Journal of Software Engineering and Applications, 2023, 16, 155-169
https://www.scirp.org/journal/jsea
ISSN Online: 1945-3124
ISSN Print: 1945-3116

Research and Implementation of Cancer Gene Data Classification Based on Deep Learning

Yuanzhou Wei[1], Meiyan Gao[1*], Jun Xiao[1*], Chixu Liu[2*], Yuanhao Tian[3*], Ya He[4*]

[1]College of Engineering and Computing, Florida International University, Miami, USA
[2]College of Intelligent Equipment, Shandong University of Science and Technology, Qingdao, China
[3]Steven J. Green School of International & Public Affairs, Florida International University, Miami, USA
[4]School of Economics, Capital University of Economics and Business, Beijing, China
Email: ywei011@fiu.edu, mgao010@fiu.edu, jxiao008@fiu.edu, chixuliu03@163.com, ytian020@fiu.ed, heyaheya97@163.com

How to cite this paper: Wei, Y.Z., Gao, M.Y., Xiao, J., Liu, C.X., Tian, Y.H. and He, Y. (2023) Research and Implementation of Cancer Gene Data Classification Based on Deep Learning. *Journal of Software Engineering and Applications*, **16**, 155-169.
https://doi.org/10.4236/jsea.2023.166009

Received: May 6, 2023
Accepted: June 25, 2023
Published: June 28, 2023

Abstract

Cancer has become a cause of concern in recent years. Cancer genomics is currently a key research direction in the fields of genetic biology and biomedicine. This paper analyzes 5 different types of cancer genes, such as breast, kidney, colon, lung and prostate through machine learning methods, with the goal of building a robust classification model to identify each type of cancer, which will allow us to identify each type of cancer early, thereby reducing mortality.

Keywords

Cancer, Healthcare, SVM, Random Forest, Neural Network, Deep Learning

1. Introduction

Cancer genomics is currently a key research direction in the fields of genetic biology and biomedicine. Research on the pathogenesis of cancer can effectively guide the formulation of reasonable cancer treatment plans and the development of cancer drugs [1]. On the other hand, due to the development of high-throughput gene sequencing instruments, the cost of gene sequencing has been greatly reduced, and cancer treatment has gradually developed towards personalized treatment. Predicting cancer incidence based on gene sequencing data is very important for early diagnosis and treatment of cancer. This paper proposes a neural network-based cancer classification model, studying 802 people who were detected with different types of cancer, each sample contains more than 20K gene

*These authors contributed equally to this work.

expression values, and the ultimate goal is to build a robust classification model to identify each type of cancer, and we processed the collected cancer data by Kafka [2]. This will allow us to identify each type of cancer early, thereby reducing mortality.

2. Proposed Methodology

In this section, we present proposed work including exploratory analysis of cancer gene data, dimensionality reduction, clustering, and classification methods using machine learning and neural networks, such as Decision tree classifier, SVM, Random Forest, Naive Bayes Classifier, Deep Neural Network's classification of cancer gene data and other stages, our final task is to build a powerful classification model to identify each type of cancer. Figure 1 shows the research path of this study.

Figure 1. Research path.

2.1. Exploratory Data Analysis

The data set consists of two parts, one is the patient number, and the type of cancer it belongs to. There are 801 patients in total, 5 different types of cancer, and the other set is the cancer genome data corresponding to each number. Each genome data contains 20531 genes, our first step is to merge these two sets of data to get a merged data set, and the follow-up work will be carried out on this merged data set. The merged data is shown in Table 1.

After the merged data, we check the merged data structure and the distribution of the merged data, and plot the merged dataset as a hierarchically clustered heatmap (Figure 2). There are 5 different types of cancer distributed in 801 samples in our dataset (Figure 3).

Table 1. Merged data.

Class	gene_0	gene_1	gene_10	gene_100	gene_1000	gene_10000	...	gene_9999
BRCA	0.011362	2.839739	0.544066	10.68149	10.30357	3.258028	...	6.954733
COAD	0.022212	3.438381	0.357278	11.01575	9.951124	3.462039	...	6.618466
KIRC	0.046544	2.398129	1.166824	10.239	11.14809	1.651798	...	6.429343
LUAD	0.041088	3.35826	0.607541	10.51767	10.5037	3.754181	...	6.429343
PRAD	0.026544	3.441041	0.765608	10.28294	9.967433	1.949878	...	7.104225

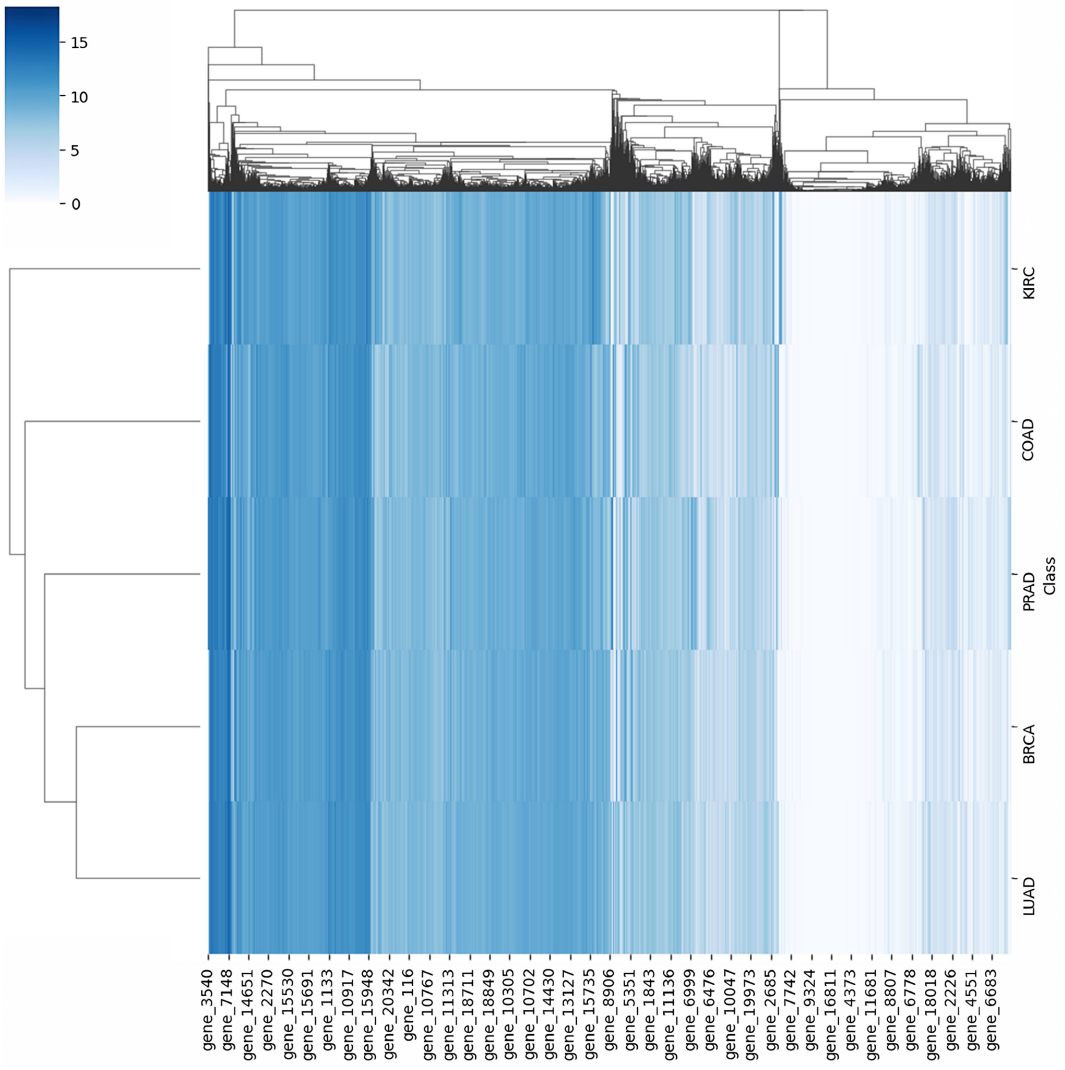

Figure 2. Clustered heatmap.

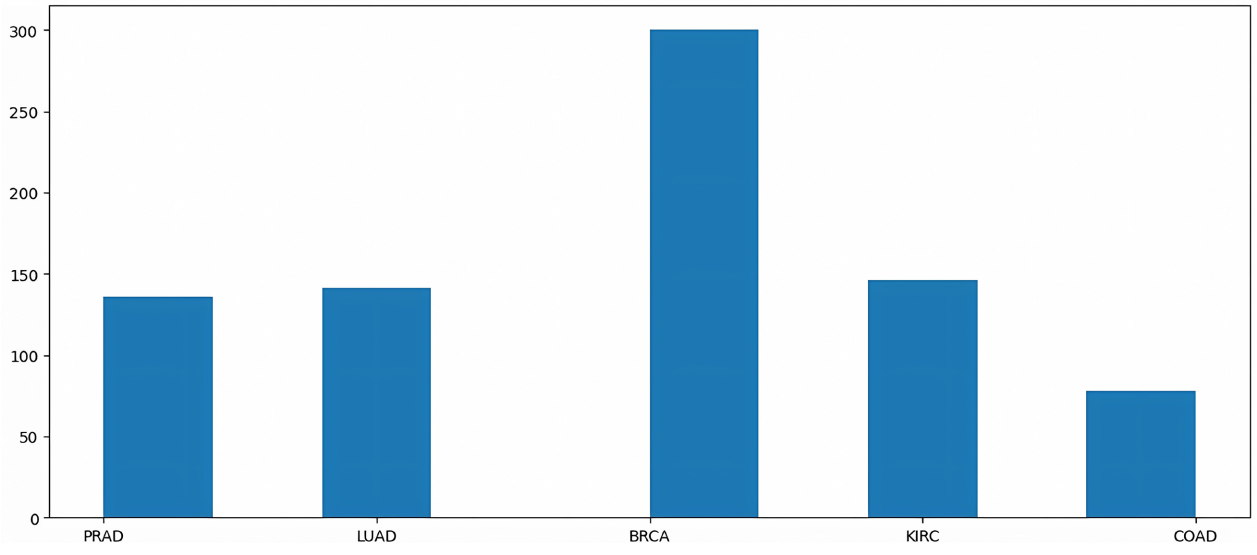

Figure 3. Cancer distribution.

2.2. Dimensionality Reduction Method

In data mining and machine learning, data is represented as vectors. There are many situations in machine learning that deal with tens of thousands or even hundreds of thousands of dimensions. In this case, the resource consumption of machine learning is unacceptable, so we must reduce the dimensionality of the data. Although dimensionality reduction certainly means the loss of information, in view of the correlation that often exists in the actual data itself, we can find ways to reduce the loss of information as much as possible while reducing dimensionality. In the research data object of this paper, each sample has expression values for around 20K genes. However, it may not be necessary to include all 20K gene expression values to analyze each cancer type. Therefore, we will identify a smaller set of attributes which will then be used to fit multiclass classification models. Therefore, the first goal of this paper is to use PCA, LDA and t-SNE for dimensionality reduction.

1) *PCA dimensionality reduction*

Principal component analysis (PCA) is a transformation technique widely used in unsupervised linear data, mainly for dimensionality reduction. Widely used in gene expression level data analysis in the field of bioinformatics. The purpose of PCA is to find the direction of maximum variance from high-dimensional data and map the data to a new feature subspace whose dimension is not larger than the original data. PCA transforms the original data into a set of linearly independent representations of each dimension through linear transformation, which can be used to extract the main feature components of the data and is often used for dimensionality reduction of high-dimensional data. The essence of PCA is to take the direction with the largest variance as the main feature, and "decorrelate" the data in each orthogonal direction, that is, to make them have no correlation in different orthogonal directions [3].

The algorithm steps of PCA:

- There are m pieces of n-dimensional data.
- Form the original data into a matrix X with n rows and m columns by column.
- Zero-meanize each row of X, that is, subtract the mean value of this row.
- Find the covariance matrix.
- Find the eigenvalues and corresponding eigenvectors of the covariance matrix.
- Arrange the eigenvectors into a matrix from top to bottom according to the size of the corresponding eigenvalues, and take the first k rows to form a matrix P.
- Y = PX is the data after dimension reduction to k dimension.

The PCA algorithm can increase the sampling density of samples by discarding part of the information, thereby alleviating the disaster of dimensionality. When the data is affected by noise, the eigenvectors corresponding to the smallest eigenvalues are often related to noise and discarding them can play a role to a certain extent [4]. To the effect of noise reduction, the main information is retained,

but this main information is only for the training set, not necessarily important information, it may discard some seemingly useless, but it happens to be important information, so PCA may also exacerbate overfitting. PCA not only compresses the data to low dimension, but it also makes the features of the reduced data independent of each other.

We want to preserve an exact amount of variance in the data after applying PCA, so specify a float between 0 and 1 for the hyperparameter n_components, we choose 0.995. Figure 4 and Figure 5 respectively show the performance of PCA with different n_components.

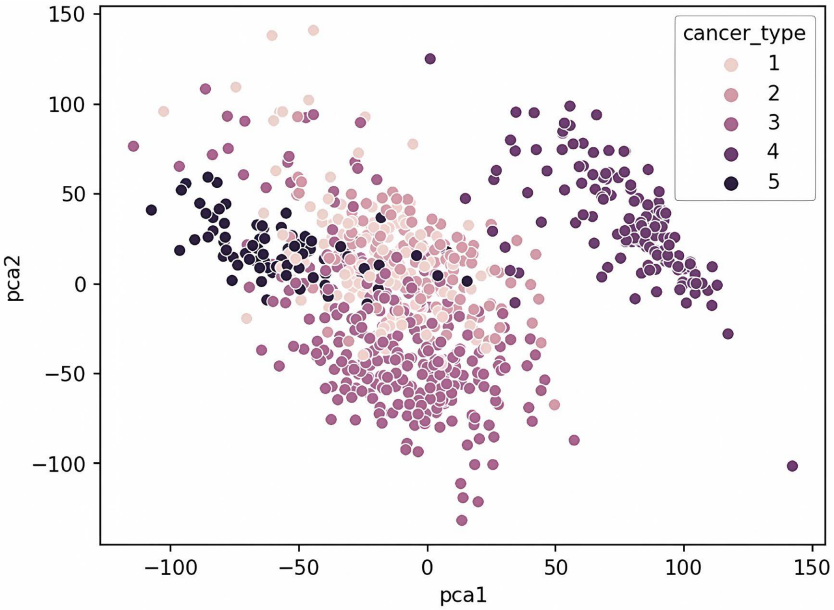

Figure 4. Perform PCA with n_components = 2.

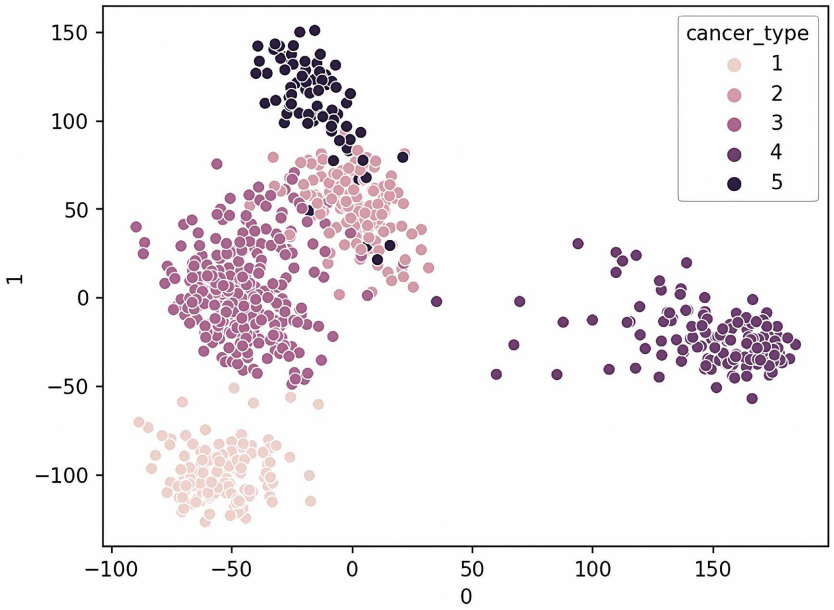

Figure 5. Perform PCA with n_components = 0.995.

2) *LDA dimensionality reduction*

Different from the PCA variance maximization theory, the idea of the LDA algorithm is to project the data into a low-dimensional space, so that the same type of data is as compact as possible, and the different types of data are as scattered as possible [5]. Therefore, the LDA algorithm is a supervised machine learning algorithm. At the same time, LDA has the following two assumptions: a) The original data is classified according to the sample mean. b) Data of different classes have the same covariance matrix. In practical situations, it is impossible to satisfy the above two assumptions. But when the data is mainly distinguished by the mean, LDA can generally achieve good results (**Figure 6**).

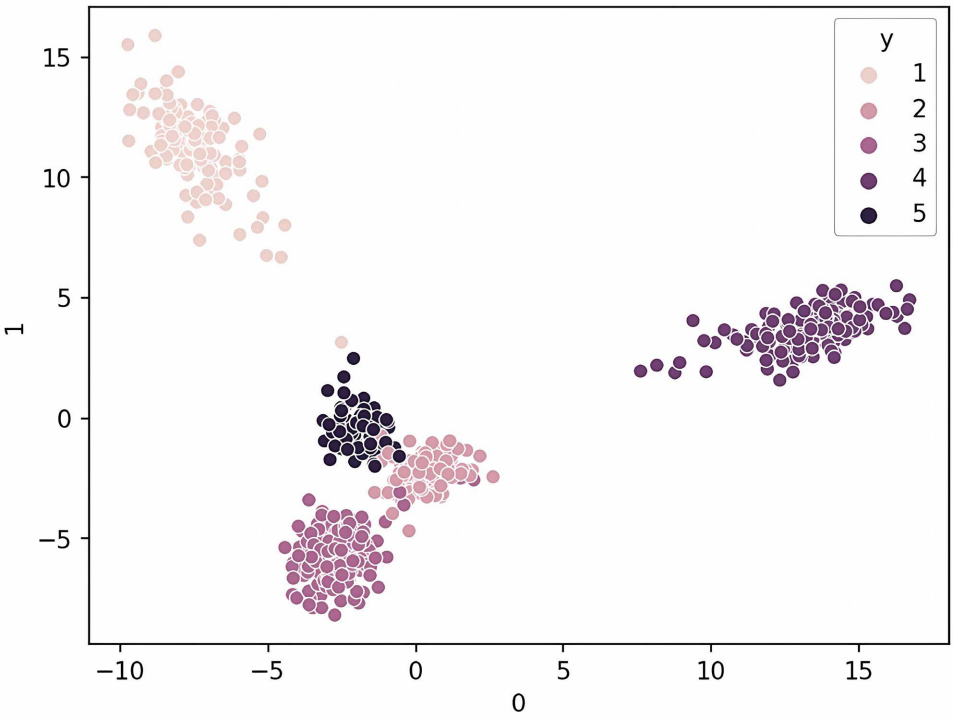

Figure 6. Dimensionality reduction using LDA.

3) *t-SNE dimensionality reduction*

The main use of t-SNE is to visualize and explore high-dimensional data. The main goal of t-SNE is to transform a multi-dimensional dataset into a low-dimensional dataset. Creates a probability distribution by selecting a random data point and calculating the Euclidean distance to other data points, more similarity values will be obtained from data points near the selected data point, and farther away from the selected data point the data points will get less similarity values, using the similarity values, a similarity matrix will be created for each data point. Unlike PCA, t-SNE can be better applied to both linear and nonlinear well-clustered datasets and produces more meaningful clusters. While t-SNE is excellent at visualizing well-separated clusters, most of the time it fails to preserve the overall geometry of the data [6] (**Figure 7**).

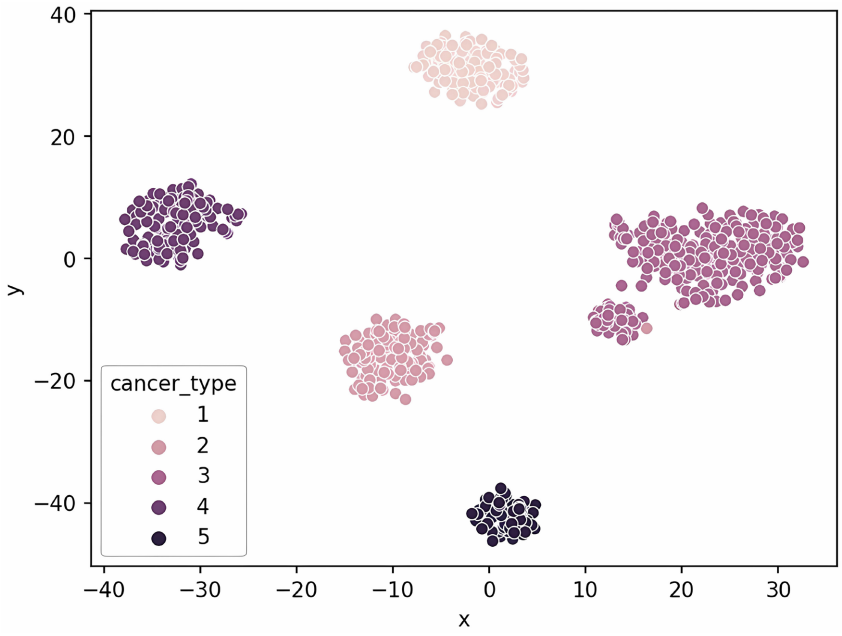

Figure 7. Dimensionality reduction using t-SNE.

2.3. Clustering Genes and Samples

Our goal is to identify groups of genes that behave similarly across samples and identify the distribution of samples corresponding to each cancer type. Therefore, this task focuses on applying various clustering techniques, e.g. k-means, hierarchical and mean shift clustering, to genes and samples.

- First, apply the given clustering technique on all genes to identify:
- Genes whose expression values are similar across all samples;
- Genes whose expression values are similar across samples of each cancer type.
- Next, apply the given clustering technique to all samples to identify:
- Samples of the same class (cancer type) which also correspond to the same cluster;
- Samples identified to be belonging to another cluster but also to the same class (cancer type).

1) *k-means*

K-means clustering is the most famous partitioning clustering algorithm, and its simplicity and efficiency make it the most widely used of all clustering algorithms. The k-means clustering algorithm (k-means clustering algorithm) is a cluster analysis algorithm for iterative solution. Its steps are to divide the data into K groups in advance, then randomly select K objects as the initial cluster centers, and then calculate the distance between each object and each seed cluster center, assign each object to the cluster center closest to it [7]. The cluster centers and the objects assigned to them represent a cluster. Each time a sample is assigned, the cluster center of the cluster is recalculated based on the existing objects in the cluster. This process will be repeated until a certain termination condition is met. Termination conditions can be that no (or minimum number) objects are reassigned to different clusters, no (or minimum number) cluster

centers change again, and the sum of squared errors is locally minimized [8]. Figure 8 and Figure 9 respectively show the performance of K-means clustering with PCA different n_components.

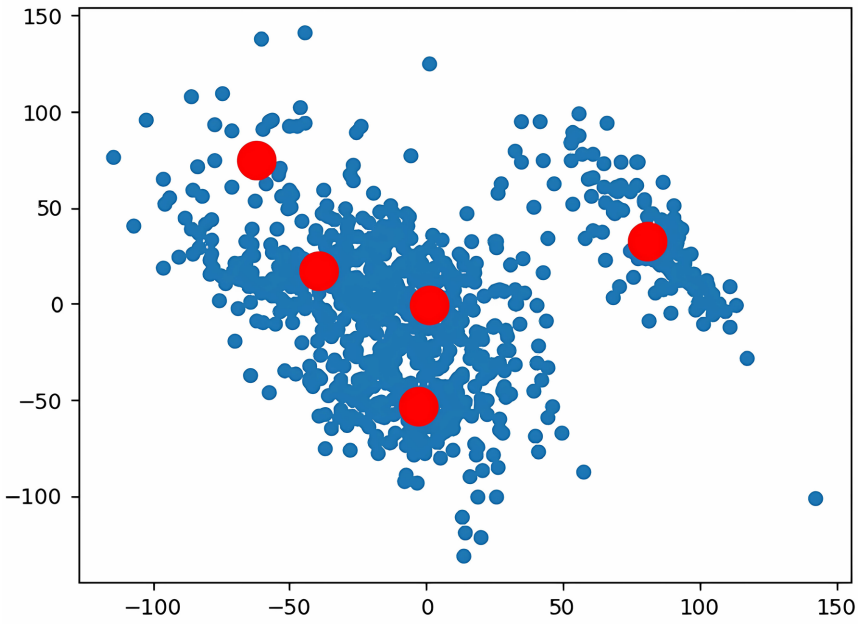

Figure 8. K-means clustering with PCA = 2.

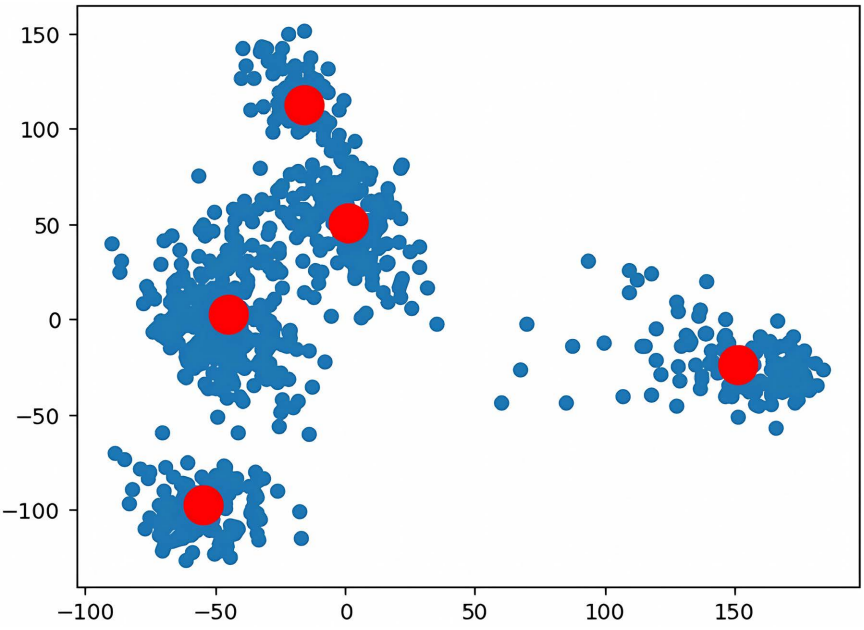

Figure 9. K-means clustering with PCA = 0.995.

2.4. Building Classification Models

Our goal is to identify groups of genes that behave similarly across samples and identify the sample distributions that correspond to each cancer type. Therefore, this task focuses on applying various clustering techniques (e.g. k-means, hierarchic-

al, and mean-shift clustering) to genes and samples. The final task is to build a robust classification model(s) for identifying each type of cancer. Build a classification model(s) using multiclass SVM, Random Forest, and Deep Neural Network to classify the input data into five cancer types.

1) *Decision tree classifier*

Decision tree is a tree built based on strategic choices. In machine learning, decision trees are a predictive model that represents a mapping relationship between object attributes and object values. Each node in the tree represents an object, and each forked path represents a possible attribute value, and the path experienced from the root node to the leaf node corresponds to a decision test sequence. Decision tree can be a binary tree or a non-binary tree, and it can also be regarded as a collection of if-else rules, or as a conditional probability distribution on the feature space [9]. What makes decision trees special in the field of machine learning models is the clarity with which they represent information. The "knowledge" acquired by the decision tree through training directly forms a hierarchical structure. Decision trees have a wide range of applications, can be used for classification and regression, and are very easy to do multi-category classification, and can handle numerical and continuous samples. The result we got using Decision tree classifier with max depth 5 is 0.987.

2) *SVM*

Support vector machines (SVMs) is a binary classification model [10]. Its basic model is a linear classifier with the largest interval defined in the feature space. The largest interval makes it different from the perceptron. The principle of SVM is to maximize the distance from the closest point (support vector) to the hyperplane.

The problem to be optimized by SVM is:

$$\min_{w,b,\beta}\left(\frac{1}{2}|w|^2 + C\sum_{i=1}^{m}\beta_i\right) \text{s.t. } y_i\left(w^{\mathrm{T}}x_i + b\right) \geq 1-\beta_i \ \beta_i \geq 0$$

Using the LaGrange dual problem to transform the problem into:

$$\max_{\alpha} \sum_{i=1}^{m}\alpha_i - \frac{1}{2}\sum_{i,j=1}^{m}\alpha_i\alpha_j y_i y_j x_i x_j$$

$$\text{s.t.} \sum_{i=1}^{m}\alpha_i y_i = 0$$

$$0 \leq \alpha_i \leq C$$

Then, we use the SMO algorithm to solve a series of α and then get the update kernel function of W and b, and replace it with the inner product form of the corresponding kernel function:

$$K\left\langle x_i, x\right\rangle$$

The result we got using SVM is 1.0.

3) *Random forest*

Random forest belongs to the Bagging method in ensemble learning. It is an algorithm that integrates multiple trees through the idea of ensemble learning. Its basic unit is a decision tree, which essentially belongs to a large branch of

machine learning—Ensemble Learning. The result we got using Random Forest is 0.987.

4) *Naive Bayes classifier*

Naive Bayes classifiers are a family of simple probabilistic classifiers based on applying Bayes' theorem under the assumption of strong (naive) independence between features. The classifier model assigns class labels represented by feature values to problem instances, drawn from a finite set. It is not a single algorithm for training such a classifier, but a family of algorithms based on the same principle: all Naive Bayesian classifiers assume that each feature of the sample is uncorrelated with other features [11].

$$P(B|A) = \frac{P(A|B)P(B)}{P(A)}$$

The Naive Bayesian classifier algorithm is simple in logic, easy to implement, and the time and space overhead in the classification process is small. In theory, the Naive Bayesian model has the smallest error rate compared with other classification methods, but it is not always the case in practice. This is because the Naive Bayesian model assumes that the attributes are independent of each other, this assumption is often not true in practical applications. When the number of attributes is large or the correlation between attributes is large, the classification effect is not good. Our result using the Naive Bayes classifier is 0.738.

5) *KNN classifier*

KNN learning (K-Nearest Neighbor algorithm, K-nearest neighbor method) is a statistical classifier. The basic idea is: the input has no label (the category of the labeled data), that is, new data that has not been classified, first extract the features of the new data and compare with each data feature in the test set; then extract the K-nearest neighbor (most similar) data feature labels from the test set, and count the most frequently occurring categories among the K-nearest neighbor data, and use it as a new data category [12]. First, calculate and sort the distance between the features of the data to be classified and the features of the training data, and take out the nearest K training data features; then determine the category of the new sample according to the category of the K similar training data features: if they all belong to one category, then the new sample also belongs to this class; otherwise, score each candidate category and determine the category of the new sample according to a certain rule. The result we got using the KNN classifier is 0.98.

6) *Deep neural network*

Deep neural network (DNN) is a framework for deep learning, which is a neural network with at least one hidden layer. Like shallow neural networks, deep neural networks can also provide modeling for complex nonlinear systems, but the extra layers provide a higher level of abstraction for the model, thus improving the ability of the model. A deep neural network is a discriminative model that can be trained using the backpropagation algorithm [13].

Stochastic gradient descent (often abbreviated SGD) is an iterative method for

optimizing an objective function with suitable smoothness properties. It can be regarded as a stochastic approximation of gradient descent optimization, since it replaces the actual gradient (calculated from the entire data set) by an estimate thereof (calculated from a randomly selected subset of the data) [14]. Especially in high-dimensional optimization problems, this reduces the very high computational burden, achieving faster iterations in trade for a lower convergence rate. The result we got using Neural Network with Stochastic gradient descent is 1.0 (**Figure 10**).

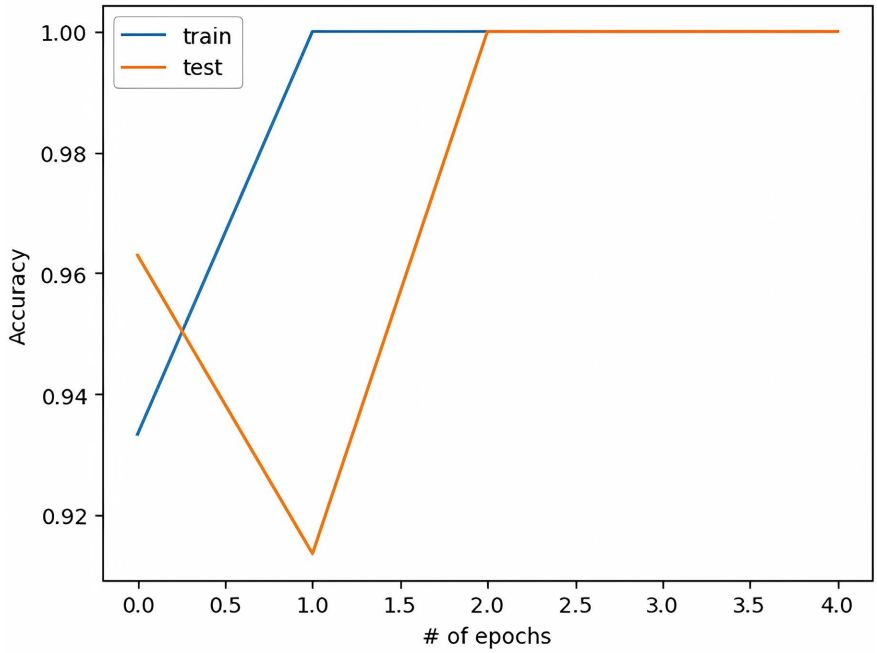

Figure 10. Accuracy plot.

7) *Recursive feature elimination*

Apply the feature selection algorithms, forward selection, and backward elimination to refine selected attributes using the classification model from the previous step. We use RFECV cross-validation to retain the best performing features. Based on REF, cross-validate different feature combinations. The learner itself remains unchanged [15]. By calculating the sum of its decision coefficients, the importance of different features for the score is finally obtained, and then the best feature combination is retained. Our result accuracy: 0.887. After we use Recursive feature elimination, the accuracy drops instead, so it is not recommended to use this method for genomic data.

3. Experimental Setup and Results Analysis

In this section, we present the description of datasets used in our experimentation, performance metrics used in evaluating the performance of the proposed method followed by experimental results and analysis. As a development tool, we have used Anaconda Python 3.5, Keras Deep Learning Library as a front-end

and TensorFlow open-source deep-learning library as the back-end to construct our model.

3.1. Dataset Description

We use the 5 types of cancer data set obtained by ICMR (Indian Council of Medical Research), the data set contains 801 samples, corresponding to 801 people who were detected with different types of cancer, each sample contains the expression values of 20531 genes, the sample has one of the following tumor types: BRCA, KIRC, COAD, LUAD, and PRAD (Table 2).

Table 2. 5 types of cancer.

Class	Full name
PRAD	Prostate adenocarcinoma
LUAD	Lung adenocarcinoma
BRCA	Breast invasive carcinoma
KIRC	Kidney renal clear cell carcinoma
COAD	Colon adenocarcinoma

3.2. Performance Measures

Validate the genes selected from the last step using statistical significance testing F-test.

F-test, the most used alias is called joint hypothesis test (English: joint hypotheses test), also known as variance ratio test, variance homogeneity test. It is a test that the statistical value obeys the F-distribution under the null hypothesis (H0). One-Way Analysis of Variance F-test is mainly used to compare the means of two or more groups to determine that the overall mean is different.

The calculation of the p-value is inseparable from the hypothesis test. The p-value is a decreasing index of the credibility of the result. The larger the p-value, the less we can think that the relationship between the variables in the sample is a reliable indicator of the relationship between the variables in the population [16]. The p-value is the probability of making an error that considers the observation to be valid, that is, generally representative. For example, p = 0.05 reminds that there is a 5% chance that the variable correlation in the sample is caused by chance. In many fields of research, a p-value of 0.05 is often considered the borderline level of acceptable error. We performed and passed the F-test (Table 3).

Table 3. 5 one-way F-test.

	tsne1	tsne2	cancer_type
0	3.307757	31.124784	PRAD
1	−6.858942	−14.268332	LUAD
2	−2.450273	36.27655	PRAD

Continued

3	−2.477862	35.25325	PRAD
4	24.84078	0.097425	BRCA
…	…	…	…
796	24.109064	3.15204	BRCA
797	−13.20399	−21.358574	LUAD
798	−13.20399	−21.358574	LUAD
799	−0.037208	31.34466	PRAD
800	−3.847794	26.823185	PRAD

4. Discussion and Comparative Analysis

This section analyzes in detail the experimental results obtained on the dataset due to the proposed method. As shown in Table 4, score results for different models, we get the result of each classification model. The accuracy of the classification results of SVM and neural network is 1, the effect of random forest and KNN classifier is also good, and the worst is decision tree classifier, indicating that in the field of cancer genome data classification, using SVM and neural network classification will get more accurate results.

Table 4. Score results for different models.

Model	Score
Decision tree classifier	0.954356846
SVM	1.0
Random forest	0.987551867
Naive Bayes classifier	0.738589212
KNN classifier	0.995850622
Neural network	1.0

5. Conclusions

In this paper, we have utilized deep learning algorithms to conduct research on cancer genomic data and develop a classification model. The primary focus of our research includes the following key aspects:

Exploratory data analysis: We performed a comprehensive exploration of the cancer genomic data, aiming to gain insights into the underlying patterns and characteristics of the data.

Dimensionality reduction: To handle the high-dimensional nature of genomic data, we employed dimensionality reduction techniques to extract the most relevant features, enhancing the efficiency and interpretability of our classification model.

Gene and sample clustering: Through clustering analysis, we grouped genes and samples based on their similarities, facilitating a deeper understanding of the

relationships and potential subtypes within the cancer data.

Feature selection and model construction: By employing feature selection methods, we identified the most informative genomic features and utilized them to construct our classification model. We trained a neural network model on the cancer genomic data and compared it with various machine learning classification methods to determine the most accurate algorithm.

The successful classification of common types of cancer genomes using machine learning methods demonstrates the significance of our research. By accurately identifying the potential causes of these cancers based on the genes responsible, we contribute to reducing mortality rates associated with cancer.

However, it is crucial to acknowledge the limitations and challenges that arise in this field of study. Access to high-quality biological samples required for genomic studies, particularly for rare tumor types, can be limited or insufficient. Therefore, further advancements in bioinformatics infrastructure and the collaboration of interdisciplinary teams are necessary to enhance the efficiency and effectiveness of cancer genomic research.

By emphasizing the importance of our research findings and acknowledging the need for enhanced resources and collaboration, we hope to pave the way for future advancements in cancer genomics and its potential applications in improving healthcare outcomes.

Conflicts of Interest

The authors declare no conflicts of interest regarding the publication of this paper.

References

[1] Alberts, B., Bray, D., Hopkin, K., *et al.* (2013) Essential Cell Biology. Garland Science, New York.

[2] Wei, Y.Z., Li, M.M. and Xu, B.S. (2017) Research on Establish an Efficient Log Analysis System with Kafka and Elastic Search. *Journal of Software Engineering and Applications*, **10**, 843-853. https://doi.org/10.4236/jsea.2017.1011047

[3] Abdi, H. and Williams, L.J. (2010) Principal Component Analysis. *Wiley Interdisciplinary Reviews*: *Computational Statistics*, **2**, 433-459. https://doi.org/10.1002/wics.101

[4] Jolliffe, I.T. (2002) Principal Component Analysis. Wiley Online Library, Hoboken.

[5] Newman, D., Asuncion, A., Smyth, P. and Welling, M. (2009) Distributed Algorithms for Topic Models. *Journal of Machine Learning Research*, **10**, 1801-1828.

[6] Kobak, D. and Berens, P. (2021) Understanding Deep Learning through T-SNE. *Journal of Machine Learning Research*, **22**, 1-37.

[7] Celebi, M.E., Kingravi, H.A. and Vela, P.A. (2013) A Comparative Study of Efficient Initialization Methods for the K-Means Clustering Algorithm. *Expert Systems with Applications*, **40**, 200-210. https://doi.org/10.1016/j.eswa.2012.07.021

[8] Kanungo, T., Mount, D.M., Netanyahu, N.S., Piatko, C.D., Silverman, R. and Wu, A.Y. (2002) An Efficient K-Means Clustering Algorithm: Analysis and Implementation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **24**, 881-892.

https://doi.org/10.1109/TPAMI.2002.1017616

[9] Verma, R. and Kumar, V. (2018) Decision Trees for Data Mining: A Review. *Current Trends in Computer Science and Mechanical Automation*, **1**, 1-10.

[10] Wang, H.F., Zheng, B.C., Yoon, S.W. and Ko, H.S. (2018) A Support Vector Machine-Based Ensemble Algorithm for Breast Cancer Diagnosis. *European Journal of Operational Research*, **267**, 687-699. https://doi.org/10.1016/j.ejor.2017.12.001

[11] Rish, I. (2001) An Empirical Study of the Naive Bayes Classifier. *IJCAI* 2001 *Workshop on Empirical Methods in Artificial Intelligence*, Seattle, 4 August 2001, 41-46.

[12] Kumar, M., Rath, N.K., Swain, A. and Rath, S.K. (2015) Feature Selection and Classification of Microarray Data Using MapReduce Based ANOVA and K-Nearest Neighbor. *Procedia Computer Science*, **54**, 301-310. https://doi.org/10.1016/j.procs.2015.06.035

[13] Schmidhuber, J. (2015) Deep Learning in Neural Networks: An Overview. *Neural Networks*, **61**, 85-117. https://doi.org/10.1016/j.neunet.2014.09.003

[14] Kourou, K., Exarchos, T.P., Exarchos, K.P., Karamouzis, M.V. and Fotiadis, D.I. (2015) Machine Learning Applications in Cancer Prognosis and Prediction. *Computational and Structural Biotechnology Journal*, **13**, 8-17. https://doi.org/10.1016/j.csbj.2014.11.005

[15] Guyon, I., Weston, J., Barnhill, S. and Vapnik, V. (2002) Gene Selection for Cancer Classification Using Support Vector Machines. *Machine Learning*, **46**, 389-422. https://doi.org/10.1023/A:1012487302797

[16] Kar, S., Sharma, K.D. and Maitra, M. (2015) Gene Selection from Microarray Gene Expression Data for Classification of Cancer Subgroups Employing PSO and Adaptive K-Nearest Neighborhood Technique. *Expert Systems with Applications*, **42**, 612-627. https://doi.org/10.1016/j.eswa.2014.08.014