Turkish Online Journal of Qualitative Inquiry (TOJQI)
Volume 12, Issue 10, October 2021: 4002-4011

Comparative Analysis of Machine Learning Models for Lung Cancer Prediction Using Radiographic Images

Kamal Kumar
Research Scholar,
Dept. of Computer Science & Engineering
Kamla Nehru Institute of Technology, India
kamalkumar.techedu@gmail.com

Dr. Arvind Kumar Tiwari,
Associate Professor
Dept. of Computer Science & Engineering
Kamla Nehru Institute of Technology, India
arvind.tiwari1980@gmail.com

Abstract

There has been an upsurge in objective to promote and enhancing lung cancer screening techniques as people become more aware of the behaviours and risks associated with lung cancer. Machine learning-based lung cancer prognosis models were proposed to assist doctors for incidental or screen found ambiguous pulmonary nodules. Using these technologies, the variability in tumour classification might be reduced and decision-making could be improved, resulting in fewer benign nodules being followed up on. For the main purpose of lung cancer prediction, we conduct comparative research of important machine learning algorithms in this paper and present statistical proofs that certain algorithms will perform better for radiographic-based detection than others for this purpose. In conjunction with machine learning approaches, pre-trained architecture like VGG provides a good detection for biomarkers, making it a viable tool for illness classification. They have the ability to classify while also reducing the number of false positives accurately.

Introduction

Diagnosing an illness is a difficult task, and many tests on patients are usually required to arrive at a precise diagnosis. This may lead to the employment of analytic gadgets, designed to assist clinicians in making decisions. Early detection cuts down on treatment time and may even save lives. Lung malignant development is one of these disorders, which occurs when cells in lung tissues develop uncontrollably. This tumour can expand beyond the lung by metastasizing into neighbouring tissue or other sections of the body. Lung cancer is caused by long-term tobacco use in the great majority of cases (85%), with 10–15 percent of cases occurring in persons who have never smoked [1]. A combination of genetic factors plus exposure to radon gas, asbestos, secondhand smoke, or other forms of air pollution is frequently responsible for these cases. Chest radiography and computed tomography (CT) scans can reveal lung cancer. Biopsy, which is commonly done under bronchoscopy or CT guidance, confirms the diagnosis. Lung cancer is a one-of-a-kind illness that kills 1.61 million people worldwide each year [2].

Lung malignant development is ranked second in men and tenth in women. When cancer is detected early on, the chances of survival are usually better. As a result, early detection of malignant lung development is critical, with roughly 80% of patients being correctly diagnosed only during the early or intermediate stages of the disease. Machine learning employs scientific algorithms to recognise

patterns in large datasets and iteratively improve this recognisable proof's performance with further data. These algorithms are commonly employed in a variety of areas and applications, including as commercial, security, fund, internet-based life, and misrepresentation discovery, to access various sorts of information obtained continuously and from a variety of sources. Because patient information is frequently unavailable for open study, employing these methodologies to analyse illness outcomes might be difficult[3].

Numerous procedures for diagnosing lung cancer have been identified, including the Chest Radiograph (X-ray), Sputum Cytology, Magnetic Resonance Imaging (MRI) and Computed Tomography (CT). However, they are highly costly and take a long time to complete. This means that the majority of techniques have been developed to diagnose lung cancer in its advancing stages, when the patient's prospects of survival are relatively slim. As a result, another novel method for diagnosing lung cancer sooner was required. Image processing techniques are a high-quality tool for improving the manual analysis of cancer. Numerous medical researchers are analysing sputum cells to detect lung cancer in its early stages, and the majority of contemporary research is based on quantitative data such as the shape, size, or ratio of afflicted cells [4].

This paper highlights detection of malignancy of lung cancer with respect to the healthy lung. This detection is performed through various well known machine learning algorithms such as Naïve Bayes, Support Vector Machine, Decision Tree Classifier, Stochastic Gradient Descent and Random Forest Classifier. The paper also presents a comparative analysis between algorithms for the detection of lung cancer. Also, for feature extraction, we are using the architecture of the pre-trained deep learning algorithm of VGG-19.This allows the model to attain higher levels of accuracy during the prediction and identification of Biomarkers [5].

Literature Review

In this section an introduction has been provided for the previous work performed for detection of malignant abnormality related to cancer. Each work presented have used different machine learning algorithms or have detected various biomarkers for the correct classification of cancerous cells. While each of these algorithms handles data in a unique way and provide different perspective of managing Biomarkers.

Chen et al. (2013) introduced a fuzzy system based on k-NN for diagnosis of Parkinson’s disease (PD)
[6].Furthermore, they performed Principal Component Analysis (PCA) to identify distinctivemarkers on which to base the best F-kNN model. During comparison
they discovered that their proposed method outperformed the SVM methodology.

According to Odajima&Pawlovsky (2014), the accuracyachieved by thekNN approach varies due to the number of neighbours and the degree of information used for categorization [7]. Meanwhile, they illustrated how the maximum and minimum accuracy values varied depending on the size of the classification set and the value of k.

Lynch et al. (2017) used the SEER database to rank lung cancer patients based on survival using classification based on supervised approaches such as decision trees, linear regression, Support Vector Machine, Gaussian Based Model (GBM) and an ensemble method [3]. The results showed that GBM, was the most precise of the five individual models used.

On a heart illness dataset with 303 occurrences and 14 characteristics, Khateeb and Usman (2017) employed J48, k-NN, NB, and other ensemble classifiers/ML classification algorithms [8]. They splittheir data into six groups and found that the kNN classifier was the most accurate. Using all 14 features, it had the greatest accuracy of 79.20 percent. Furthermore, Tayeb et al. (2017) used k-NN to analyse two illnesses with 90% accuracy using datasets gathered by the University of California [9].

Pradeep and Naveen (2018) used Support Vector Machine, Naïve Bayes, and C4.5 on the North Central Cancer Treatment Group (NCCTG) lung cancer data set to assist medical practitioners in making more accurate predictions about cancer survival rates [10].The outcomes suggest that as the training data set grows more prominent, C4.5 gets better at predicting lung cancer. To diagnose lung cancer, Alharbi (2018) used a mixed genetic-fuzzy algorithm. He used the procedure on 32 patients with 56 attributes and achieved 97.5 percent accuracy with a 93 percent confidence level without any reduction in dimensions [11].

To categorise lung lumps as malignant or benign, Lakshmanaprabu et al. (2019) developed a hybrid method combining linear discriminate analysis (LDA) andan optimum deep neural network (ODNN)[12]. When it was first developed, the ODNN extractsimperative indicators from radiographic images. The features were then reduced in dimensionality using LDA. Finally, the ODNN was optimised using a modified gravitational search strategy. Their algorithm's accuracy, specificity, and sensitivity were determined as 96.2 percent, 94.2 percent, and 94.56 percent, respectively.

The proposed approach

The proposed approach revolves around the combination of convolution-based deep neural networks and primary machine learning algorithms. This section provides a background of all such methods and the advantage of using a pre-trained deep neural network (VGG)for feature selection [13]. The feature extracted act as an input to the respective machine learning models that were chosen for comparison.

Feature Extraction

VGG is a pre-trained convolution neural network proposed in 2014 by Karen Simonyan and Andrew Zisserman from Oxford[25]. VGG takes an input RGB image of desired size. The pre-processing layer subtracts the mean image values determined for the complete ImageNet training set from the RGB image with pixel values ranging from 0–255. These weight layers are applied to the pre-processed input images. A series of convolutional layers is applied to the sample images. VGG16 architecture consists of 5 convolutional blocks each having 3x3 convolution and a pooling layer. Rather than utilising massive filters, VGG makes use of smaller (3x3) filters with a deeper level. The perceptron is identical when only one 7 × 7 convolutional layer is used.

The output from the last convolution block is reduced using a global average pooling layer that reduces the tensor size. The received output feature is then fed to a series of fully connected neural layers, reducing the features to the desired size. Finally, each image of the respective class is traversed through the above architecture and features are selected that numerically define the image. The features are labelled and further fed to any one the desired machine learning algorithm for classification.

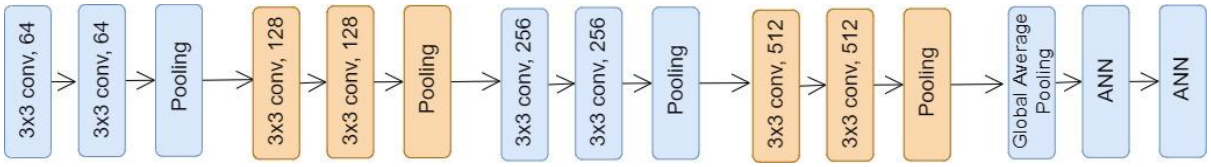

Figure 1: Architecture of Pre-trained VGG Model

This is followed by using the supervised learning algorithms for the correct classification using the features extracted.

Gaussian Naïve Bayes

The Naive Bayes Classifier is a prediction technique using probabilistic models and the Bayes theorem with a high independence condition. It is presumed in this situation that occurance of one event in a cluster doesn't have any impact on the behaviour of any other event of the cluster [14]. In multi-label learning, Training data sets contain several instances of labels related to other labels, so Nave Bayes Classifiers were used to predict the labels from unobserved encounters [15].

One typical method for dealing with feature values in Naive Bayes classification is to use Gaussian distributions to represent the probabilities of the features based on the classes. In gaussian probability distribution is represented using a bell-shaped curve using the formula

$$N(\mu, \sigma^2)(x) = 1 /\sqrt{} 2\pi\sigma^2 * \exp(-(x-\mu)^2/2\sigma^2)$$

Here mean is (μ) and variance is represented σ^2. The parameters required in Naive Bayes are in the order of O(nc), where n denotes the number of attributes and c denotes the classes. A gaussian distribution $P(Xi|C) \sim N(\mu, \sigma^2)$ is created for each continuous property.

Decision Tree Classifier

A decision tree functions in a manner similar to a decision support tool. It makes use of a tree-like graph to represent decisions and their probable outcomes, including resource costs, event results, and efficiency [16]. One of the machine learning algorithms used in the decision tree for classification is ID3.

ID3 is an algorithm for supervised learning. It learns formally using training examples from numerous different classes [17] [18]. It makes predictions about an item's class based on the notion it develops. By using ID3, we're looking for attributes that set one example class apart from another. ID3 necessitates the knowledge of all characteristics ahead of time and the behaviour of each one.

ID3 Algorithm

Step 1: After measuring the entropy $E(a_i)$ for each feature a_i, the attribute with the lowest entropy is chosen.

Step 2: After partitioning the entire object set by the values of the attribute a_i, the relevant sub-nodes are created. If all the items in a sub-node belong to the same class, then it is a terminal sub-node.

Step 3: Thissubnode is not terminal. Select the followinga_j attribute with the smallest entropy H for each node (a_i, a_j).

For attribute a_j, repeat Steps 2 and 3.

Stochastic Gradient Descent

Gradient descent is a method for minimising an objective function $J(\theta)$ parameterized by the parameters of a model $\theta(R^d)$ by updating the parameters in the opposite direction of the gradient of the objective function $\nabla_\theta J(\theta)$ with respect to the parameters. In order to get to a (local) minimum, the learning rate dictates the number of the steps we must take [19].

For each training example, stochastic gradient descent, a form of gradient descent, adjusts these parameters. Batch gradient descent eliminates superfluous calculations for large datasets by recalibrating gradients for comparable cases prior to updating each parameter. SGD eliminates redundancy by doing one update at a time. As a result, it's usually a lot faster than the alternative method. The objective function changes a lot due to SGD's frequent, high-variance updates [20].

$$\theta = \theta - \eta \cdot \nabla_\theta J(\theta; x^{(i)} ; y^{(i)})$$

Random Forest Classifier

Using ensemble-based learning methods, random forest classifiers are a subset of machine learning. They're easy to set up, quick to use, and have had great results across a wide range of industries. The random forest approach builds several "basic" decision trees during the training step and then uses a majority vote (mode) to classify the results [21]. This voting technique has the added benefit of correcting decision trees' tendency to overfit training data, among other things. Random forests use a technique known as bagging to train separate set of information using individual trees during training stage [22].

A consequence of this is that individual ensemble predictions and errors are becoming more or less dissimilar. Averaging these less-correlated trees' predictions yields better outcomes than bagged decision trees when trying to make a prediction.

Support Vector Classifier

This strategy aims to determine the ideal separation amid classes by concentrating on the training examples that fall on the edges of the class disposition, the support vectors, and effectively discarding the remaining training cases. Since a perfect hyperplane has been fitted, the technique is expected to achieve good accuracy even with small training sets. This leads to save in time for the training and development of a classification model. Thus, the SVM methodused for classification is predicated on the concept that discriminating requires only small data for training on class margins[23].

For the most straightforward illustration of the fundamental nature of classification with an SVM, consider the simple situation where there are two linearly separable classes in the q-dimensional space. When working with the training data represented by $\{x_i, yi\}$, $i= 1,\ldots..r$, $yi\in\{1,-1\}$ in the q dimensional space, the goal is to construct an accurate classifier that generalises to new data. Class separation can be achieved using a variety of multidimensional separating structures, such as a line, plane, or hyperplane, however only one of these hyperplanes is projected to generalise well in comparison to the others. Ideally, all class cases should be located on one side of the separating hyperplane, which should be situated so that the distance between the nearest training data points in both classes is as great as possible [24].

Dataset

Researchers at the National Institutes of Health developed a database of radiographic scans for more than 30,000 patients, many of whom had serious lung conditions. Medical researchers treat the patient at the NIH Clinical Center as collaborators in their study, and they willingly engage in clinical studies. Data security was a top priority when compiling this dataset, so all individually identifying information was cleansed before to distribution.

Experimental Results

Several performance metrics are used for the evaluation of the comparative analysis of the above-mentioned machine learning algorithms. Parameters such as TP that represents the correctly classified samples, FP represents the samples wrongly classified, TN represents sample does not match to any of the classes. Finally, FN again is misclassification. Using these parameters several performance metrics can be computed as:

Accruacy: Represents the ratio between truly identified samples with respect to total number of examined samples.

$$\text{Accuracy} = TN + TP/TN+ TP+ FN + FP$$

Precision: It is defined as the ratio between number of correctly predicted positive events with respect to the total predicted positive events.

$$\text{Precision} = TP/FP + TP$$

Recall: It is calculated by dividing the total number of positive predictions by the total number of positives.

$$\text{Recall} = TP / FN + TP$$

F1- Score: Precision and recall are weighted averages. In cases of unequal class distribution, F1-score showed to be superior to accuracy since it considers both false positives and negatives.

$$\text{F1-Score} = 2 \text{ X } (\text{Recall} + \text{Precision})/ (\text{Precision} + \text{Recall})$$

The accuracy reported using various machine learning algorithm is shown in table 1. The highest accuracy was reported by Support Vector Classifier (SVC) at 98.36 while the lowest was reported by Gaussian Naïve based at 92.73. This is due to the fact that Naïve based assumes there exists no dependency between features. Majorly real-world datasets contain features that might be co-dependent hence some features may be treated with a considerably bigger bias than desired.

Methods	Accuracy
Gaussian Naïve Bayes	92.73
Decision Tree Classifier	96.01
Stochastic Gradient Descent	98.36
Support Vector Classifier	98.43
Random Forest Classifier	98.20

Table 1: Accuracy chart for Machine Learning Models

The highest accuracy is posted by SVC since SVCs generally do not suffer from the situation of overfitting and perform effectively when there is a strong sign of a distinction between classes to be

distinguished. Also, it performs effectively and is well-generalized on out-of-sample data. As a result of its superior performance on out of generalisation sample data, SVM demonstrates its speed, as the certain fact states that when a single sample is classified, the kernel function is evaluated and done for every support vector.

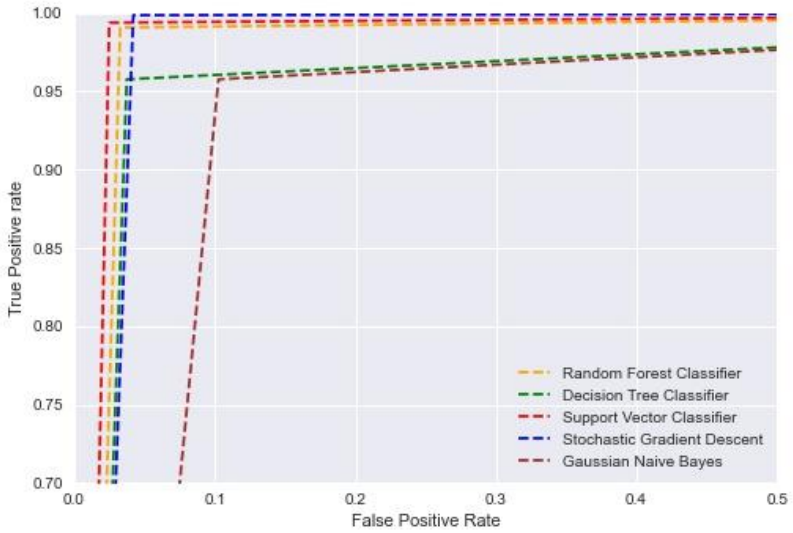

Figure 2. Comparative ROC Curves for Machine Learning Models

A receiver operating characteristic (ROC) curve is used to visualise the accuracy of a classification. The false positive rate (FPR) is plotted on the x-axis, while the True Positive rate (TPR) is plotted on the y-axis in the ROC curve.As a result, a link between FPR and TPR is formed, which can be used to find a threshold for FPR ito achieve optimal TPR. The TPR was reported as 96%, 96.3%, 99.0%, 99.1% and 100% for Gaussian Naïve based, Decision tree classifier, Random Forest Classifier, SGD and SVC at 10% FPR, respectively as shown in Figure 2.

Methods	**Precision**	**Recall**	**F1-Score**
Gaussian Naïve Bayes	0.9288	0.9275	0.9273
Decision Tree Classifier	0.9602	0.9601	0.9602
Stochastic Gradient Descent	0.9839	0.9836	0.9835
Support Vector Classifier	0.9845	0.9844	0.9844
Random Forest Classifier	0.9790	0.9790	0.9789

Table 2. Precision, Recall F1- Score for all Machine Learning Models

To efficiently illustrate the performance of deep learning models, a confusion matrix can be built utilizing True Positives, True Negatives, False Positives, and False Negatives. It is a summary of the classification findings. The confusion matrix makes it easy to see how many correct and wrong predictions there are in each class. It demonstrates the categorization model's ambiguity while making predictions. The x-axis and y-axis, respectively, show the actual and expected class designations. Each square in the confusion matrix represents the likelihood of the projected class matching the actual class. Although there is no positive or negative class in multi-class problems, the number of TP, TN, FP, and FN cannot be computed immediately. These figures can be calculated separately for each

class. The TP for class '0', for example, will be the number of right predictions for that class. The number of TN for class '1' is the sum of probabilities in all squares of the confusion matrix that does not belong to class '1' in either predicted or actual label. In Fig. 3. Two separate confusion matrices are shown, Fig. 3a shows the classifier with the highest wrong matches i.e. Gaussian Naïve Bayes, while Fig. 3b. shows Classifier with 16 false positives for the class "0" while only 4 false positives for class "1"

For symmetric databases, where false positives (FP) and false negatives (FN) have the same cost, accuracy is a helpful performance parameter. However, higher accuracy models are not always preferable. As a result, other performance indicators, including as precision, recall, and F1-score, are studied. The results of all machine learning methods for performance criteria such as precision, recall, and F1-score are reported in Table 2.

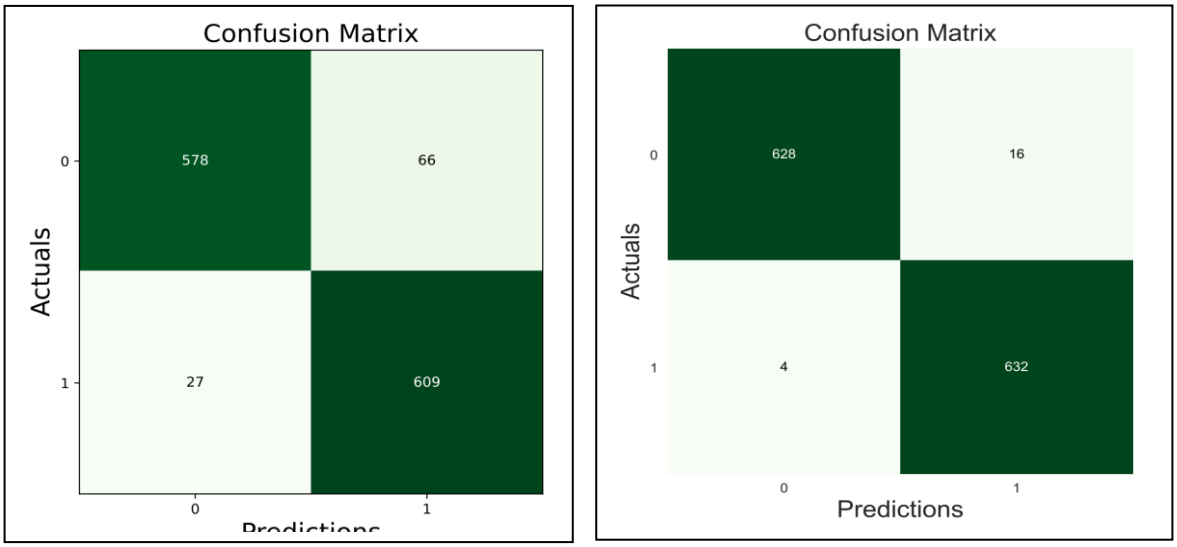

Figure 3. a) Confusion Matrix for GaussianFigure 3. b) Confusion Matrix for SGD Naïvebayes

Conclusion

Lung diseases are illnesses that damage the lungsthat leads to various issues including breathing. This ailment is fairly common in India, and it affects people from all walks of life. The effort intends to detect and classify such lung disorders utilising deep learning approaches for feature extraction and classification using a variety of machine learning front runner algorithms.Among different types of classification algorithms, Support Vector Classifier achieved the best accuracy 98.42%, while the lowest accuracy of 92.73 was reported by Gaussian Naive Bayes Algorithm. The features were extracted by the state-of-the-Art VGG19 pre-trained architecture. A total of 200 features were selected for each of the input sample that acted as input for the machine learning algorithm.

The encouraging results mentioned thus far suggest three possible paths for future research. To begin, the system's performance and variability might be investigated for tiny differences in tumour volume segmentation. Additionally, we can make use of recent state-of-the-art algorithms. In future study, pre-trained deep learning algorithms for more prominent feature extraction and the use of Artificial Neural Networks (ANN) for classification may offer improved results.

References

1. Thun, M. J., Hannan, L. M., Adams-Campbell, L. L., Boffetta, P., Buring, J. E., Feskanich, D., … Samet, J. M. (2008). Lung cancer occurrence in never-smokers: An analysis of 13 cohorts and 22 cancer registry studies. PLoS Medicine, 5(9), 1357–1371.
2. Li, J., Usevich, K., & Comon, P. (2018). Globally convergent Jacobi-type algorithms forsimultaneous orthogonal symmetric tensor diagonalization. SIAM Journal on Matrix Analysis Applications, 39(1), 1–22.
3. Lynch, C. M., Abdollahi, B., Fuqua, J. D., de Carlo, A. R., Bartholomai, J. A., Balgemann, R. N., … Frieboes, H. B. (2017). Prediction of lung cancer patient survival via supervised machine learning classification techniques. International Journal of Medical Informatics, 108, 1–8.
4. Sujitha, R., Seenivasagam, V. Classification of lung cancer stages with machine learning over big data healthcare framework. *J Ambient Intell Human Comput* **12,** 5639–5649 (2021).
5. Hoseok I, Je-Yoel Cho, Chapter Three - Lung Cancer Biomarkers, Advances in Clinical Chemistry, Elsevier, Vol 72, 2015, pp. 107-170.
6. Chen, H.-L., Huang, C.-C., Yu, X.-G., Xu, X., Sun, X., Wang, G., & Wang, S.-J. (2013). An efficient diagnosis system for detection of Parkinson's disease using fuzzy k-nearest neighbor approach. Expert systems with applications, 40(1), 263–271.
7. Odajima, K., Pawlovsky, A. P. (2014). A detailed description of the use of the KNN method for breast cancer diagnosis. In Proceedings of the 7th international conference on biomedical engineering and informatics (BMEI), Dalian, China.
8. Khateeb, N., Usman, M. (2017). Efficient heart disease prediction system using K-nearestneighbor classification technique. In Proceedings of BDIOT2017 proceedings of the international conference on big data and internet of thing, London, UK (pp. 21–26)
9. Tayeb, S., Pirouz, M., Sun, J., Hall, K., Chang, A., Li, J., … Sager, T. (2017). Toward predicting medical conditions using k-nearest neighbors. Proceedings of 2017 IEEE International Conference on Big Data (Big Data), Boston, MA, USA.
10. Pradeep, K. R., & Naveen, N. C. (2018). Lung cancer survivability prediction based on performance using classification techniques of support vector machines, C4. 5 and naive Bayes algorithms for healthcare analytics. Procedia Computer Science, 132, 412–420.
11. Alharbi, A. (2018). An automated computer system based on genetic algorithm and fuzzy systems for lung cancer diagnosis. International Journal of Nonlinear Sciences and Numerical Simulation, 19(6), 583–594.
12. Lakshmanaprabu, S. K., Mohanty, S. N., Shankar, K., Arunkumar, N., & Ramirez, G. (2019). Optimal deep learning model for classification of lung cancer on CT images. Future Generation Computer Systems, 92, 374–382.
13. K. Simonyan and A. Zisserman, " Very deep convolutional networks for large-scale image recognition," in Proc. Int. Conf. Learn. Represent., 2015.
14. G. Qiang, "An Effective Algorithm for Improving the Performance of Naive Bayes for Text Classification," *2010 Second International Conference on Computer Research and Development*, 2010, pp. 699-701, doi: 10.1109/ICCRD.2010.160.
15. U. Widodo Wijayanto and R. Sarno, "An Experimental Study of Supervised Sentiment Analysis Using Gaussian Naïve Bayes," *2018 International Seminar on Application for Technology of Information and Communication*, 2018, pp. 476-481.
16. R. C. Barros, M. P. Basgalupp, A. C. P. L. F. de Carvalho and A. A. Freitas, "A Survey of Evolutionary Algorithms for Decision-Tree Induction," in *IEEE Transactions on Systems,*

Man, and Cybernetics, Part C (Applications and Reviews), vol. 42, no. 3, pp. 291-312, May 2012.

17. H. Zhang and R. Zhou, "The analysis and optimization of decision tree based on ID3 algorithm," *2017 9th International Conference on Modelling, Identification and Control (ICMIC)*, 2017, pp. 924-928
18. Shuo Yang, Jing-Zhi Guo, Jun-Wei Jin, An improved Id3 algorithm for medical data classification, Computers & Electrical Engineering, Vol. 65, 2018, pp. 474-487.
19. S. Bonnabel, "Stochastic Gradient Descent on Riemannian Manifolds," in *IEEE Transactions on Automatic Control*, vol. 58, no. 9, pp. 2217-2229, Sept. 2013.
20. Y. Qiao, B. P. F. Lelieveldt and M. Staring, "An Efficient Preconditioner for Stochastic Gradient Descent Optimization of Image Registration," in *IEEE Transactions on Medical Imaging*, vol. 38, no. 10, pp. 2314-2325, Oct. 2019
21. Md. ZahangirAlam, M. Saifur Rahman, M. Sohel Rahman, A Random Forest based predictor for medical data classification using feature ranking, Informatics in Medicine Unlocked, Vol. 15, 2019.
22. Zhaozhao Xu, Derong Shen, TiezhengNie, Yue Kou, A hybrid sampling algorithm combining M-SMOTE and ENN based on Random forest for medical imbalanced data, Journal of Biomedical Informatics, Vol. 107, 2020.
23. Ocak, H. A Medical Decision Support System Based on Support Vector Machines and the Genetic Algorithm for the Evaluation of Fetal Well-Being. *J Med Syst* **37,** 9913 (2013).
24. Liming Shen, Huiling Chen, Zhe Yu, Wenchang Kang, Bingyu Zhang, Huaizhong Li, Bo Yang, Dayou Liu, Evolving support vector machines using fruit fly optimization for medical data classification, Knowledge-Based Systems, Volume 96, 2016, pp. 61-75.
25. T. Kaur and T. K. Gandhi, "Automated Brain Image Classification Based on VGG-16 and Transfer Learning," *2019 International Conference on Information Technology (ICIT)*, 2019, pp. 94-98.