribution across various classes is drastically asymmetrical. Data imbalance arises when there is an unequal representation of samples among various blood cancer types or target classes in a microarray gene collection. This implies that a dataset may be uneven since some blood cancer types may have a disproportionately higher number of samples, whereas others may have fewer

samples.

A data imbalance can make it difficult to create reliable predictive models. A machine learning model that has been trained on unbalanced data may show bias in favor of the majority class, which will result in subpar performance when predicting the minority classes. A skewed decision boundary is the outcome of the classifier's propensity to emphasize on the accuracy of the majority class while overlooking the minority classes.

Data imbalance can occur in microarray gene datasets for a variety of reasons, including sample collection biases, the rarity of specific cancer subtypes, or differences in the population's incidence of particular blood cancer kinds. When training a model, it's crucial to manage data imbalance in order to produce predictions that are accurate and fair for all classes. Predictive model performance can be enhanced by employing strategies like undersampling the majority class, oversampling the minority class, or ensemble approaches created especially for unbalanced datasets.

When there are more characteristics or variables in a dataset than there are samples, the dataset is said to be high dimensional. High dimensionality occurs in the context of a microarray gene collection as a result of the enormous number of genes being measured or profiled.

The simultaneous assessment of thousands of genes' levels of gene expression is made possible by microarray technology. Each gene represents a characteristic or variable in the dataset, and each sample's expression level corresponds to a numerical value for that gene. As a result, microarray gene collections frequently contain many genes and features—possibly up to tens of thousands.

Microarray gene datasets' high dimensionality poses a number of problems and is-

sues. First, because analytic methods must process and store a lot of data, working with high-dimensional data can result in computational and memory limitations. Second, it may raise the danger of overfitting, which is when a model does well on training data but stops to generalize to fresh, untested samples.

High dimensionality can also result in the "curse of dimensionality," a problem where data sparsity rises as the number of characteristics does. Finding significant patterns or relationships in the data may become challenging as a result, as the number of samples becomes limited in the high-dimensional feature space.

Several methods of including feature selection and dimensionality reduction techniques such as t-distributed Stochastic Neighbor Embedding (t-SNE) and Principal Component Analysis (PCA) are used to deal with high dimensionality. These techniques try to either project the data into a lower-dimensional space that captures the main qualities or decrease the number of features while retaining the most informative ones.

In order to analyze, interpret, and model gene expression patterns effectively and to improve predictive performance in tasks like disease classification or outcome prediction, it is essential to manage the high dimensionality of microarray gene datasets.

6 Result and Discussion

6.1 Preprocessing

To maintain the quality and integrity of the dataset, managing missing values is frequently a part of preprocessing tabular data. Removing rows or columns that have null (missing) entries is one method for dealing with missing values. To exclude null items from a tabular dataset, use the dropna() function from scikit-learn (sklearn).

Examining the dataset and locating the missing values is crucial before using any preparation procedures. In tabular data, missing numbers are often shown as NaN (Not a Number) or None.
It's a good idea to make a copy of the dataset before making any alterations in order to preserve the integrity of the original dataset. This enables you to work with the updated version while maintaining the integrity of the original data.
To delete rows or columns with null entries from the tabular dataset, use the dropna() function. Rows with any null values are automatically removed by default (axis=0), effectively removing those samples from the dataset.
You can effectively remove the incomplete data points from your dataset by using dropna() to delete rows or columns with missing values. However, it's crucial to take into account the potential information loss and carefully assess how eliminating those entries would affect the effectiveness of your study or model. Alternative approaches to handling missing values may be available depending on the circumstance, such as imputation techniques that estimate and fill in the missing values based on the existing data.

6.2 Evaluation Metrics

FINAL TEST MAE: 1.4253394373676234
FINAL TEST MSE: 3.201251019353927
FINAL TEST MSLE: 0.4609574587660871

The MAE calculates the average absolute difference between the test dataset's true values and the anticipated values. The number of 1.4253394373676234 shows that the absolute difference between the model's predictions and the actual values is typically 1.43 units. Lower MAE values signify greater model performance because they mean that the predictions are more accurate.

The average squared difference of the test dataset's true values from the projected values is calculated by MSE. The average squared difference between the

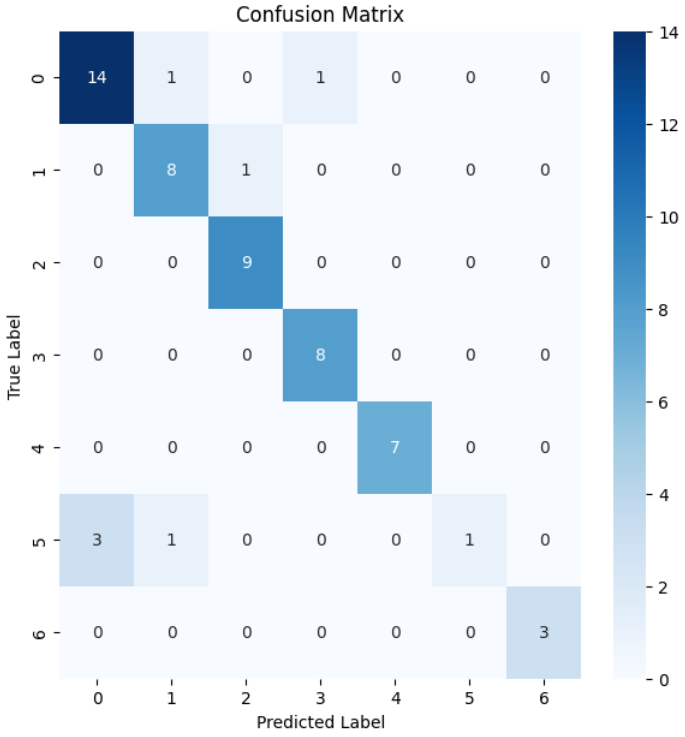

Figure 9: Confusion Matrix for the Logistic Regression

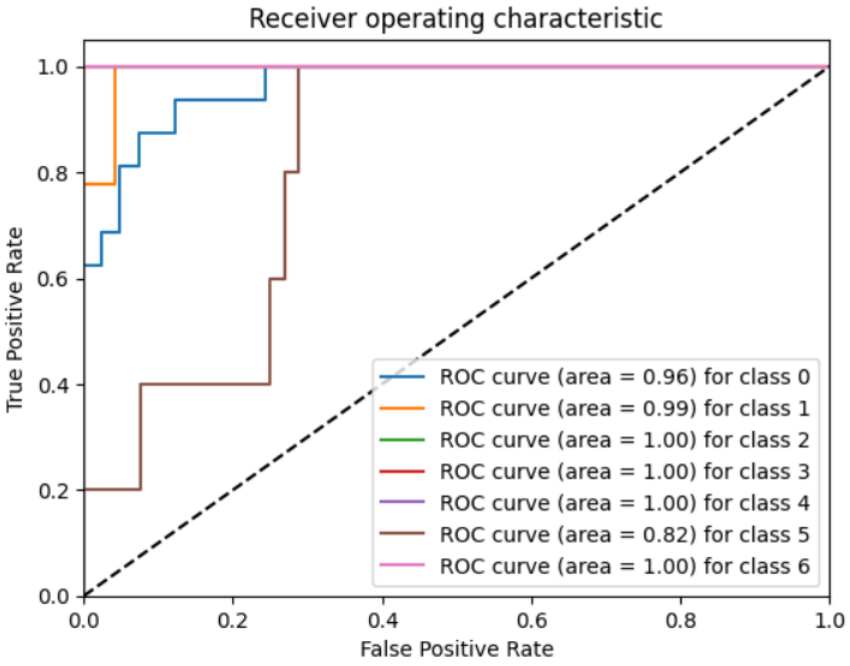

Figure 10: ROC for the Logistic Regression

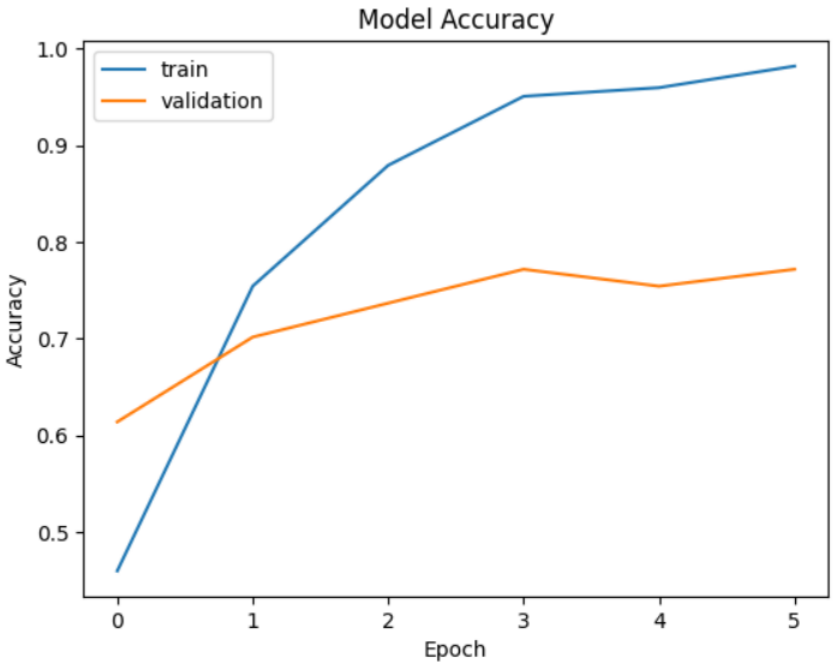

Figure 11: Model Accuracy for the Neural Network

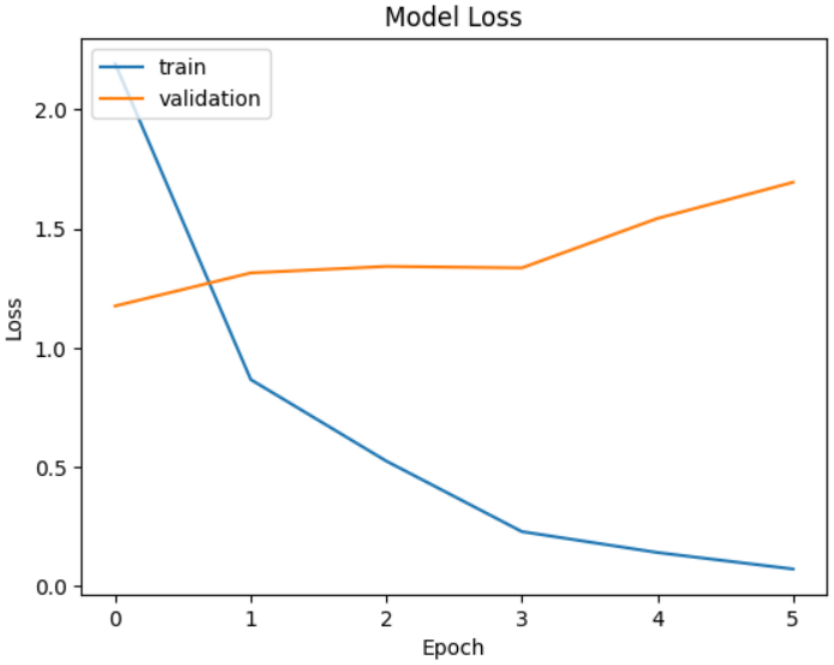

Figure 12: Model Loss for the Neural Network

predicted values and the actual values is 3.201251019353927. Given that the differences are squared, MSE is sensitive to greater errors. It consequently tends

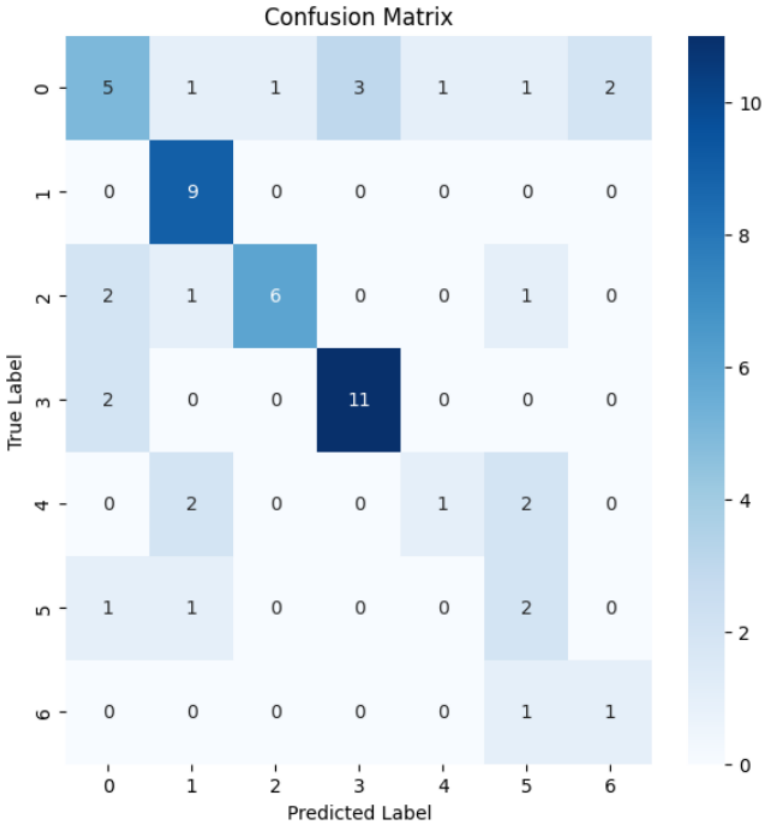

Figure 13: Confusion Matrix for the Neural Network

to penalize significant prediction errors more severely. Lower MSE values signify higher model performance since, on average, they signal that predictions are more accurate.

The average squared logarithmic difference between the test dataset's true values and the anticipated values is what MSLE calculates. The average squared logarithmic difference of the forecasts from the actual values is 0.4609574587660871. When the desired variable spans multiple orders of magnitude, MSLE is helpful. Lower MSLE values signify better model performance since they mean that, on a logarithmic scale, the predictions are more in line with the true values.

Accuracy: 0.2631578947368421

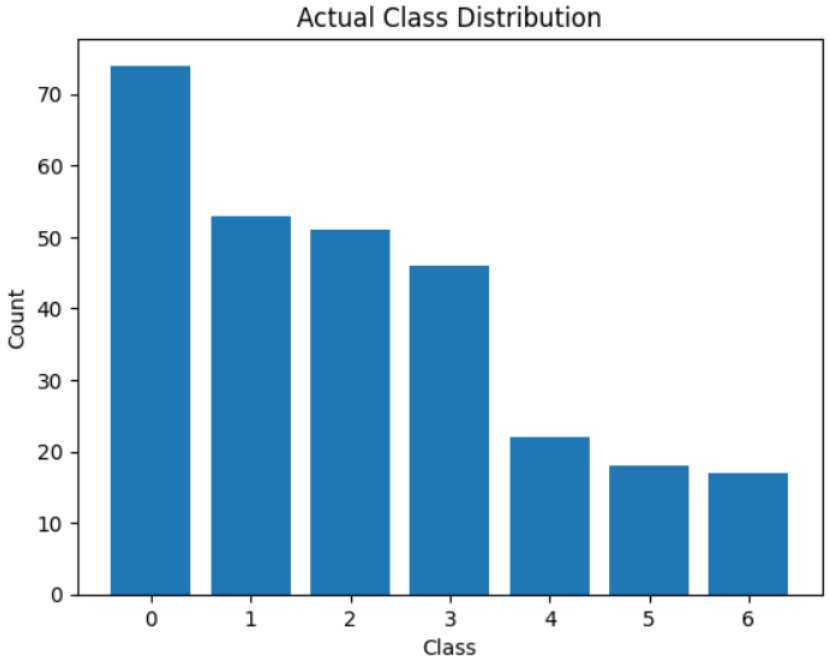

Figure 14: Actual Class Distribution for the Neural Network

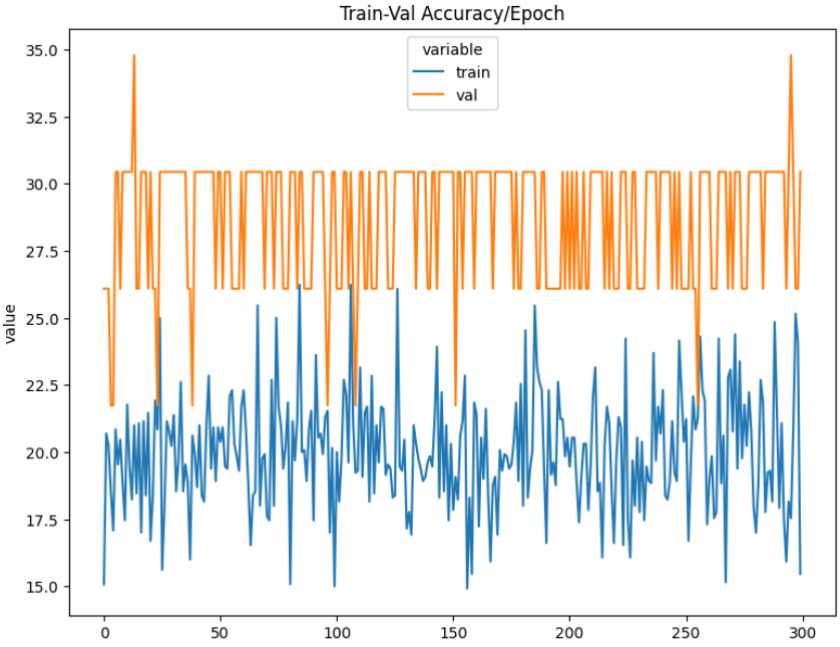

Figure 15: Train Value Accuracy for the TabNet

A frequently used indicator for classification tasks is accuracy, which quantifies the percentage of properly predicted occurrences among all examples. According to the value of 0.2631578947368421, the accuracy of this model on this dataset

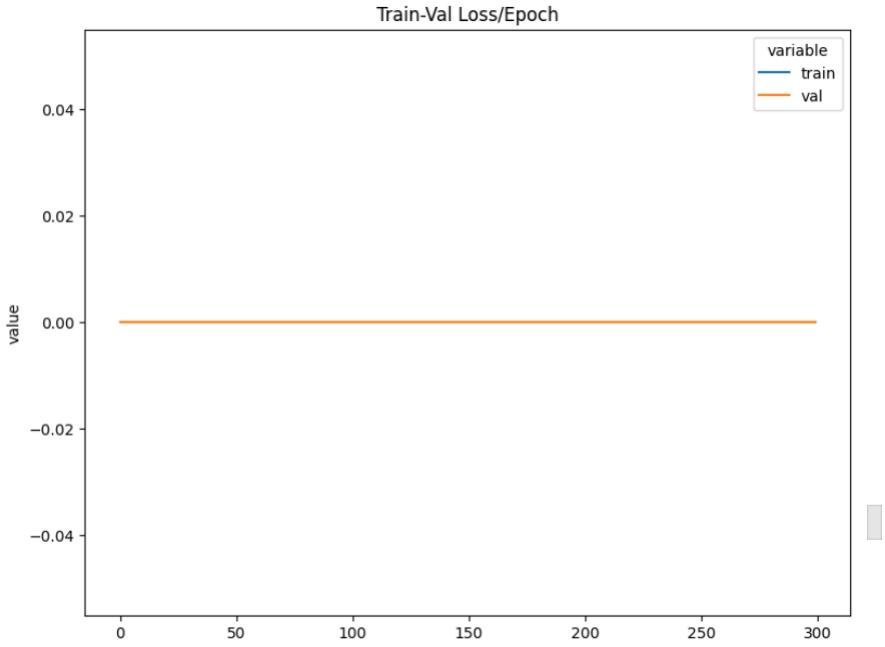

Figure 16: Train Loss Accuracy for the TabNet

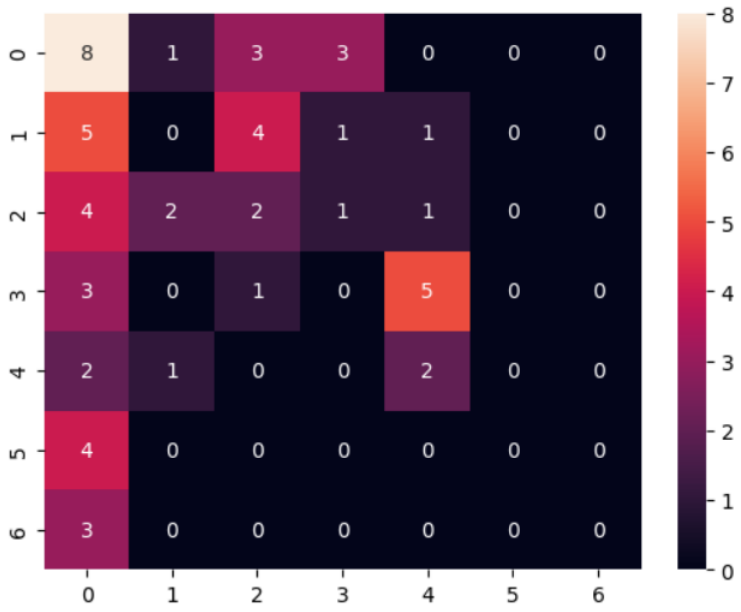

Figure 17: Confusion Matrix for the TabNet

or test set it was evaluated on was roughly 26.32%. In other words, for around 26.32% of the dataset's cases, the model's predictions were accurate.

7 Conclusion

In conclusion, the analysis of the Leukemia GSE_28497 gene microarray dataset was the main focus of our study. We used null entry removal during the preprocessing stage to assure the dataset's integrity by removing any missing values. The dataset was then trained using three distinct models: TabNet, single-layer neural network, and logistic regression.

Our findings showed that all of the models performed well, though to different degrees of accuracy. The highest accuracy, 87%, was shown using logistic regression, demonstrating its usefulness in categorizing leukemia subtypes based on gene expression patterns. With a respectable accuracy of 72.2%, the single-layer neural network exhibited its capacity to recognize intricate correlations in the data. However, TabNet only managed a 26% accuracy rate, indicating that it had trouble identifying the dataset's underlying patterns.

These results emphasize how crucial it is to choose the right models for classifying leukemia subtypes. In our investigation, logistic regression proved to be the most reliable method and offered insightful information about the gene expression profiles linked to various leukemia kinds. The single-layer neural network also performed well, although not better than logistic regression. However, TabNet's relatively poor accuracy suggests that the architecture for this particular dataset needs more investigation or improvement.

8 Future Plans

There are a number of potential future directions to take into consideration for further research and project improvement in light of the data and conclusions from our thesis project. Here is a future plan that identifies possible topics to concentrate on:

1. Expansion of the dataset: Even if the Leukemia GSE28497 dataset offered insightful information, it is worthwhile to investigate other freely accessible gene expression datasets associated with leukemia. Multiple datasets can be used to increase the generalizability of created models and provide a more thorough understanding of leukemia subgroups.

2. Investigating feature selection and engineering methods particular to the gene expression data is vital to improve the performance of the classification models. Finding the most informative of features for precise classification might be aided by investigating techniques like differential gene expression analysis, dimensionality reduction, or applying domain knowledge.

3. Model optimization: To further increase the precision of the classification models, take into account investigating more complicated deep learning architectures or advanced machine learning approaches. To take advantage of the intricate relationships seen in the gene expression data, methods such as ensemble learning, deep neural networks, or transfer learning can be researched.

4. Comparative Analysis: Extend the comparative analysis by using more machine learning techniques which are frequently used in the work of cancer. To determine the most efficient method for leukemia subtype classification, contrast the performance of logistic regression, neural networks, and TabNet with alternative methods like random forests, support vector machines, or gradient boosting.

References

[1] Ahmed Abd El-Nasser, Mohamed Shaheen, and Hesham El-Deeb. Enhanced leukemia cancer classifier algorithm. In *2014 Science and Information Conference*, pages 422–429. IEEE, 2014.

[2] SK Abdullah, SK Hasan, and Ayatullah Faruk Mollah. Acute leukemia subtype prediction using eodclassifier. In *Intelligent Data Communication Technologies and Internet of Things*, pages 129–137. Springer, 2022.

[3] Daniel Castillo, Juan Manuel Galvez, Luis J Herrera, Fernando Rojas, Olga Valenzuela, Octavio Caba, Jose Prados, and Ignacio Rojas. Leukemia multiclass assessment and classification from microarray and rna-seq technologies integration at gene expression level. *PloS one*, 14(2):e0212127, 2019.

[4] Kun-Huang Chen, Kung-Jeng Wang, Kung-Min Wang, and Melani-Adrian Angelia. Applying particle swarm optimization-based decision tree classifier for cancer classification on gene expression data. *Applied Soft Computing*, 24:773–780, 2014.

[5] Ayushi Gupta, Saad Ahmad, Atharva Sune, Chandan Gupta, Harleen Kaur, Rintu Kutum, and Tavpritesh Sethi. Evaluating sample augmentation in microarray datasets with generative models: A comparative pipeline and insights in tuberculosis. *bioRxiv*, 2021.

[6] Sai Pavan Kamma, Guru Sai Sharma Chilukuri, Guru Sree Ram Tholeti, Rudra Kalyan Nayak, and Tapaswi Maradani. Multiple myeloma prediction from bone-marrow blood cell images using machine learning. In *2021 Emerging Trends in Industry 4.0 (ETI 4.0)*, pages 1–6. IEEE, 2021.

[7] Serhat Kilicarslan, Kemal Adem, and Mete Celik. Diagnosis and classification of cancer using hybrid model based on relieff and convolutional neural network. *Medical hypotheses*, 137:109577, 2020.

[8] Igor Kononenko. Machine learning for medical diagnosis: history, state of the art and perspective. *Artificial Intelligence in medicine*, 23(1):89–109, 2001.

[9] Boyu Lyu and Anamul Haque. Deep learning based tumor type classification using gene expression data. In *Proceedings of the 2018 ACM international conference on bioinformatics, computational biology, and health informatics*, pages 89–96, 2018.

[10] Pradeep Kumar Mallick, Saumendra Kumar Mohapatra, Gyoo-Soo Chae, and Mihir Narayan Mohanty. Convergent learning–based model for leukemia classification from gene expression. *Personal and Ubiquitous Computing*, pages 1–8, 2020.

[11] Elham Nazari, Amir Hossein Farzin, Mehran Aghemiri, Amir Avan, Mahmood Tara, and Hamed Tabesh. Deep learning for acute myeloid leukemia diagnosis. *Journal of Medicine and Life*, 13(3):382, 2020.

[12] Mary-Elizabeth Percival, Catherine Lai, Elihu Estey, and Christopher S Hourigan. Bone marrow evaluation for diagnosis and monitoring of acute myeloid leukemia. *Blood reviews*, 31(4):185–192, 2017.

[13] Vaibhav Rupapara, Furqan Rustam, Wajdi Aljedaani, Hina Fatima Shahzad, Ernesto Lee, and Imran Ashraf. Blood cancer prediction using leukemia microarray gene data and hybrid logistic vector trees model. *Scientific Reports*, 12(1):1–15, 2022.

[14] Xiaoxi Shen, Chang Jiang, Yalulu Wen, Chenxi Li, and Qing Lu. A brief review on deep learning applications in genomic studies. *Frontiers in Systems Biology*, page 10.

[15] Yawen Xiao, Jun Wu, Zongli Lin, and Xiaodong Zhao. A deep learning-based multi-model ensemble method for cancer prediction. *Computer methods and programs in biomedicine*, 153:1–9, 2018.

[16] Seung-Min Yoo, Jong-Hyun Choi, Sang-Yup Lee, and Nae-Choon Yoo. Applications of dna microarray in disease diagnostics. *Journal of microbiology and biotechnology*, 19(7):635–646, 2009.