A Comparative analysis study of lung cancer detection and relapse prediction using XGBoost classifier

Rana Dhia’a Abdu-aljabar¹ and Osama A. Awad²

¹²College of Information Engineering, Al-Nahrain University, Baghdad, Iraq

¹E-mail: ranadhia1@gmail.com

Abstract. Lung cancer is the leading cancer for causing death for both men and women. It also has one of the lowest survival rates in five-year of all cancer types. It remains a challenge to lung cancer relapse prediction after surgery, especially for non-small cell lung cancer (NSCLC). This study aimed to enhance prediction and detection using eXtreme Gradient Boosting (XGBoost) model to detect lung cancer diagnoses and predict its relapse after surgery by using gene expression and its transcriptome changes due to cancer. This can aid to enhance early tumour progression handling and reducing the painful treatment. In this study, it used real New Generation RNA_seq (NGS) and microarray gene expression datasets for different types of lung cancer. The results demonstrated the effectiveness of the XGBoost model compared to other machine learning models especially in handling unbalance datasets.

1. Introduction
Lung cancer is the fourth most common diagnosed cancer and the second common cause of cancer-related death worldwide. Despite development in cancer treatment after surgically resected non-small cell lung cancer (NSCLC), the survival rate is still around 50% for 5-year. And 20% of patients showed recurrence within 5 years even if they were at stage I. Thus, the identification of patients with poor prognosis after surgery is of considerable clinical relevance [1]. Accounting for about 40% of lung cancers is adenocarcinoma, which is subtypes of (NSCLC). It tends widely to form metastases at an early stage, and the response to radiation therapy is not as effective as it is in small cell lung carcinoma [2]. The reason why lung cancer is so deadly is that the tumor is hard to diagnosis in its early stages. A long time it may take to grow and no symptoms are usually there at that stage. By the time that the symptoms be noticed, the other parts of the body may infect by cancer. Therefore, it appeared urgently needing identification improvement especially for the patients who have a high-risk to infected by cancer [3].

The gene expression profile technologies development, such as microarray has become the standard gene expression profiling technology for use in toxicogenomic [4-6]. In recent years, a further opportunity provided by the next generation sequencing (NGS). It comprehensively characterizes the molecular features of cancer.

Many studies interest in improving the prediction of lung cancer, some of them have proposed gene expression-based signatures in patients with lung cancer for survival stratification [3]. Recently, researchers try to develop the detection and predictive methods based on gene expression datasets to classify lung cancer patients with distinct clinical outcomes, including relapse, infection stages and overall survival. Previous studies have used different machine learning algorithms for their power
tools and have an excellent result in different fields, like Russul A. et al.[7-12] has multi-study proposed different new optimization models such as Gene Expression programming (GEP) and Deep Gene Selection (DGS) to improve NSCLC prognosis using microarray datasets, one of them is a hybrid method to rank and select relevant prognostic genes that are related to NSCLC recurrence prediction also Hasseeb A. et al.[13-16] has variant contribution on multi class using GEP algorithm for better classification lung cancer to assists in determining the specific treatment to reduce the fatality rates. While Guobin Z. et al. [17] used a convolutional neural network with a gradient boosting machine (GBM) algorithm based on hybrid features for classification of lung nodules as benign or malignant from CT images. Also, Shulong Li et al. [18] used a deep convolutional neural network to predict lung nodule malignancy with low dose CT (LDCT) lung cancer screening. Patra R. [19] study, has classified available lung data sample in the UCI machine learning repository into benign and malignant using different machine learning classifiers techniques. Yu-Heng Lai et al. [20] study, it developed a deep neural network (DNN) combining heterogeneous data sources of gene expression and clinical data. It predicted the 5-year survival status of NSCLC patients, based on microarray data. In this study, it applied XGBoost algorithm to different real lung cancer dataset types to have more accurate results in detection and relapse prediction and made a comparison with other machine learning algorithms. The results showed that XGBoost has more accurate and confident values especially for unbalance distributed classes in the datasets.

2. XGBoost (Extreme Gradient Boosting)

2.1. Definition

XGBoost is a scalable machine learning system for tree boosting. The impact of the system has been widely recognized in several machine learning and data mining challenges [21]. The most important factor behind the success of XGBoost is its scalability in all scenarios and its fast learning because of distributed and parallel computing which enables quicker model exploration. This lets the researchers and data scientists to build powerful variants of tree boosting algorithms. Recently, it has gained much attention and popularity, as it becomes the algorithm of choice for many winning teams of machine learning competitions. A general schematic diagram in Figure 1, illustrating an ensemble of depth-3 trees [22]. Using gradient-boosting, the trees are built greedily to minimize a regularized objective on the training loss. In XGBoost algorithm is an implementation of a parallelized gradient boosting algorithm.

![Figure 1. Schematic diagram of a boosted ensemble of decision trees [22].](image-url)
2.2 Using variance datasets with XGBoost
There are different types of datasets depend on the way used in gene expression representing. For example, microarray [23], and New generation Sequencing (NGS) [24] datasets. An Integration of clinical and gene expression datasets are usually used in machine learning models to learn on a specific point. This kind of datasets have different sizes (samples), so it needs a machine learning type that can handle that dataset’s differences in type and size. XGBoost can learn with a small number of samples and have good accuracy unlike other types of machine learning like the deep neural networks that need a big dataset size to learn.

3. Lung cancer datasets

3.1 Datasets information
In this study, it applied XGBoost classification on six datasets from two types of gene expressions (Microarray and new generation sequence (NGS)) to predict the lung cancer tumor diagnosis and the probability of relapse after the surgery. All datasets are taken from the National Center for Biotechnology Information site (NCBI). The first one (GSE30219) in microarray datasets which is part of the Cartes d’Identite des Tumeurs (CIT) program from the French Ligue Nationale Contre le Cancer, which analyzed 293 lung tumor samples and 14 non-tumoral lung samples [25]. The second one is (GSE19188), which is gene expression data for early-stage NSCLC. The analysis was performed on a cohort of 156 persons. The results have 91 tumor samples and 65 normal lung tissue samples [26]. The third one is (GSE31210) which it’s a gene expression data for pathological stage I-II lung adenocarcinomas for 226 patient’s lung adenocarcinomas with their clinical information about the patients’ relapse state after surgery [27]. The fourth dataset is (GSE8894) which it is a gene expression of non-small cell lung cancer type for 138 patients and its relapse state after surgery [28]. The fifth dataset is (GSE68465) which it is a gene expression for 442 patients; after remove the patients with missing information; it becomes 362 patients have full information with their clinical reference of relapse state [18]. The last dataset is (GSE81089) [29]. This dataset includes 218 expression profiling by high throughput sequencing; which is called a new generation sequence (NGS); for 199 patients diagnosed with NSCLC and 19 normal lung tissues. This kind of sequence (NGS) provides a gene expression with more precise and sensitive measurement levels than microarrays in the analysis of many samples [30].

| Datasets            | Type                                      | patients | Features | The Class                | Sample distribution (Cancer/Relapse) |
|---------------------|-------------------------------------------|----------|----------|--------------------------|---------------------------------------|
| GSE30219            | Microarray                                | 307      | 54675    | Cancer/Normal            | Yes: 293, No: 14                      |
| GSE19188            | Microarray                                | 156      | 54546    | Cancer/Normal            | Yes: 91, No: 65                       |
| GSE8894             | Microarray                                | 135      | 54675    | Relapse/No Relapse       | Yes: 67, No: 68                       |
| GSE68219            | Microarray                                | 362      | 22283    | Relapse/No Relapse       | Yes: 205, No: 157                     |
| GSE81089            | New Generation Sequencing (NGS)           | 218      | 63129    | Cancer/Normal            | Yes: 199, No: 19                      |

3.2 Data pre-processing
It’s a very important step in any machine learning algorithm. Data gathering methods are often loosely controlled, leading to having unreasonable, missing and out of range data. So, it needs to have cleaned the datasets at first by removing irrelevant, incomplete and unwanted data. In lung cancer datasets there is a lot of missing information especially in clinical information. To overcome this problem,
these records are considered irrelevant records and need to be deleted because they cause unsuitable learning results.

In classification, the decision class needs to be in numeric format. Meanwhile, in some datasets, the format is still in nominal format. It has to be changed from nominal format to numeric format. In this study, all classes are in nominal form, for example, Cancer /Normal or Relapse/No Relapse. Therefore, there is a need to change them to numeric format (0 and 1).

Finally, before it applies the machine learning algorithm (XGBoost), it must split each dataset into two parts; the training set and testing set. In this study, it splits the datasets to 70% of its record for the training set and the rest for the testing set.

4. XGBoost Algorithm
Study the gene expression datasets is an interdisciplinary field in which biology, mathematics and computer science are combined closely together, not only to understand the biological pathways but also to discover the hidden genetic variants causing many incurable diseases nowadays, such as cancer. Cancer detection and relapse prediction from the gene expression data continue to face a challenge due to the high complexity and dimensionality of these data. After decades of research, there is still uncertainty in the clinical information of cancer and the identification of tumor-specific markers.

4.1. the Loos function of XGBoost
In prediction problems involving unstructured data (text, images, etc.) artificial neural networks tend to outperform all other frameworks or algorithms. However, right now, decision tree-based algorithms are considered best-in-class when it deals with small-to-medium structured/tabular data. XGBoost algorithm was developed at the University of Washington as a research project by Tianqi Chen and Carlos Guestrin presented their paper at SIGKDD Conference in 2016 [31]. Since its presentation, this algorithm has not only been credited with winning competitions but also for being the driving force for several cutting-edge industry applications. In this study, it used the same strategy.

The extreme gradient boosting (XGBoost) algorithm is an advanced version of gradient boosting, which gives better performance and less computational time. $L(\theta)$ represents the objective function [31, 32]. It consists of two factors, training loss and the regularization term, that can be described as the following:

$$L(\phi) = \sum_i \text{loss}(\hat{y}_i, y_i) + \sum_k \Omega(f_k)$$  \hspace{1cm} (1)

Where

$$\Omega(f) = \gamma T + \frac{1}{2} \lambda \|\theta\|^2$$  \hspace{1cm} (2)

$T$: number of leaf node of the tree

$f_k$: independent tree structure

$q$: leaf weights

$\text{loss}()$: the loss function that measures the difference between the actual $y_i$ and its prediction $\hat{y}_i$.

$\Omega$: used to penalizes the complexity of the model for avoiding overfitting.

$\gamma$: the leaf weight penalty parameter [32]

$\lambda$: the tree size penalty parameter [32]

In Euclidean space, equation (2) cannot be optimized using traditional optimization methods. So, $f_t$ is added to minimize the following objective $L^{(t)}$ which represents the objective function at the $t^{th}$ iteration.

$$L^{(t)} = \sum_i \text{loss}(y_i, \hat{y}_i^{(t-1)} + f_t(x_i)) + \Omega(f_t)$$  \hspace{1cm} (3)
To quickly optimize the objective a 2nd-order approximation is used in the general [32]. So, equation (3) become as the following:

$$L^{(2)} = \sum_{i=1}^{n} [l(y_i, \hat{y}_i^{(t-1)}) + g_i f_i(x_i) + \frac{1}{2} h_i f_i^2(x_i)] + \Omega(f)$$

The classification type of our case study is binary. So the gradient and hessian will be [33,34]

$$g_i = y_i - \hat{y}_i$$

$$h_i = \hat{y}_i (1 - \hat{y}_i)$$

And by removing the constant terms and expanding $\Omega$, the equation (4) will be written as:

$$\hat{L}^{(2)} = \sum_{j=1}^{T} \left[ \left( \sum_{i \in S_j} g_i \right) \omega_j + \frac{1}{2} \left( \sum_{i \in S_j} h_i + \lambda \right) \omega_j^2 \right] + \gamma T$$

The $S_j = \{i \mid q(x_i) = j\}$ is defined as instance set of leaf j. For a fixed structure $q(x)$, the optimal weight $\omega_j^*$ of leaf j can be computed as following [33]:

$$\omega_j^* = \frac{\sum_{i \in S_j} g_i}{\sum_{i \in S_j} h_i + \lambda} + \gamma T$$

The Equation (5) can be rewritten by $\omega_j^*$ as the following, which can be used to measure the quality of a tree structure $q$:

$$\hat{L}^{(2)}(q) = -\frac{1}{2} \sum_{j=1}^{T} \left( \sum_{i \in S_j} g_i \right)^2 + \frac{1}{2} \left( \sum_{i \in S_j} h_i + \lambda \right)$$

 Normally it is impossible to enumerate all the possible tree structures. So, a greedy algorithm equation (8) is used in XGBoost to evaluate the split candidates [21]. ($S_L$ and $S_R$ are the instance sets of left and right nodes after the split, and $S = S_L \cup S_R$)

$$L_{\text{split}} = \frac{1}{2} \left[ \frac{\left( \sum_{i \in S_L} g_i \right)^2}{\sum_{i \in S_L} h_i + \lambda} + \frac{\left( \sum_{i \in S_R} g_i \right)^2}{\sum_{i \in S_R} h_i + \lambda} - \frac{(\sum_{i \in S_L} g_i)^2}{\sum_{i \in S_L} h_i + \lambda} \right] - \gamma$$

4.2. The XGBoost workflow

It implements machine learning algorithms under the Gradient Boosting framework. XGBoost provides a parallel tree boosting. In the beginning, the datasets partitioning to 70% for training and 30% for testing. The hyperparameters are tuning and several trees are building depend on the best feature (gene) selection by the XGBoost then n trees are built depend on several criteria; the best split point, tree pruning and others. After the trees are built the predicting stage begins. The test data go through the building trees then take the average of the outputs. In Figure 2 is shown how the workflow goes through the XGBoost [35].

5. Comparative analysis results

In this study, it applied XGBoost to different types and sizes of datasets to show the ability of XGBoost in handling them and compare its results with the representation machine learning algorithms: SVM [36] and gcForest [37].
5.1. Tuning parameters

In this study, the XGBoost parameters is settled to the following: Max depth to 6 levels, learning rate (eta) to 0.3, the gamma (for splitting threshold) to 1, lambda (for regularization) to 1 and the number of rounds to 5. While the SVM parameters have the following values: It chooses the Radial Basis Function (RBF) for the kernel. The parameter C which behaves as a regularization parameter in the SVM set to 1, and gamma to 1. And for gcForest, the Max. depth equal 6, slicing window is 100, the step of window moving is 1 feature at a time and the number of trees for each of multi-grained stage and cascading stage is 500.

5.2. The results

Each algorithm has been applied five times for different random training sets and takes the average of each metric as shown in Tables 2, 3, 4, 5 and 6. And its corresponding charts in Figures 3, 4, 5, 6 and 7. While in Figures 8, 9, 10 and 11 illustrate the ROC and AUC metrics.

Table 2. Result comparison of lung cancer detection for GSE81089 dataset.

| Classifier Type | Sensitivity | Specificity | Precision | Accuracy | Std. | Time (hr.) |
|-----------------|-------------|-------------|-----------|----------|------|------------|
| XGBoost         | 1           | 0.95        | 0.997     | 0.997    | 0.006| 00:02:12   |
| SVM             | 0.991       | 0.48        | 0.991     | 0.971    | 0.01663| 00:05:23   |
| gcForest        | 0.893       | 1           | 1         | 0.988    | 0.01122| 00:43:00   |

Table 3. Result comparison of lung cancer detection for GSE30219 dataset.

| Classifier Type | Sensitivity | Specificity | Precision | Accuracy | Std. | Time (hr.) |
|-----------------|-------------|-------------|-----------|----------|------|------------|
| XGBoost         | 0.917       | 1           | 1         | 0.995    | 0.0157| 00:03:10   |
| SVM             | 1           | 0.214       | 0.969     | 0.969    | 0.02129| 00:03:69   |
| gcForest        | 1           | 0.583       | 0.978     | 0.978    | 0.01508| 01:10:00   |
Table 4. Result comparison of lung cancer detection for GSE19188 datasets.

| Classifier Type | Sensitivity | Specificity | Precision | Accuracy | Std. | Time (hr.) |
|-----------------|-------------|-------------|-----------|----------|------|------------|
| XGBoost         | 0.971       | 0.940       | 0.956     | 0.957    | 0.0136 | 00:01:32   |
| SVM             | 0.972       | 0.957       | 0.975     | 0.963    | 0.0195 | 00:01:37   |
| gcForest        | 0.964       | 1           | 1         | 0.979    | 0.015  | 01:31:19   |

Table 5. Result comparison of Lung cancer relapse prediction for GSE8894 dataset.

| Classifier Type | Sensitivity | Specificity | Precision | Accuracy | Std. | Time (hr.) |
|-----------------|-------------|-------------|-----------|----------|------|------------|
| XGBoost         | 0.964       | 0.7         | 0.815     | 0.794    | 0.061 | 00:02:06   |
| SVM             | 0.6         | 0.476       | 0.522     | 0.537    | 0.061 | 00:01:40   |
| gcForest        | 0.526       | 0.636       | 0.556     | 0.585    | 0.064 | 02:09:12   |

Table 6. Result comparison of lung cancer relapse prediction for GSE68219 dataset.

| Classifier Type | Sensitivity | Specificity | Precision | Accuracy | Std. | Time (hr.) |
|-----------------|-------------|-------------|-----------|----------|------|------------|
| XGBoost         | 0.663       | 0.592       | 0.641     | 0.632    | 0.0264 | 00:01:20   |
| SVM             | 0.955       | 0.093       | 0.617     | 0.615    | 0.0280 | 00:01:25   |
| gcForest        | 0.818       | 0.279       | 0.635     | 0.606    | 0.0425 | 02:12:02   |

Figure 3. Specificity and Sensitivity of XGboost, SVM and gcForest for all tables.
Figure 4. Precision of XGboost, SVM and gcForest for all tables.

Figure 5. Accuracy of XGboost, SVM and gcForest for all tables.

Figure 6. Standard deviation of XGboost, SVM and gcForest for all tables through the five times different training sets.

Figure 7. The learning time taken by XGboost, SVM and gcForest for all tables.
It can illustrate the function loss performance of the XGBoost model on the evaluation dataset by plotting it with classification error to get insight into how learning unfolded while training, see Figure 12, 13 and 14

**Figure 8.** The ROC curve and AUC values for each of gcForest, XGBoost and SVM models on GSE81089 dataset.

**Figure 9.** The ROC curve and AUC values for each of gcForest, XGBoost and SVM models on GSE30219 dataset.

**Figure 10.** The ROC curve and AUC values for each of gcForest, XGBoost and SVM models on GSE19188 dataset.

**Figure 11.** The ROC curve and AUC values for each of gcForest, XGBoost and SVM models on GSE8894 dataset.

Figure 1. The XGBoost log loss and classification error on GSE81089 dataset.

Figure 2. The XGBoost log loss and classification error on GSE30219 dataset.

Figure 3. The XGBoost log loss and classification error on GSE19188 dataset.
5.3. Analyzing the results
In this study, it used sensitivity, specificity, precision, accuracy, standard deviation, learning time, ROC curve and AUC metrics to evaluating the effectiveness of each of the machine learning models.

5.3.1. Sensitivity and Specificity. Sensitivity (true positive rate), measures the proportion of truly prediction lung cancer disease detected by machine learning to actual lung cancer disease cases. While specificity (true negative rate) measures the proportion of truly healthy cases predicted by Machine learning to actual health cases. In Figure 3 shows the charts of sensitivity and specificity for all subsets. These two metrics are always evaluated together because it gives more detail on the machine learning behaviour especially for unbalanced in distributed classes in the datasets like in GSE81089 and GSE30219 datasets (see Table 1). In the GSE81089 dataset the XGBoost has a 100% value in sensitivity and 95% in specificity while SVM has 99.1% in sensitivity metric but it has 48% in specificity, which means, it failed to train the system in detecting the healthy case. The gcForest has a better state than SVM, but it also considers less than XGBoost. While in GSE81089 both SVM and gcForest algorithms have 100% for each, which is higher than XGBoost (that have 91.7%) but they are both failed in specificity by having 21.4% for SVM and 58.3% for gcForest, while XGBoost has 100%. The rest of the datasets are having a sort of balance classes distributed. In GSE19188, the three models have a slight difference in both sensitivity and specificity. The SVM has the best result in sensitivity than XGBoost and gcForest, by taking 97.2%, 97.1% and 96.5% for each of them respectively. And in specificity, the gcForest has the greatest score, by taking 100%. Then SVM has 95.7% and 94% for XGBoost. In GSE8894 and GSE68219, there are both interesting in lung cancer relapse prediction. In GSE8894, the XGBoost has a higher degree in both sensitivity and specificity than gcForest. SVM although has higher score in sensitivity than gcForest but they failed in specificity by having 47.6%. In GSE68219, although both of SVM and gcForest have a better score in sensitivity than XGBoost, but they are failed in specificity by having 9.3% and 27.9% respectively, while XGBoost has 59.2%.

5.3.2. Precision It is not the same as sensitivity, it is the truly detected of lung cancer divided by true and false detection of lung cancer. It means that it can give us information on how well the machine learning model performs concerning the incorrect prediction. It can recognize from Figure 4 that, the values of the precision are slightly different between the models for all datasets, except when applying to the GSE8894 dataset. In this dataset, the XGBoost has a much higher value than gcForest and SVM models by having 81.5%, while gcForest has 55.6% and 52.2% for SVM.

5.3.3. Accuracy. It the proportion of all true prediction (whether they are lung cancer detection or healthy case) to all prediction. The XGBoost has a higher value in all datasets. Accuracy alone doesn't give us the full picture of system performance when it works with a class-imbalanced dataset, where there is a significant difference between the number of positive and negative classes. As it seen in Figure 5 for GSE81089 and GSE30219 datasets, the SVM have a high accuracy value (97.1% and 96.9% respectively) while it failed in specificity; which is the prediction of healthy case. So, it needs to farther metrics to give the full evaluation to the used machine learning. But the accuracy is work well in balance datasets.

5.3.4. Standard deviation (Std.). It can be used to measure confidence of the machine learning model against real world data. The less the value the more confidence it is. From Figure 6 it seen that, in all the datasets the XGBoost has the most less value. Which it is mean that, the XGBoost is the most confidence between the comparative models. Then the gcForest is the second one in GSE81089, GSE30219 and GSE19188 datasets. While in GSE8894 and GSE68219 the SVM is the second.

5.3.5. The learning time. Machine learning models needs enough time to let the model learn and develop to fulfil their purpose with a considerable amount of relevancy and accuracy. From Figure 7,
it seen that XGBoost has the minimum learning time for most of datasets. While gcForest has the longest time comparing to the other models.

5.3.6. The ROC curve and AUC. The ROC (receiver operating characteristic) curve in our case, is the relation of true cancer rate in y-axis and false cancer rate on x-axis. So, it concerned the true and false prediction. It used with binary classification and it’s highly considered in imbalance datasets. While the AUC is the Area Under Curve, its implicit goal is to deal with situations where have a very skewed sample distribution, and don’t want to overfit to a single class. As it seen in Figure (8, 9, 10 and 11) the XGBoost in all datasets either has the highest value or very close to it.

5.4. The pros and cons of XGBoost

Through this study and from the results obtained, it can notice some pros and cons to this XGBoost model. They summarized in Table 7.

| The Pros | The cons |
|----------|----------|
| It is extremely fast because of its parallel computation capability. | It is only working with numeric features. |
| Highly efficient in balance and imbalanced datasets. | May lead to an overfitting state, if hyperparameters are not tuned properly. |
| It is versatile because it can be regression, classification or ranking. | |
| Missing values is imputation, so there is no need for feature engineering. | |

6. Conclusions

This study produced a prediction model using XGBoost classification algorithm. The results showed that, using XGBoost classification algorithms improved the prediction of lung cancer diagnosis detection and relapse prediction. It showed that XGBoost algorithm can manage the unbalance datasets better and within a shorter time comparing with other comparative methods. As it is seen from the prognosis detection and relapse prediction results how much the accuracy of classification is strongly dependent on the sensitive and precise measurement levels of gene expression in the datasets. Therefore, our future direction is to investigate the best datasets preprocessing that reduce the noise, to increase the detection and prediction accuracy of the model.

References
[1] Bora Lee et al. 2020 DeepBTS: Prediction of recurrence free survival of non-small cell lung cancer using a time-binned deep neural network, Sci. Rep. 10 1952.
[2] Xufan Li, Jia Li, Pin Wu, Liyuan Zhou, Bingjian Lu, Kejing Ying, Enguo Chen, Yan Lu and Pengyuan Liu 2018. Smoker and non-smoker lung adenocarcinoma is characterized distinct tumor immune microenvironments OncoImmunology 7 10.
[3] Yanfang Wang, Quanli Zhang, Zhaojia Gao, Shan Xin, Yanbo Zhao, Kai Zhang, Run Shi and Xuanwen Bao 2019 A novel 4-gene signature for overall survival prediction in lung adenocarcinoma patients with lymph node metastasis Cancer Cell Int. 19 100.
[4] E. F. Nuwaysir, M Bittner, J. Trent, J. C. Barrett and C. A. Afshari 1999 Microarrays and toxicology: the advent of toxicogenomics Molecular Carcinogenesis 24 3 pp 153-9.
[5] Yi YangEric, A.G. BlommeEric, A.G. BlommeJeffrey, F WaringJeffrey and F Waring 2004 Toxicogenomics in drug discovery: From preclinical studies to clinical trials Chem. Biol. Interact. 150 1 pp 71-85
[6] Héctor A. Rueda-Zárate, Iván Imaiz-Rosshandler, Roberto A. Cárdenas-Ovando, Juan E. Castillo-Fernández, Julieta Noguez-Monroy and Claudia Rangel-Escareño 2017 A computational toxicogenomics approach identifies a list of highly hepatotoxic compounds from a large microarray database Plos One 12 4.

[7] Russul Al-Anni, Jingyu Hou, Rana Dhaia’a Abdu-aljabar and Yong Xiang 2017 Prediction of NSCLC recurrence from microarray data with GEP IET systems biology, 11 3 pp77-85.

[8] Russul Al-Anni, Jingyu Hou, Hasseeb Azzawi, and Yong Xiang 2018 Cancer adjuvant chemotherapy prediction model for non-small cell lung cancer IET systems biology 13 3.

[9] Russul Al-Anni, Jingyu Hou, Hasseeb Azzawi, and Yong Xiang 2018 Risk classification for NSCLC survival using microarray and clinical data Proc. of 207th The IIER Int. Conf. 12th-13th December Paris France.

[10] Russul Al-Anni, Jingyu Hou, Hasseeb Azzawi, and Yong Xiang 2018 A novel gene selection algorithm for cancer classification using microarray datasets BMC Med. Genomics 12 10.

[11] Russul Al-Anni, Jingyu Hou, Hasseeb Azzawi and Yong Xiang 2018 deep gene selection method to select genes from microarray datasets for cancer classification BMC-informatics 20 608.

[12] Russul Al-Anni, Jingyu Hou, H. Azzawi and Y. Xiang 2019 New Gene Selection Method Using Gene Expression Programming Approach on Microarray Data Sets Int. Conf. on Computer and Information Science 4th Sep. 2018 Springer, Cham. 791 pp 17-31.

[13] Hasseeb Azzawi, Jingyu Hou, Yong Xiang and Russul Alanni 2016 Lung cancer prediction from microarray data by gene expression programming IET Syst. Biol. 10.5 168-78

[14] Hasseeb Azzawi, Jingyu Hou, Russul Alanni, Yong Xiang, R. Abdu-aljabar and Ali Azzawi 2017 Multiclass lung cancer diagnosis by gene expression programming and microarray datasets 13th Int. Conf. on Advanced Data Mining and Applications 14 Oct. 2017 Singapore Springer, Cham. chapter 38 pp 541-53.

[15] Hasseeb Azzawi, Jingyu Hou, Russul Alanni, Yong Xiang 2018 SBC: A new strategy for multiclass Lung cancer classification based on tumour structural information and microarray data 17th IEEE/ACIS Int. Conf. on Computer and Information Science 6-8 June 2018 Singapore IEEE pp 68-73.

[16] Hasseeb Azzawi, Jingyu Hou, Russul Alanni and Yong Xiang 2018 A hybrid neural network approach for lung cancer classification with gene expression dataset and prior biological knowledge Int. Conf. on Machine Learning for Networking May 2019 Paris France Springer, Cham Lecture Notes in Computer Science 11407 pp 279-93.

[17] Guobin Zhang, Zhiyong Yang, Li Gong, Shan Jiang and Lu Wang 2019 Classification of benign and malignant lung nodules from CT images based on hybrid features Physics in Medicine & Biology 64 12.

[18] Shulong Li, et al 2019 Predicting lung nodule malignancies by combining deep convolutional neural network and handcrafted features Physics in Medicine & Biology 64 17.

[19] Patra R. 2020 Prediction of Lung Cancer Using Machine Learning Classifier. Int. Conf. on Computing Science, Communication and Security Computing Science, Communication and Security Springer Singapore 1235. pp 132-42.

[20] Yu-Heng Lai, Wei-Ning Chen, Te-Cheng Hsu, Che Lin, Yu Tsao and Semon Wu5 2020 Overall survival prediction of non-small cell lung cancer by integrating microarray and clinical data with deep learning Sci. Rep. Nature research 10 4679

[21] Tianqi Chen, Carlos Guestrin 2016. XGBoost: A scalable tree boosting system. In Proc. of the 22Nd ACM SIGKDD Int. Conf. on Knowledge Discovery and Data Mining pp 785–94.

[22] Adi L. Tarca, Roberto Romero and Sorin Drughi 2006 Analysis of microarray experiments of gene expression profiling American Journal of Obstetrics & Gynecology 195 2 pp 373-88.

[23] Jun Zhang,Rodrick, J Chiodini,Rodrick, J Chiodini,Ahmed, Badr,Ahmed, Badr Genfa and Zhang Genfa Zhang 2011 Journal of Genetics and Genomics 38 3 pp 95-109.

[24] Sophie Rousseaux, et al. 2013 Ectopic activation of germline and placental genes identifies aggressive metastasis-prone Lung cancers Science Translational Medicine 5 186.
[25] Hou J, Aerts J, et al. 2010. Gene expression-based classification of non-small cell Lung carcinomas and survival prediction PLoS One 5 4.

[26] Yamauchi M, et al. 2012 Epidermal growth factor receptor tyrosine kinase defines critical prognostic genes of stage I Lung adenocarcinoma PLoS One 7 9.

[27] Lee ES, et al. 2008. Prediction of recurrence-free survival in postoperative non-small cell lung cancer patients by using an integrated model of clinical information and gene expression Clin. Cancer Res. 14 22 pp 7397–404.

[28] Artur Mezheyevski, Christian Holst Bergsland, Max Backman, Dijana Djureinovic, Tobias Sjöblom, Jarle Bruun and Patrick Micke 2018 Multispectral imaging for quantitative and compartment-specific immune infiltrates reveals distinct immune profiles that classify Lung cancer patients J Pathol. 244 4 pp 421-31.

[29] David C Bell, W Kelley Thomas, Katelyn M Murtagh, Cheryl A Dionne, Adam C Graham, Jobriah E Anderson and William R Glover 2012 DNA base identification by electron microscopy Microsc Microanal. 18 10 pp 49–53.

[30] Ozsolak F and Milos PM 2011 RNA sequencing: Advances, challenges and opportunities Nat Rev Genet 12 2 pp 87–98.

[31] Tianqi Chen and Carlos Guestrin 2016 ACM. XGBoost: A Scalable Tree Boosting System. In Proc. of the 22nd ACM SIGKDD Int. Conf. on Knowledge Discovery and Data Mining, pp 785–94.

[32] Jin Liu, Xin Zhang, Yunhui Li, Jin Wang and Hye-jin Kim 2017 Deep learning-based reasoning with multi-ontology for IoT applications IEEE Access 7 pp. 124688-701.

[33] J. Friedman, T. Hastie, and R. Tibshirani 2000 Additive logistic regression: a statistical view of boosting. Annals of statistics 28 2 pp 337-407.

[34] M. Shoaran, B. A. Haghi, M. Taghavi, M. Farivar and A. Emami-Neyestanak 2018 Energy-Efficient Classification for Resource-Constrained Biomedical Applications IEEE Journal on Emerging and Selected Topics in Circuits and Systems 8 4 pp. 693-707

[35] Yuanyuan Li, David M. Umbach, Adrienna Bingham, Qi-Jing Li, Yuan Zhuang and Leping Li 2019 Putative biomarkers for predicting tumor sample purity based on gene expression data BMC Genomics 20 1021

[36] Wang L 2005. Support Vector Machines: Theory and Applications (USA: Springer)

[37] Zhou Z-H and Feng J. 2017 Deep Forest: towards an alternative to deep neural networks. In: ArXiv e-prints 1702.08835v1.