Analysis of Gene Expression Cancer Data Set: Classification of TCGA Pan-cancer HiSeq Data

Yusaku Nitta
Department of Computer Science
North Dakota State University
Fargo, USA
yusaku.nitta@ndsu.edu

Mitchell Borders
Department of Computer Science
North Dakota State University
Fargo, USA
mitchell.borders@ndsu.edu

Simone A. Ludwig
Department of Computer Science
North Dakota State University
Fargo, USA
simone.ludwig@ndsu.edu

2021 IEEE International Conference on Big Data (Big Data) | 978-1-6654-3902-2/21/$31.00 ©2021 IEEE | DOI: 10.1109/BigData52589.2021.9671793

***Abstract*—In our research, supervised machine learning algorithms were applied to analyze and compare their capability of cancer classification. Our research used eight machine learning algorithms: Decision Tree, Gradient Boosting, K-Nearest Neighbors, Logistic Regression, Naïve Bayes, Neural Network, Random Forest, and Support Vector Machine. Machine learning models were generated by training the algorithms on the TCGA Pan-cancer HiSeq data set. This data set is an RNA sequencing (RNA-seq) data set consisting of five separate cancer types such as breast invasive carcinoma (BRCA), colon adenocarcinoma (COAD), kidney renal clear cell carcinoma (KIRC), lung adenocarcinoma (LUAD), and Prostate adenocarcinoma (PRAD). The data set was preprocessed with feature selection, oversampling, and normalization techniques. The preprocessed methods were implemented by selecting only the best features and thus removing the inconsequential features, balancing the sample size of each cancer type, and rescaling the values of numeric attributes. Our goal was to determine which algorithm generates a classification model that shows the best performance when categorizing cancer types by employing the following evaluation measures: accuracy, precision, recall, area under curve (AUC) score, F-1 score, and processing time.**

***Index Terms*—RNA Sequencing, TCGA Pan-cancer HiSeq data, breast invasive carcinoma, colon adenocarcinoma, kidney renal clear cell carcinoma, lung adenocarcinoma, and Prostate adenocarcinoma.**

I. Introduction

Cancer refers to a general term that describes diseases caused by abnormal cell growth and proliferation coupled with metastatic and invasive traits [1]. Most of the cancers fall into three major categories: carcinomas, sarcomas, and leukemia or lymphomas. Carcinoma is the most common type of cancer that develops from epithelial cells and accounts for 90% of cancers seen in the human body. Carcinoma can be divided into subgroups that include adenocarcinoma, adenosquamous carcinoma, anaplastic carcinoma, large cell carcinoma, and squamous cell carcinoma [2]. Cancer is widely known as one of the leading causes of death in the world. According to statistics from the Global Cancer Observatory, approximately 10 million people died from cancer worldwide in 2020, indicating lung cancer (1.8 million deaths), colon and rectum cancer (935,000 deaths), and liver cancer (830,000 deaths) as the top three major causes of cancer death [3]. Furthermore, a prediction of the World Health Organization (WHO) indicated that the incidence of cancer may be increased by over 70% within the next two decades. Hence, investigating the genetic background of cancer and developing cancer therapeutic methods are urgently needed in the medical domain [1]. However, scientists have identified more than 277 types of cancer, and each of which requires a specific treatment [1], [4]. Considering the increase of cancer cases and the complexity of cancer diagnosis and treatment, it is of critical importance to invent an efficient method for identification and the early detection of cancer types.

In response to the critical need for cancer diagnosis, gene expression analysis provides ways to enhance the performance of the diagnosis and cancer classification. Gene expression is defined as a highly regulated biological mechanism in which the information encoded in a gene is converted to synthesize a functional gene product such as a protein, transfer ribonucleic acid (tRNA), or small nuclear RNA (snRNA) [5]. The function and adaptability of all living cells are regulated by this process. Hence, gene expression data tells the level of gene activity in tissue and provides information about cellular activities [5] [6]. This data is obtained by measuring the activation and function of genes during the translation process. Cancers are caused by genetic abnormalities, and gene expression data can detect and display these abnormalities. Pattern analysis of gene expression enhances the diagnosis and classification of risk for many cancers [6]. Therefore, evaluating gene expression data is an effective way to identify genes that are associated with cancer regulation and progression [5].

The Cancer Genome Atlas (TCGA) is a landmark joint project that collects genomics cancer data set including gene expression, microRNA expression, and protein expression [2]. This project was conducted by the National Cancer Institute (NCI) and the National Human Genome Research Institute (NHGRI) [7]. TCGA's primary aims are to create, quality control, assemble, investigate, and interpret molecular profiles at DNA, RNA, protein, and epigenetic levels for various kinds of clinical tumors that represent multiple tumor types and subtypes [8]. In 2006, TCGA began as a pilot project focused on three cancer types: glioblastoma, lung, and ovarian. After completing the production phase in 2009, TCGA assembled and classified 11,000 samples across 33 cancer types in the following decade due to the success of the initial project [7].

Those large data sets have offered a great opportunity for the classification and detection of cancer-specific molecular alteration [8]. In 2012, TCGA launched the Pan-Cancer analysis project, which was aimed to collect consistent TCGA data sets of various cancer types and to analyze and interpret these data [8].

Machine learning is an excellent way to investigate gene expression data for cancer analysis. In computer science, machine learning is a branch of artificial intelligence that provides systems with the ability to learn and enhance their performance from experience. Machine learning enables the systems to extract insights or useful information from data and make decisions without any external help [9]. To intelligently investigate data and to improve real-world applications, such as healthcare and COVID-19, machine learning algorithms are the key [10]. Recently in the artificial intelligence field, machine learning methods have developed clinical decision support systems to analyze gene expression, which has made advances in the medical prognosis for cancer. Regarding cancer research, studies have reported the high accuracy of machine learning algorithms in predicting cancer survival [6]. Machine learning methods are mainly categorized into three types, and supervised learning is one paradigm of the methods. Supervised Learning is used for interpreting the input-output relationship information of a system based on a given set of training samples that contain both independent values (inputs) and a dependent value, or a label of input (an output). The primary objective of supervised learning is to generate a model that learns the connection between the input and the output and can predict the output given unseen inputs [11]. Analyzing gene expression data by machine learning methods may contribute to the development of effective cancer identification and classification strategies for early cancer treatment.

Thus, in this paper we are investigating the TCGA Pan-cancer HiSeq data set that consists of five separate cancer types such as breast invasive carcinoma, colon adenocarcinoma, kidney renal clear cell carcinoma, lung adenocarcinoma, and Prostate adenocarcinoma. Instead of only looking at one cancer type at a time, in this investigation we are looking at all five cancer types at once and train a classifier that can correctly identify unknown gene expression data. We will be applying a series of different combinations of preprocessing methods and machine learning algorithms in order to identify the best classification model.

The remainder of the article is structured as follows. The next section (Section II) presents the related work that is based on the pan-cancer project data set. Then, the methods used are discussed in detail in Section III, specifically the data set, data preprocessing methods, and proposed classification algorithms are outlined. Section IV describes the experiments and the results obtained for the investigation of the data set. Finally, Section V provides conclusions and directions for further research.

II. Related Work

The main aim of the ICGC/TCGA pan-cancer project was to combine data from the separate diseases into one all encompasing data set that consists of multiple tumor types [12]. The goal was to find the commonalities and differences across various tumor types by analyzing and interpreting the data. Up to now there have been a few machine learning and deep learning methods, which have been applied to the analysis of pan-cancer data. We will further outline a few of the recent studies that are related to the pan-cancer analysis.

One approach used machine learning to build a reliable classification model that recognizes 33 types of cancer patients [13]. Five machine learning algorithms were applied (decision tree, k-nearest neighbor, linear support vector machine, polynomial support vector machine, and artificial neural network) in order to analyze the pan-cancer data. The results reported show that linear SVM is the best classifier with an accuracy of 95.8%.

In [14], a new method for the classification of multiple tumor types is proposed. The method uses a relaxed Lasso selection feature subsets and an improved support vector machine (GenSVM) classifier. GenSVM is compared with the three classifiers, namely KNN, L1logreg, L2logreg, on a four multi-label data sets. The results from the experiment showed that GenSVM performs better in terms of generality, flexibility, and classification accuracy.

Another approach looked at the pan-cancer atlas to recognize 9,096 TCGA tumor samples representing 31 tumor types [15]. The researchers applied k-nearest neighbors (KNN) to classify the 31 different tumor types, and additionally employed a genetic algorithm to improve the accuracy of the KNN classifier. This approach achieved an accuracy of around 90% for all 31 tumor types.

Deep learning has been applied to many medical classification problems in order to identify cancer types. Another solution used a stacked auto-encoder to extract high-level features from the expression values in [16]. Afterwards, the features were fed into a single layer artificial neural network to classify whether a tumor is present or not. The results obtained was a 94% accuracy. However, a shortcoming of this work is that the authors only investigated breast cancer given the more complicated network structure and parameter settings needed for the neural network approach and also to save processing time and cost.

In [17], an optimized deep learning approach was introduced based on binary particle swarm optimization with decision tree (BPSO-DT) as well as using a Convolutional Neural network (CNN) to classify different types of tumor. The authors analyzed five different tumor types (KIRC, BRCA, LUSC, LUAD, and UCEC). The experiments conducted showed an accuracy of 96.6% was achieved with this approach.

A new method was designed in [18] which used the high dimensional RNA-Seq data and converted it into 2-D images and afterwards applied a CNN for the classification of the 33 tumor types. This approach achieved an accuracy

of 95.59% for all tumor types. However, as was pointed out by the authors, their proposed approach did not achieve good classification performance on the tumor data sets with small samples and thus runs into the overfitting problem.

III. Methods

This section introduces the techniques used for this investigation namely feature selection, oversampling technique, normalization, and the machine learning techniques applied (neural networks, decision trees, support vector machine, gaussian naïve bayes, random forest, logistic regression, K-nearest neighbor, and gradient boosting).

A. Data Set

The data set investigated in our research is a TCGA Pan-cancer HiSeq data set. This data set is composed of RNA sequencing values from samples that correspond to five cancer types: 300 samples of breast invasive carcinoma (BRCA), 78 samples of colon adenocarcinoma (COAD), 146 samples of kidney renal clear cell carcinoma (KIRC), 141 samples of lung adenocarcinoma (LUAD), and 136 samples of PRAD (Prostate adenocarcinoma). These cancer categories were encoded numerically: 0 = BRCA, 1 = COAD, 2 = KIRC, 3 = LUAD, 4 = PRAD. The data set contains 801 samples and 20,531 numeric attributes.

B. Oversampling (SMOTE)

Since the TCGA Pan-cancer HiSeq data set is imbalanced, Synthetic Minority Oversampling Technique (SMOTE) [19] was applied to balance it. SMOTE is an approach that increases the amounts of samples for minority classes. Its effect is to identify similar but more specific regions in the feature space as the decision region for the minority class [19]. In our research, the sample number of all cancer types was adjusted to 300, which increased the whole sample size from 801 to 1,500.

C. Feature Selection

Feature selection is a process of keeping relevant features and removing irrelevant or redundant ones to gain a subset of features that more accurately describes a given problem [13]. Univariate feature selection is one of the extraction techniques which provides a ranking of features determined by setting a cutoff threshold or limits a certain number of attributes to retain [13]. The balanced data set went through univariate feature selection that discards all features but not a certain percentage of top features that have a strong relationship with cancer types. To determine which percentile of top attributes works the best for the classification algorithms, six different parameters were used and compared for feature selection: top 100% (20,531 features), 50% (10,265 features), 25% (5,133 features), 10% (2,053 features), 5% (1,027 features), 1% (206 features), and 0.1% (21 features). There are in general two reasons why feature selection is used:

1) Reducing the number of features reduces overfitting and improves the generalization of models.
2) To gain a better understanding of the features and their relationship to the response variables.

Univariate feature selection examines each feature individually to determine the strength of the relationship of the feature with the response (dependent) variable.

D. Normalization

Data normalization is known as an essential preprocessing operation which transforms or rescale numeric attributes into a common range of values to minimize the biased contribution of greater numeric features in discriminating patterns of a data set. Normalization deals with the presence of dominant features and outliers, two main components that hamper the learning process of machine learning algorithms [14]. After oversampling and feature selection, normalization was implemented to our RNA-seq data set so that it rescaled the values in all numeric attributes of the data set to values between 0 to 1. After oversampling and feature selection, normalization was implemented for our RNA-seq data set so that it rescaled the values for all numeric attributes of the data set to values ranging between 0 to 1.

E. Neural Network

In computing, Artificial neural networks (neural networks) are created by mimicking the human brain. Deep neural networks are a type of neural network that contain three main components. The input layer, multiple hidden layers, and an output layer [20]. The input layer is the first step in the process, and its number of neurons is determined by the number of samples in the data set. Signals are then sent to the hidden layers, where information is broken down. Determining the size of the hidden layers is largely experimental [21]. Finally, the output layer determines the classification. Therefore, our data set, having 5 different types of cancer to classify samples to, results in an output layer with 5 neurons.

F. Decision Tree

The decision tree classifier is a relatively simple classifier. The name is derived from how the tree data structure is often used, that is, to subdivide data from the root node through yes-no like decisions until a leaf node is reached. For our purposes, the leaf node is the type of classification. Decision trees are popular because it is easy to view how the algorithm determines the classification, and its wide variety of applications [22]. Information gain was determined using entropy, meaning that children were created based upon features that lowered unpredictability. Additionally, the Gini impurity was used to measure the quality of a split.

G. Support Vector Machine

A support vector machine (SVM) is used solely in supervised learning. The separating hyperplane, maximum-margin hyperplane, soft margin, and kernel function are four key functions that help create an SVM [23]. Essentially, samples are plotted, resulting in clusters, the separating hyperplane serves as a divider and any points on either side of the

hyperplane would be classified according to which side they are on. Expanding on this divider is the maximum-margin hyperplane, which alters the hyperplane, so it is, on average, the furthest from the sample points of the different classes. Meanwhile, the soft margin allows points that end up on the wrong side of the divider, within a certain distance, to be correctly classified. Finally, kernel functions can introduce dimensionality to the data and hyperplane in the case that a single hyperplane would not work with.

H. Gaussian Naïve Bayes

Naïve Bayes classifiers are unique because they assume no relation between features, which is seldom the case in experiments where data is gathered. Success of the algorithm is surprising, and its success may be a result of how features either frequently support a classification or cancel the other features out [24]. The Gaussian Naïve Bayes classifier is an alteration of the Naïve Bayes that maintains its core principle, but it finds the mean and standard deviation of the features of the classification types. Therefore, if a sample has an approximately same mean and standard deviation as a classification type, then that sample is most likely of that type.

I. Random Forest

At its core, a random forest is a collection of decision trees. However, unlike decision trees, the trees in random forests are generated from randomly picking data from the data set and creating a tree based upon that specific data (not the whole data set as a decision tree would). After the forest is done being grown, each tree casts a vote for what classification type it thinks the sample is. The sample is then classified as the classification type with most votes. Overfitting occurs when the algorithm learns a data set to the point where it is unable to generalize to other, similar, data. One benefit of the random forest classifier is that it is immune to overfitting because of the law of large numbers [25]. The number of trees that populate the forest can be altered as a parameter, for our experiment we used the default value of 100 trees.

J. Logistic Regression

Despite its name containing the word regression, logistic regression is a binary classifier. For the logistic regression algorithm to be compatible with multi-class classification, one-vs-all classification must be applied. Essentially, one-vs-all creates separate N number of binary classifiers corresponding to N types of classes. The algorithm operates using the logistic function (also called the sigmoid function), which is an s-shaped curve containing a non-negative derivative at every point. Additionally, the logistic function is defined for numbers from 0 to 1 making it suitable to characterize probabilities [26]. When classifying, the sigmoid function can therefore determine which class a sample is because the samples class would have a high probability compared to other classes with a low probability based upon the same features.

K. K-Nearest Neighbor

K-nearest-neighbor (KNN) classification is often used in unsupervised learning as the output variables of a data set are not provided, and the clustering component of the algorithm works well with no output variables. KNN can also be utilized as a supervised learning model. The KNN algorithm is also popular because of its rather simplistic and intuitive design. In a supervised learning situation, the KNN algorithm creates a model by comparing a new sample with its k-number of nearest samples, hence the name k-nearest neighbor [27] because the neighbors of a sample are crucial to determine its classification. Therefore, its is important to restrain the number of neighbors used to determine a classification to a fixed amount as those neighbors closest would likely lead to the correct result. For our experiments we set the $n\_neighbors$ parameter to the default value of 5.

L. Gradient Boosting

The gradient boosting decision tree (GBDT) classifier [28] is like the random forest classifier because it uses a simpler classifier; in our experiments we utilize the decision tree classifier for both algorithms, and then it alters or expands the simpler classifier. However, unlike the random forest classifier, the GBDT does not create a forest of trees. The GBDT only instantiates a single tree, it then modifies the tree with every training iteration. Improving and correcting the mistakes of the previous tree. There are several important parameters that the GBDT can take in such as $n\_estimators$, $max\ depth$, and $learning\ rate$. The $n\_estimators$ parameter controls how many iterations of algorithm will be ran when creating a model. The $max\ depth$ limits the number of nodes that will be in the tree. These two parameters' values vary with separate experiments. While the learning rate determines how much the current tree will impact the next iterations tree. For all experiments the default value of 0.1 is used.

IV. Experiments and Results

The experiments were run as follows. First, the original data set without oversampling and feature selection was used and all different classifiers were applied. This is then compared to the data set with feature selection and oversampling applied. Afterwards, different percentages of features selected are experimented with. For all experiments five-fold cross-validation was used.

A. Experiment 1: Original Data Set versus Oversampled Data Set

Table I shows the results of the original data set used without oversampling and feature selection. In order to compare the results, Table II shows the results obtained by using the data set after oversampling has been applied. As can be clearly seen from these two tables, oversampling has a beneficial effect on the improvement of the classification results. All results for accuracy, AUC, precision, recall, and F-1 score are improved. The largest differences in terms of accuracy are achieved by the Naïve Bayes classifier with values of 0.8027465668 versus

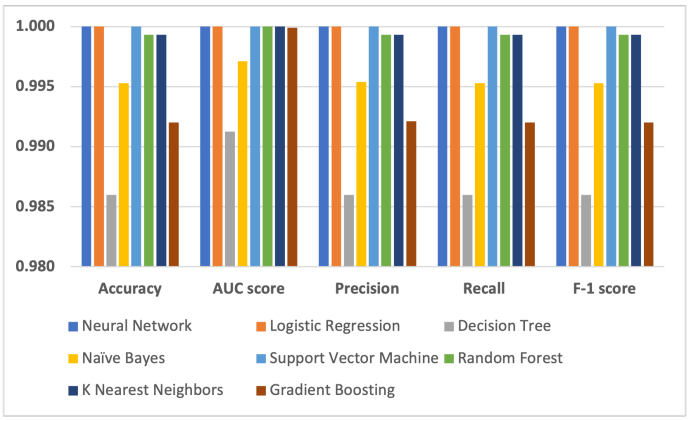

Fig. 1. Overall Results of the 5% Feature Selection Run

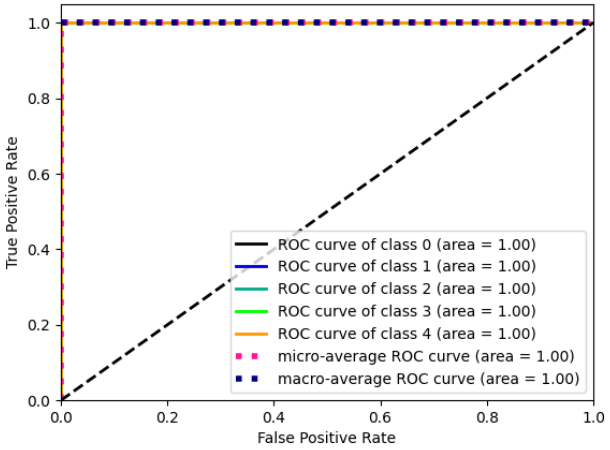

Fig. 2. ROC Curve of K-Nearest Neighbor

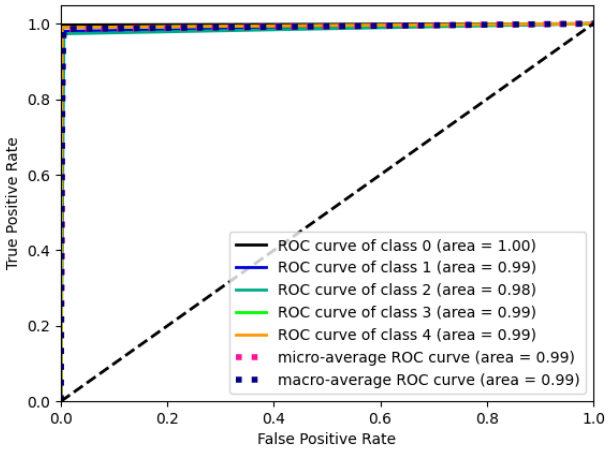

Fig. 3. ROC Curve of Decision Tree

0.9440000000. The same goes for the AUC score, where Naïve Bayes achieved 0.9649888889 versus 0.8353669495.

B. Experiment 2: Various Percentages of Features Used

This set of experiments were conducted in order to find the best number as well as the most important features of the data set for the classification. Tables III-VIII show the results of 50%, 25%, 10%, 5%, 1%, and 0.1%, respectively. The tables list the accuracy, AUC, precision, recall, and F-1 score applying all classifiers. As can be seen, the best classification results are achieved when 5% of the features are used for the classification task. Figure 1 shows the overall results in a graphical format for the results obtained using 5% of the features.

C. Experiment 3: Comparing Different ML Classifiers

In this section, we evaluated the different ML classifiers. We could have listed all the AUC ROC curves, however, we would like to show two particular examples to highlight the differences between K-Nearest Neighbor and Decision Tree. The ROC plots for K-Nearest Neighbor and Decision Tree are shown in Figure 2 and 3, respectively. We can observe a perfect ROC curve for K-Nearest Neighbor and an almost perfect ROC curve for Decision Tree.

Table IX shows the training times of the different classifiers. As can be seen from the table, the most time-consuming classifier is Gradient Boosting with 1214.48 seconds, whereas the fastest algorithm is K-Nearest Neighbor with a training time of 1.06 seconds.

V. Conclusions

In this paper, we applied supervised machine learning algorithms to analyze and compare their capability applied to cancer classification. In particular, the TCGA Pan-cancer HiSeq data set was investigated. The RNA sequencing data set consists of five separate cancer types such as breast invasive carcinoma, colon adenocarcinoma, kidney renal clear cell carcinoma, lung adenocarcinoma, and Prostate adenocarcinoma. The data set was preprocessed with feature selection, oversampling, and normalization techniques. The goal was to determine which algorithm generates a classification model that shows the best performance when categorizing cancer types using accuracy, precision, recall, AUC score, F-1 score, and processing time as evaluation measures.

The data set was investigated by applying eight classifiers with feature selection, including the top 100% (original data), 50%, 25%, 10%, 5%, 1%, and 0.1% of features. The results demonstrated that most of the classifiers achieved their highest accuracy using the top 5% (= 1,027) features. All classification models achieved nearly 99% or 100% accuracy and none of their scores went below 98%.

Among the eight models, the K-Nearest Neighbor classifier is the best algorithm for predicting cancer types due to its notably higher accuracy and time efficiency. It is noteworthy that K-Nearest Neighbor, which is considered a simple and most straightforward classifier, achieved as high as or higher accuracy than the other more sophisticated algorithms and

TABLE I
DATA - OVERSAMPLING AND NO FEATURE SELECTION

Classifier	Accuracy	AUC score	Precision	Recall	F-1 score
Neural Network	1.0000000000	1.0000000000	1.0000000000	1.0000000000	1.0000000000
Logistic Regression	0.9980000000	1.0000000000	1.0000000000	1.0000000000	1.0000000000
Decision Tree	0.9866666667	0.9916666667	0.9900000000	0.9900000000	0.9900000000
Naïve Bayes	0.9440000000	0.9649888889	0.9500000000	0.9400000000	0.9400000000
Support Vector Machine	1.0000000000	1.0000000000	1.0000000000	1.0000000000	1.0000000000
Random Forest	0.9993333333	1.0000000000	1.0000000000	1.0000000000	1.0000000000
K-Nearest Neighbors	1.0000000000	1.0000000000	1.0000000000	1.0000000000	1.0000000000
Gradient Boosting	0.9926666667	0.9990777778	0.9900000000	0.9900000000	0.9900000000

TABLE II
ORIGINAL DATA - NO OVERSAMPLING AND NO FEATURE SELECTION

Classifier	Accuracy	AUC score	Precision	Recall	F-1 score
Neural Network	0.9975031211	0.9999776786	1.0000000000	1.0000000000	1.0000000000
Logistic Regression	0.9513108614	0.9997723214	0.9800000000	0.9200000000	0.9400000000
Decision Tree	0.9712858926	0.9807700465	0.9700000000	0.9700000000	0.9700000000
Naïve Bayes	0.8027465668	0.8353669495	0.8400000000	0.7400000000	0.7600000000
Support Vector Machine	0.9962546816	0.9999776786	1.0000000000	0.9900000000	1.0000000000
Random Forest	0.9975031211	1.0000000000	1.0000000000	1.0000000000	1.0000000000
K-Nearest Neighbors	0.9975031211	0.9999715209	1.0000000000	1.0000000000	1.0000000000
Gradient Boosting	0.9737827715	0.9989271307	0.9800000000	0.9700000000	0.9700000000

TABLE III
FEATURE SELECTION: 50% WITH SMOTE

Classifier	Accuracy	AUC score	Precision	Recall	F-1 score
Neural Network	0.9993333333	0.9999944444	1.0000000000	1.0000000000	1.0000000000
Logistic Regression	0.9993333333	1.0000000000	1.0000000000	1.0000000000	1.0000000000
Decision Tree	0.9840000000	0.9900000000	0.9800000000	0.9800000000	0.9800000000
Naïve Bayes	0.9773333333	0.9861611111	0.9800000000	0.9800000000	0.9800000000
Support Vector Machine	0.9993333333	0.9999972222	1.0000000000	1.0000000000	1.0000000000
Random Forest	0.9986666667	1.0000000000	1.0000000000	1.0000000000	1.0000000000
K-Nearest Neighbors	1.0000000000	1.0000000000	1.0000000000	1.0000000000	1.0000000000
Gradient Boosting	0.9940000000	0.9999777778	0.9900000000	0.9900000000	0.9900000000

TABLE IV
FEATURE SELECTION: 25% WITH SMOTE

Classifier	Accuracy	AUC score	Precision	Recall	F-1 score
Neural Network	0.9993333333	1.0000000000	1.0000000000	1.0000000000	1.0000000000
Logistic Regression	0.9986666667	1.0000000000	1.0000000000	1.0000000000	1.0000000000
Decision Tree	0.9880000000	0.9925000000	0.9900000000	0.9900000000	0.9900000000
Naïve Bayes	0.9846666667	0.9907430556	0.9800000000	0.9800000000	0.9800000000
Support Vector Machine	1.0000000000	1.0000000000	1.0000000000	1.0000000000	1.0000000000
Random Forest	0.9986666667	1.0000000000	1.0000000000	1.0000000000	1.0000000000
K-Nearest Neighbors	1.0000000000	1.0000000000	1.0000000000	1.0000000000	1.0000000000
Gradient Boosting	0.9913333333	0.9990722222	0.9900000000	0.9900000000	0.9900000000

TABLE V
FEATURE SELECTION: 10% WITH SMOTE

Classifier	Accuracy	AUC score	Precision	Recall	F-1 score
Neural Network	1.0000000000	1.0000000000	0.9993000000	0.9993000000	0.9993000000
Logistic Regression	0.9993333333	1.0000000000	0.9993000000	0.9993000000	0.9993000000
Decision Tree	0.9860000000	0.9912500000	0.9887000000	0.9887000000	0.9887000000
Naïve Bayes	0.9946666667	0.9966638889	0.9915000000	0.9913000000	0.9914000000
Support Vector Machine	0.9993333333	1.0000000000	0.9993000000	0.9993000000	0.9993000000
Random Forest	0.9980000000	0.9999972222	0.9987000000	0.9987000000	0.9987000000
K-Nearest Neighbors	0.9993333333	1.0000000000	0.9993000000	0.9993000000	0.9993000000
Gradient Boosting	0.9913333333	0.9945833333	0.9921000000	0.9920000000	0.9920000000

TABLE VI
FEATURE SELECTION: 5% WITH SMOTE

Classifier	Accuracy	AUC score	Precision	Recall	F-1 score
Neural Network	1.0000000000	1.0000000000	1.0000000000	1.0000000000	1.0000000000
Logistic Regression	1.0000000000	1.0000000000	1.0000000000	1.0000000000	1.0000000000
Decision Tree	0.9860000000	0.9912500000	0.9860000000	0.9860000000	0.9860000000
Naïve Bayes	0.9953000000	0.9971000000	0.9954000000	0.9953000000	0.9953000000
Support Vector Machine	1.0000000000	1.0000000000	1.0000000000	1.0000000000	1.0000000000
Random Forest	0.9993000000	1.0000000000	0.9993000000	0.9993000000	0.9993000000
K-Nearest Neighbors	0.9993000000	1.0000000000	0.9993000000	0.9993000000	0.9993000000
Gradient Boosting	0.9920000000	0.9999000000	0.9921000000	0.9920000000	0.9920000000

TABLE VII
FEATURE SELECTION: 1% WITH SMOTE

Classifier	Accuracy	AUC score	Precision	Recall	F-1 score
Neural Network	0.9993000000	0.9999000000	0.9993000000	0.9993000000	0.9993000000
Logistic Regression	0.9993000000	0.9999000000	0.9993000000	0.9993000000	0.9993000000
Decision Tree	0.9807000000	0.9879000000	0.9808000000	0.9807000000	0.9807000000
Naïve Bayes	0.9987000000	0.9992000000	0.9987000000	0.9987000000	0.9987000000
Support Vector Machine	0.9993000000	1.0000000000	0.9993000000	0.9993000000	0.9993000000
Random Forest	0.9966000000	0.9999000000	0.9967000000	0.9967000000	0.9967000000
K-Nearest Neighbors	0.9993000000	0.9996000000	0.9993000000	0.9993000000	0.9993000000
Gradient Boosting	0.9906000000	0.9999000000	0.9907000000	0.9907000000	0.9907000000

TABLE VIII
FEATURE SELECTION: 0.1% WITH SMOTE

Classifier	Accuracy	AUC score	Precision	Recall	F-1 score
Neural Network	0.9927000000	0.9997000000	0.9927000000	0.9927000000	0.9927000000
Logistic Regression	0.9867000000	0.9999000000	0.9868000000	0.9867000000	0.9866000000
Decision Tree	0.9867000000	0.9917000000	0.9866000000	0.9867000000	0.9866000000
Naïve Bayes	0.9880000000	0.9978000000	0.9884000000	0.9880000000	0.9881000000
Support Vector Machine	0.9947000000	0.9999000000	0.9947000000	0.9947000000	0.9947000000
Random Forest	0.9920000000	0.9999000000	0.9921000000	0.9920000000	0.9920000000
K-Nearest Neighbors	0.9927000000	0.9970000000	0.9927000000	0.9927000000	0.9927000000
Gradient Boosting	0.9900000000	0.9992000000	0.9900000000	0.9900000000	0.9900000000

TABLE IX
TRAINING TIME IN SECONDS

Classifier	Time
Neural Network	79.05057096
Logistic Regression	6.98290682
Decision Tree	8.83977962
Naïve Bayes	1.47989392
Support Vector Machine	17.65491033
Random Forest	9.86569095
K-Nearest Neighbors	1.05722189
Gradient Boosting	1214.48126125

produced the second fastest model.

REFERENCES

[1] Y. H. Zhang et al., Identifying and analyzing different cancer subtypes using RNA-seq data of blood platelets, Oncotarget, vol. 8, no. 50, pp. 87494-87511, Oct 2017.

[2] N. E. M. Khalifa, M. H. N. Taha, D. Ezzat Ali, A. Slowik, and A. E. Hassanien, Artificial Intelligence Technique for Gene Expression by Tumor RNA-Seq Data: A Novel Optimized Deep Learning Approach, IEEE Access, vol. 8, pp. 22874-22883, 2020.

[3] J. Ferlay, M. Ervik, F. Lam, M. Colombet, L. Mery, M. Piñeros, et al. Global Cancer Observatory: Cancer Today. Lyon: International Agency for Research on Cancer; 2020 https://gco.iarc.fr/today, [Accessed: July 2021].

[4] S. H. Hassanpour and M. Dehghani, Review of cancer from perspective of molecular, Journal of Cancer Research and Practice, vol. 4, no. 4, pp. 127-129, Dec 2017.

[5] M. Monobe and R. Silva, Gene Expression: An Overview of Methods and Applications for Cancer Research, Veterinaria e Zootecnia, vol. 23, p. 532, Sep 2016.

[6] A. Bashiri, M. Ghazisaeedi, R. Safdari, L. Shahmoradi, and H. Ehtesham, Improving the Prediction of Survival in Cancer Patients by Using Machine Learning Techniques: Experience of Gene Expression Data: A Narrative Review, Iran J Public Health, vol. 46, no. 2, pp. 165-172, Feb 2017.

[7] C. Hutter and J. C. Zenklusen, The Cancer Genome Atlas: Creating Lasting Value beyond Its Data, Cell, vol. 173, no. 2, pp. 283-285, Apr. 2018, doi: 10.1016/j.cell.2018.03.042.

[8] J. N. Weinstein, The Cancer Genome Atlas Pan-Cancer analysis project, nature genetics, vol. 45, no. 10, p. 8, 2013.

[9] S. Sah, Machine Learning: A Review of Learning Types, Mathematics & Computer Science, preprint, Jul 2020.

[10] I. H. Sarker, Machine Learning: Algorithms, Real-World Applications and Research Directions, SN COMPUT. SCI., vol. 2, no. 3, p. 160, May 2021.

[11] Q. Liu and Y. Wu, Supervised Learning, in Encyclopedia of the Sciences of Learning, N. M. Seel, Ed. Boston, MA: Springer US, 2012, pp. 3243-3245. Jan 2012.

[12] J. N. Weinstein, E. A. Collisson, G. B. Mills, K. R. M. Shaw, B. A. Ozenberger, K. Ellrott, et al. The cancer genome atlas pan-cancer analysis project. Nat. Genet. 45:1113, 2013. doi: 10.1038/ng.2764.

[13] Y.-H. Hsu, and D. Si. Cancer type prediction and classification based on rna-sequencing data, in 2018 40th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC) (Honolulu, HI: IEEE), 5374-5377, 2018.
[14] C. Kang, Y. Huo, L. Xin, B. Tian, and B. Yu. Feature selection and tumor classification for microarray data using relaxed lasso and generalized multi-class support vector machine. J. Theoret. Biol. 463, 77-91, 2019. doi: 10.1016/j.jtbi.2018.12.010.
[15] Y. Li, K. Kang, J. M. Krahn, N. Croutwater, K. Lee, D. M. Umbach, et al. A comprehensive genomic pan-cancer classification using the cancer genome atlas gene expression data. BMC Genomics 18:508, 2017. doi: 10.1186/s12864-017-3906-0.
[16] P. Danaee, R. Ghaeini, and D. A. Hendrix. A deep learning approach for cancer detection and relevant gene identification, in Pacific Symposium on Biocomputing 2017 (Hawaii: World Scientific), 219-229, 2017.
[17] N. E. M. Khalifa, M. H. N. Taha, D. E. Ali, A. Slowik, and A. E. Hassanien. Artificial intelligence technique for gene expression by tumor rna-seq data: a novel optimized deep learning approach. IEEE Access 8, 22874-22883, 2020. doi: 10.1109/ACCESS.2020.2970210.
[18] B. Lyu, and A. 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 (Washington, DC), 89-96, 2018.
[19] N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer, SMOTE: Synthetic Minority Over-sampling Technique, Journal of Artificial Intelligence Research, vol. 16, pp. 321-357, Jun 2002.
[20] W. Liu, W. Zidong, L. Xiaohui, Z. Nianyin, L. Yurong and F. E. Alsaadi, A survey of deep neural network architectures and their applications, Neurocomputing, vol. 234, pp. 11-26, 2017.
[21] T. Vujicic, M. Tripo, L. Jelena, B. Adis and S. Zoran, Comparative analysis of methods for determining number of hidden neurons in artificial neural network, in In Central European conference on information and intelligent systems, 2016.
[22] A. Navada, A. Aamir Nizam, P. Siddharth and S. Balwant A., Overview of use of decision tree algorithms in machine learning, In 2011 IEEE control and system graduate research colloquium, pp. 37-42, 2011.
[23] W. S. Noble, What is a support vector machine?, Nature biotechnology, vol. 24, no. 12, pp. 1565-1567, 2006.
[24] H. Zhang, The optimality of Naïve Bayes, Proceedings of FLAIRS2004 conference, vol. 1, no. 2, pp. 3, 2004.
[25] L. Breiman, Random forests, Machine learning, vol. 45, no. 1, pp. 5-32, 2001.
[26] D. G. Kleinbaum, K. Dietz, M. Gail and M. Klein, Logistic Regression, New York: Springer-Verlag, 2002.
[27] D. T. Larose, k-Nearest Neighbor Algorithm, in Discovering Knowledge in Data An Introduction to Data Mining, Wiley Interscience, 2005, pp. 90-106.
[28] J. Friedman, Greedy function approximation: a gradient boosting machine, The Annals of Statistics, vol. 29, no. 5, pp. 1189-1232, Oct 2001.