Application of Swarm-Based Feature Selection and Extreme Learning Machines in Lung Cancer Risk Prediction

Priya Garg*, Deepti Aggarwal

Department of Computer Science, Delhi Technological University, Bawana Road, Delhi, India

*Corresponding author
doi: https://doi.org/10.21467/proceedings.115.1

ABSTRACT

Lung cancer risk prediction models help in identifying high-risk individuals for early CT screening tests. These predictive models can play a pivotal role in healthcare by decreasing lung cancer's mortality rate and saving many lives. Although many predictive models have been developed that use various features, no specific guidelines have been provided regarding the crucial features in lung cancer risk prediction. This study proposes novel risk prediction models using bio-inspired swarm-based techniques for feature selection and extreme learning machines for classification. The proposed models are applied on a public dataset consisting of 1000 patient records and 23 variables, including sociodemographic factors, smoking status, and lung cancer clinical symptoms. The models, validated using 10-fold cross-validation, achieve an AUC score in the range of 0.985 to 0.989, accuracy in the range of 0.986 to 0.99 and F-Measure in range of 0.98 to 0.985. The study also identifies smoking habits, exposure to air pollution, occupational hazards and some clinical symptoms as the most commonly selected lung cancer risk prediction features. The study concludes that the developed lung cancer risk prediction models can be successfully applied for early screening, diagnosis and treatment of high-risk individuals.

Keywords: Lung Cancer Risk Prediction, Machine Learning, Extreme Learning Machine, Swarm-based Techniques, Feature Selection

1   Introduction

According to WHO [1], lung cancer is the leading cause of cancer deaths worldwide, with around 2.09 million cases and 1.76 million deaths in 2018. In 2011, The National Lung Screening Trial (NLST) found that low-dose helical computed tomography (CT) can reduce the risk of lung cancer deaths by 15% to 20 % [2]. Although these screening tests can lower lung cancer mortality, they are only beneficial if given to the right target individuals who have a high risk of developing lung cancer. The reason is that the screening process involves exposure to radiations, invasive diagnostic procedures and false positives [3].

In recent years, lung cancer risk prediction models have been developed for identifying the individuals who are at a high risk of developing lung cancer in the future using various variables like socio-demographic factors, socio-economic factors, smoking history, family history of lung cancer, biomarkers and lung cancer clinical symptoms [4]. Such models help early diagnosis and treatment as they identify the target population for low-dose computed tomography (LDCT) screening. These models can increase the CT process's overall cost-effectiveness while minimizing the associated effects of harmful CT-radiations [3, 5]. Many lung cancer risk prediction models have been developed by past studies using different classification techniques, independent variables and validation procedures [3-6]. Although these prediction models achieve good performance, there

are certain sources of bias due to improper variable selection, lack of model validation and poor research design [3]. Additionally, different sets of independent variables have been used for the model development, but there is little analysis regarding which factors play an essential role in determining lung cancer risk.

Specific feature selection techniques have been applied to study and identify the significant variables in lung cancer prediction. Xie et al. [7] applied a combination of ReliefF and Fisher for feature selection, followed by BP Neural Networks, to classify lung cancer data and control health data. They determined 15 top risk factors from the 41 factors for lung cancer risk prediction and achieved an accuracy of 83.816%. Dallri [8] proposed a model using Genetic Algorithms (GA) for feature selection and fuzzy Extreme Learning Machine (ELM) for classification for lung cancer prediction and achieved an accuracy of 97.5%. Chen et al.[9] used Deep Neural Networks to identify the risk factors for lung cancer incidence in older people. The study found that cancer history, smoking-related factors and physical exercise level were essential factors and achieved an accuracy ranging from 0.927 to 0.962. Markaki et al. [10] proposed a lung cancer screening model using backward feature selection and Cox regression. The model identified seven risk factors for lung cancer risk prediction and achieved an externally validated AUC score of 0.87 within six years.

This study proposes novel lung cancer prediction models that use bio-inspired swarm-based techniques for feature selection, followed by Extreme Learning Machines for classification. Bio-inspired techniques are often used for optimal feature subset selection, but such techniques have not been employed in this research field. These techniques mimic real-life phenomena and are known for their global search capability and low computational time compared to exhaustive search [11]. The extensive use of these techniques in other medical fields [12, 13] motivated us to use these in lung cancer risk prediction. ELMs are randomization based neural network techniques that show the predictive power of deep neural networks while minimizing the learning time and large dataset requirements [14]. ELMs have been used as the classification technique in lung cancer prediction in previous studies and have shown high accuracy and generalization capability, which motivated us to use ELM as the classification technique in this study.

The major contributions of the study in the field of lung cancer risk prediction are:

- The study investigates the importance of feature selection in the improvement of risk prediction. This study also presents a comparison between various bio-inspired feature selection techniques.
- The results of the study also identify the most important features that are effective and contribute the most towards prediction.

The rest of the paper has the following sections. Section 2 describes the three bio-inspired swarm-based techniques (Particle Swarm Optimization, Artificial Bee Colonization and Firefly algorithm) used for feature reduction, Section 3 explains ELM, which is the classification technique, Section 4 presents the experimental setup, Section 5 shows the experimental results and observations, Section 6 concludes the study and possible future extensions.

2   Bio-Inspired Swarm Swarm-Based Techniques for Feature Selection

In this study, bio-inspired swarm-based techniques have been used for effective feature selection. These techniques use the properties and behaviour of biological forms like optimal food source location, luminescence attraction etc., to search for the optimized solution in the problem space. In this study, three such techniques have been used - Particle Swarm Optimization, Firefly algorithm, and Artificial Bee Colonization. These techniques have been further explained in the following subsections.

2.1 Particle Swarm Optimization

PSO [15] is a swarm-based optimization technique based on bird flocks' behaviour, which tends to optimize their position according to the food source available. For feature reduction problems, every particle in the swarm denotes a vector of size m, where m is the number of features or independent variables in the dataset. The algorithm optimizes each particle's position to get the highest fitness value which in this study is the AUC-ROC value achieved by selecting the features that are represented in the position vector of each particle. The pseudo-code of PSO is explained in Figure 1.

Algorithm 1: Pseudo Code of PSO

Initialization:

1. *Number of particles:* P
2. *Threshold:* Θ
3. *Initial position of each particle:*

    Pos = {pos_i}, i= 1,2,3...P and pos_i is an m-dimensional vector.
4. *Initial Velocity of each particle:*

    V = {v_i} i= 1,2,3...P and v_i is an m-dimensional vector
5. pbest = [$pbest_i$], i= 1,2,3...P and $pbest_i$ is a m-dimensional vector
6. gbest = {0} m-dimensional vector
7. maxiteration = it_{max}

Functions:

  nof(pos): Calculate number of features according to Θ and pos.

  fit(pos): Calculate the fitness function according to Θ and pos.

Algorithm:

for each particle i in (0, 1..., P):

  $pbest_i$ = pos_i

  if(fit($pbest_i$)>fit(gbest))

      gbest = $pbest_i$

Repeat for it_{max} iterations:

   for each particle i in (0, 1..., P):

Calculate fit($pbest_i$), fit(pos_i), fit(gbest), nof($pbest_i$), nof(pos_i), nof(gbest)

        if(fit(pos_i)>fit($pbest_i$))

          $pbest_i$ = pos_i

        if(fit($pbest_i$)>fit(gbest))

          gbest = $pbest_i$

   for each particle i in (0,1,.., P):

Update v_i to v_i' and pos_i to pos_i' according to equation (1) & (2) respectively

        $v_j^{t+1} = v_j^t * w + c_1 * r_{1j} * (pbest_j - x_j^t) + c_1 * r_{2j} * (gbest - x_j^t)$ (1)

        $x_j^{t+1} = x_j^t + v_j^{t+1}$ (2)

return gbest

Figure 1: Pseudo Code for PSO

2.2 Firefly Optimization

The firefly (FF) algorithm [16] is based on the attraction of fireflies towards each other according to their luminescence intensity. In feature selection problems, the intensity of fireflies represents the achieved AUC-ROC value using the position vector. The position vector represents a vector of m features, and each value ranges from 0 to 1 and features greater than a threshold is selected for the final problem and less are not. The pseudo-code of the FF optimization technique is given in Figure 2.

Algorithm 2: Pseudo Code of FF

Initialization:

1. *Number of fireflies:* F
2. *Threshold: θ*
3. *Alpha a*
4. *Beta β(Random Parameter)*
5. *gamma γ (Light intensity Coefficient)*
6. *Intensity vector I = {Ii} where i = 1...F*
7. *Pos = {Pos$_i$} where i = 1… F, Pos$_i$ = {P$_{i1}$...P$_{im}$} where m is number of features*
8. *maxiteration = it$_{max}$*

Functions:

obj(): Objective function to calculate Fitness value AUC-ROC

Algorithm:

Determine Intensity (I) of each firefly using objective function obj().

Repeat for it$_{max}$ iterations:

  For each firefly i = 1,2, … N:

    For each firefly j = 1,2, ...N:

      if(Ij > Ii)

        Move firefly i towards j according to equation (3)

       r = Distance between i and j firefly

        Dominance = $(\beta e \gamma r2 + a(z\text{-}0.5) + 1)/2$

       *Pos(i) = (1-*Dominance*)Pos(i) + *Dominance**Pos(j) (3)*

  Evaluate new solution and update Intensity (I)

  Rank the fireflies according to intensity and find gbest

return gbest

Figure 2 : Pseudo Code for FF

2.3   Artificial Bee Colonization

ABC [17] is a swarm-based optimization technique based on the food searching behaviour of bees. In this technique, different types of bees perform various iterations to search for the optimal position where nectar is present. The optimal position is defined by a fitness function that calculates the optimal selected feature that gives the best AUC-ROC value. The pseudo-code of the technique is given in Figure 3.

Algorithm 3: Pseudo Code of ABC

Initialization:

1. *Number of employed bees = Number of onlooker bees:* B
2. *Threshold: θ*
3. *Abandonment Limit = L*
4. *Posb = {Posb$_i$} where i = 1… B, Posb$_i$ = {P$_{i1}$...P$_{im}$} where m is number of features*
5. *maxiteration = it$_{max}$*

Functions:

Fitness(Posb$_i$): Objective function to calculate Fitness value AUC-ROC using position of a bee

Algorithm:

Repeat for *it$_{max}$* iterations:

  // Employed Bees

  For each employed bee 1…B:

    New food source vi initialization

      Calculate Fitness(vi)

Update Posb$_i$ to vi if Fitness(vi) > Fitness(Posb$_i$)

     Fit$_i$ = 1/Fitness(Posb$_i$)

   Calculate probability *pi = Fit/$\sum$Fit$_i$*

   //Onlooker Bees

   For each onlooker bee 1…B:

Select a food source with maximum probability (p)

     Produce a new food source at vi

Update Posb$_i$ to vi if Fitness(vi) > Fitness(Posb$_i$)

   //Scout Bees

   If any employed bee reach its abandonment limit Then

   Replace it with a new random position

   Store the best solution so far in gbest

return gbest

Figure 3: Pseudo Code for ABC

3   Extreme Learning Machine

Extreme Learning Machine (ELM) [14] is a class of randomization-based neural networks that use randomized weights and biases for the evaluation and analytically calculate the weights between the hidden and the output layers. This technique reduces the learning time to a great extent and maintains the predictive capability of neural networks. In this study, a single hidden layer ELM has been used with one input layer and one output layer as shown in Figure 4.
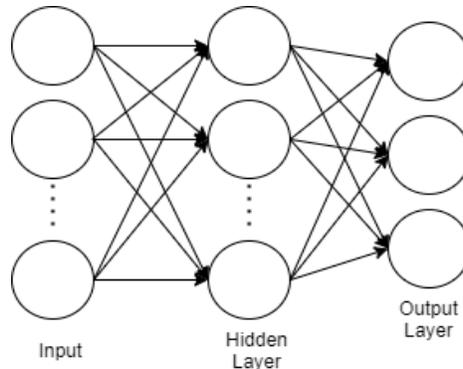
Figure 4 : Structure of Extreme Learning Machine

Consider dataset D = [X, Y] where X = {x$_1$, x$_2$, …. x$_N$}^{T} and x$_i$ = {x$_{i1}$, x$_{i2}$ … x$_{if}$}, Y = {y$_1$, y$_2$, … y$_N$}^{T} and y$_i$ = {y$_{i1}$, y$_{i2}$ … y$_{im}$} is the output vector, f be the number of independent variables in the dataset, N be the number of samples in the dataset, and m be the output classes in the dataset.

The ELM output equation is defined as:

$$\sum_{t=1}^{h} \beta_t * act\ (w_t.x_i\ +\ bias_t) = o_i \tag{4}$$

where h is the number of hidden layer nodes, β is the required weights between the hidden and the output layer, w$_t$ are the randomly initialized weights between input and hidden layer nodes, x$_i$ is the input feature vector, bias$_t$

is the bias of hidden layer, act() is the activation function, and o_i is the output vector of size m for sample i = 1, 2, 3, …, N.

Equation (4) can be re-written as:

$$\beta * H = O \tag{5}$$

where, $H = \begin{bmatrix} act(w_1 x_1 + bias_1) & \cdots & act(w_k x_1 + bias_h) \\ \vdots & \vdots & \vdots \\ act(w_1 x_N + bias_1) & \cdots & act(w_k x_N + bias_h) \end{bmatrix}_{NXh}$ $\tag{6}$

The goal of the problem is to minimize the output error i.e. $||Y-O|| = 0$ where Y is the target output vector and O is the expected output vector:

$$H * \beta = Y \tag{7}$$

The optimal value of β is given as:

$$\hat{\beta} = H^+ Y \tag{8}$$

In this study, we use Moore-Penrose generalized inverse H^+ of the hidden layer output matrix. The optimal value of β̂ gives the minimum output error and are the final weights between the hidden and output layer. The pseudo code of ELM is shown in Figure 5.

Algorithm 4: Pseudo Code of ELM

Initialization:
1. *Randomization Range for weights and bias : (-1,1)*
2. *Number of hidden layer nodes h*
3. *Weights W: {w_t} where t = 1…h*
4. *Biases bias:{$bias_t$} where t =1…h*
5. *Hidden layers: 1*

Algorithm:
1. Create H matrix i.e. output from hidden layer matrix using equation (6).
2. Calculate β matrix using H and Y according to equation (8).
3. Using β matrix and new H matrix created using the testing dataset, calculate the predicted output O using equation (5).
return O

Figure 5 : Pseudo Code for ELM

4 Experimental Setup

4.1 Datasets and Variables

The lung cancer risk prediction dataset used in the study is taken from the open-source data repository [18]. The dataset contains 1000 patient records and has 23 independent variables and one dependent variable. The dependent variable denotes the risk of the patient having lung cancer and has three values - "Low", "Medium", and "High", making this a multi-class classification dataset. The 23 independent variables are shown in Table 1.

Table 1: *Variables in the dataset*

Variable	Range	Variable	Range
Age	between 14-73 years	Passive Smoker	between 1-8
Gender	binary male and female	Chest Pain	between 1-9
Air Pollution	between 1-8	Coughing of Blood	between 1-9
Alcohol use	between 1-8	Fatigue	between 1-9
Dust Allergy	between 1-8	Weight Loss	between 1-8
Occupational Hazards	between 1-8	Shortness of Breath	between 1-9
Genetic Risk	between 1-7	Wheezing	between 1-8
Chronic Lung Disease	between 1-7	Swallowing Difficulty	between 1-8
Balanced Diet	between 1-7	Clubbing of Finger Nails	between 1-9
Obesity	between 1-7	Frequent Cold	between 1-7
Smoking	between 1-8	Dry Cough	between 1-7
Snoring	between 1-7		

4.2   Performance Measures

In this study, five performance measures are used to evaluate risk prediction models. These measures are - Precision, Recall, F-Measure, AUC-ROC and Accuracy. To define these measures, we first define the four terms that form a confusion matrix.

- True Positive (TP) - Number of positive samples that are predicted positive
- True Negative (TN) - Number of negative samples that are predicted negative
- False Positive (FP) - Number of negative samples that are predicted positive
- False Negative (FN) - Number of positive samples that are predicted negative.

The performance measures used are defined as:

- Recall is defined as the number of correctly predicted positive samples (TP) to the total positive samples predicted as positive (TP + FP).
- Precision is the ratio of the number of correctly predicted positive samples in the dataset (TP) to the total number of positive samples in the dataset (TP+FN).
- F-Measure is the harmonic mean of Precision and Recall.
- AUC-ROC is the area under the plot of True Positive Rate (TPR), and the FPR(False Positive Rate) where TPR is the same as the Recall and FPR is the ratio of the number of negative samples that are predicted positive to the total negative class samples in the dataset. It measures the predictive power of the classification technique.
- Accuracy is the sum of the correctly predicted positive and negative samples to the dataset's total samples.

The mathematical representation of these measures is given in Table 2. Since this is a multi-class classification problem, these metrics are evaluated using One-Vs-Rest method and micro-averaging of the values is performed on the individual class values to obtain the final results.

4.3 Validation Technique

This study uses 10-fold cross validation technique for training and evaluation of the classification techniques. In 10-fold cross validation technique, the dataset is randomly divided in 10 equal sized subsets. In one iteration, training is done on 9 subsets and testing is done on the remaining subset. This process is performed for 10 iterations allowing every subset to be a testing set once and the results of each iteration are macro-averaged to give the final performance measures.

4.4 Parameter Settings

The parameter settings of all the techniques have been carefully chosen using a grid search technique on various combinations. The final parameter settings of the techniques are shown in Table 2.

Table 2: *Parameter Settings used in the Study*

ML Technique	Parameter Settings
PSO	$c_1 = c_2 = 1.496$, w = 0.7298, number of particles = 30, number of iterations = 50, vmax =2, threshold for feature selection = 0.5.
ABC	number of employed bees = number of onlooker bees = 20, number of iterations = 50, limit for abandonment = 10, threshold for feature selection = 0.6
FF	population size = 100, alpha = 1, beta = 0.5, gamma = 1, number of function iterations = 10, threshold for feature selection = 0.7
ELM	initial randomization range for hidden weights and biases = (-1,1), activation function = sigmoid, number of hidden layer nodes = 80

5 Experimental Results and Observations

This section presents the results and observations of the experiments conducted in the study. All the experiments are conducted according to the framework described in Section 4. Figure 6 shows the AUC-ROC, F-Measure, Precision, Recall and Accuracy values for the three proposed models, i.e. PSO-ELM, ABC-ELM, and FF-ELM. The table also displays the performance measure values for a case where no feature selection technique is applied. PSO-ELM has achieved the highest value for all the performance measures, followed by ABC-ELM and then FF-ELM. It is observed that the case where no feature selection technique is applied performs the worst in terms of all performance measures. The application of swarm-based feature selection techniques has led to an increase in AUC-ROC in the range of 0.22% (for FF-ELM) - 0.68% (for PSO-ELM) the case where no feature selection technique is applied. Similarly, the accuracy is increased between 0.20% (for FF-ELM) - 0.61% (for PSO-ELM). A similar increase of more than 0.30% is observed for other performance indicators like Precision, Recall and F-Measure.

Table 3 shows the features selected by PSO, ABC and FF for training ELM classifiers. Factors like smoking habits, exposure to air pollution, occupational hazards, genetic risks are commonly selected. FF and ABC also include some clinical symptoms like snoring, weight loss, fatigue, shortness of breath and cold as strong predictors for lung cancer risk prediction.
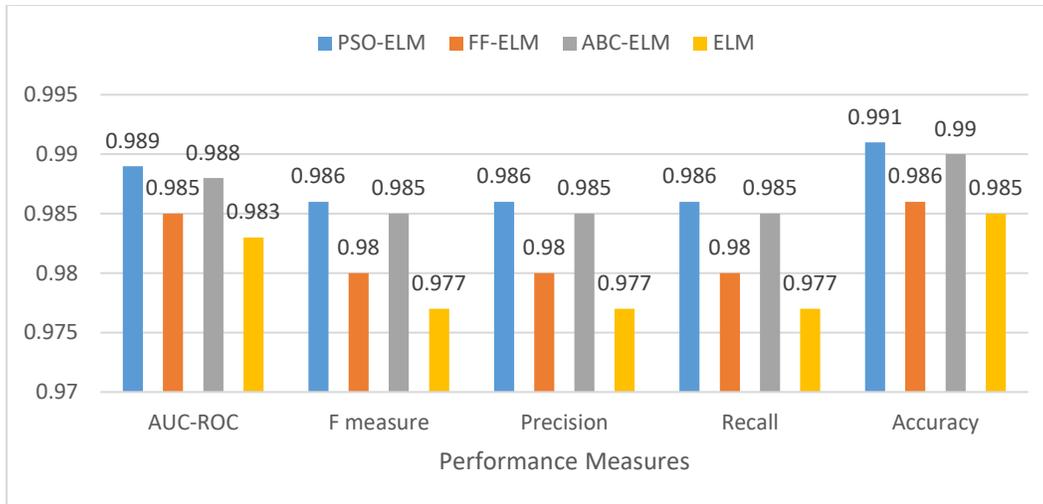
Figure 6: Results of ML Techniques

Table 3: *Features Selected by Different Techniques*

ML Technique	Selected Features
PSO-ELM	Age, Air Pollution, Genetic Risk, Chronic Lung Disease, Balanced Diet, Passive Smoker
FF-ELM	Gender, Air Pollution, Occupational Hazards, Chronic Lung Disease, Balanced Diet, Coughing of Blood, Fatigue, Shortness of Breath, Snoring
ABC-ELM	Alcohol use, Dust Allergy, Occupational Hazards, Genetic Risk, Obesity, Passive Smoker, Weight Loss, Swallowing Difficulty, Frequent Cold, Snoring

6   Conclusion and Future Work

The study proposed novel lung cancer risk prediction models using swarm-based techniques for feature selection and extreme learning machines for classification. A dataset of 1000 records and 23 variables was used where the risk was categorized into three classes according to severity. Five performance measures, i.e., AUC-ROC, Accuracy, Precision, Recall and F-Measure, were used to evaluate the model. The models achieved an AUC score in the range of 0.985 to 0.989, Accuracy in the range of 0.986 to 0.99 and F-Measure in the range of 0.98 to 0.985. PSO-ELM performed the best, followed by ABC-ELM in the second position and FF-ELM in the third position in terms of performance. All three models showed significant improvement over the baseline ELM with no feature selection in all performance measures. This shows the importance of identifying the effective subset of independent variables for lung cancer risk prediction through feature selection techniques. The improvement in performance of measures also show that the effective feature subset is much smaller than the actual number of features initially calculated. The most commonly selected features include exposure to air pollution, smoking, occupational hazards and clinical symptoms like snoring, shortness of breath and weight loss.

In the future, this study can be extended by including more categories of feature selection techniques, and the proposed models can be validated on more if publicly available datasets.

References

[1]   World Health Organization. Cancer; 2018. Available from: https://www.who.int/news-room/fact-sheets/detail/cancer

[2]   Reduced Lung-Cancer Mortality with Low-Dose Computed Tomographic Screening. New England Journal of Medicine, 365(5), 395–409 (2011). doi:10.1056/nejmoa1102873

[3]   Tang, W., Peng, Q., Lyu, Y., Feng, X., Li, X., Wei, L., Li, N., Chen, H., Chen, W., Dai, M., Wu, N., Li, J., Huang, Y. Risk prediction models for lung cancer: Perspectives and dissemination. Chin J Cancer Res 2019. 31(2) 316-328 (2019). doi: 10.21147/j.issn.1000-9604.2019.02.06

[4]   Toumazis, I., Bastani, M., Han, S. S., & Plevritis, S. K. (2020). Risk-Based Lung Cancer Screening: A Systematic Review. Lung Cancer. doi:10.1016/j.lungcan.2020.07.007

[5]   Field, J. K., Chen, Y., Marcus, M. W., Mcronald, F. E., Raji, O. Y., & Duffy, S. W. The contribution of risk prediction models to early detection of lung cancer. Journal of Surgical Oncology. 108(5), 304–311 (2013). doi:10.1002/jso.23384

[6]   Spitz, M. R., Hong, W. K., Amos, C. I., Wu, X., Schabath, M. B., Dong, Q., … Etzel, C. J. A Risk Model for Prediction of Lung Cancer. JNdCI Journal of the National Cancer Institute, 99(9), 715–726 (2007). doi:10.1093/jnci/djk153

[7]   Xie, N.-N., Hu, L., Li, T.-H. .Lung Cancer Risk Prediction Method Based on Feature Selection and Artificial Neural Network. Asian Pacific Journal of Cancer Prevention. 15(23), 10539–10542 (2015).

[8]   Daliri, M. R.. A Hybrid Automatic System for the Diagnosis of Lung Cancer Based on Genetic Algorithm and Fuzzy Extreme Learning Machines. Journal of Medical Systems. 36(2), 1001–1005 (2011). doi:10.1007/s10916-011-9806-y

[9]   Chen S, Wu S, Identifying Lung Cancer Risk Factors in the Elderly Using Deep Neural Networks: Quantitative Analysis of Web-Based Survey Data, J Med Internet Res 2020;22(3):e17695 (2020), doi: 10.2196/17695

[10]   Markaki, M., Tsamardinos, I., Langhammer, A., Lagani, V., Hveem, K., & Røe, O. D. A Validated Clinical Risk Prediction Model for Lung Cancer in Smokers of All Ages and Exposure Types: A HUNT Study. EBioMedicine, 31, 36–46 (2018). doi:10.1016/j.ebiom.2018.03.027

[11]   Selvaraj, C., A survey on application of bio-inspired algorithms. Int. J. Comput. Sci. Inf. Technol. **67**, 366–370 (2014)

[12]   Sakri, S. B., Abdul Rashid, N. B., Muhammad Zain, Z., Particle Swarm Optimization Feature Selection for Breast Cancer Recurrence Prediction. IEEE Access, 6, 29637–29647 (2018). doi:10.1109/access.2018.2843443

[13]   Sawhney, R., Mathur, P., Shankar, R., A Firefly Algorithm Based Wrapper-Penalty Feature Selection Method for Cancer Diagnosis. In: Gervasi O. et al. (eds) Computational Science and Its Applications – ICCSA 2018. ICCSA 2018. Lecture Notes in Computer Science, vol 10960. Springer, Cham. (2018) doi: 10.1007/978-3-319-95162-1_30

[14]   Huang, G. B., Zhu, Q.Y., Siew, C.K., Extreme learning machine: Theory and applications, Neurocomputing 70 (1-3), 489–501 (2006) doi: 10.1016/j.neucom.2005.12.126

[15]   Kennedy, J., Eberhart, R. (n.d.). Particle swarm optimization. Proceedings of ICNN'95 - International Conference on Neural Networks (1995). doi:10.1109/icnn.1995.488968

[16]   Emary, E., Zawbaa, H. M., Ghany, K. K. A., Hassanien, A. E., & Parv, B. Firefly Optimization Algorithm for Feature Selection. Proceedings of the 7th Balkan Conference on Informatics Conference - BCI '15 (2015). doi:10.1145/2801081.2801091

[17]   Karaboga, D., & Akay, B. A comparative study of Artificial Bee Colony algorithm. Applied Mathematics and Computation, 214(1), 108–132 (2009). doi:10.1016/j.amc.2009.03.090

[18]   [dataset] https://data.world/cancerdatahp/lung-cancer-data