LUNG CANCER DIAGNOSIS USING MACHINE LEARNING

[1]Samrajyam Singu, [2]Mahesh Tunguturi
Department of MIS Researcher/Sr. Software Engineer USA[1], Department of Information Technology Sr. Software Engineer USA[2]
samrajyamsingu@gmail.com[1], mahesht529@gmail.com[2]

ABSTRACT

Lung cancer is the world's most lethal and life-threatening disease. Although early identification and accurate treatment are essential to minimize lung cancer death rates. A computed tomography (CT) scan-based picture is one of the finest imaging modalities for lung cancer diagnosis utilizing deep learning algorithms. In this paper, we present a deep learning model based on Convolutional Neural Networks (CNN) for the early diagnosis of lung cancer utilizing CT scan pictures. We also compared our suggested model to various existing models. We discovered that CNN outperformed other models with an accuracy of 95%, an AUC of 96%, a recall of 95%, and a loss of 0.18.

Keywords— Lung cancer, CT scan imaging, Deep Learning.

INTRODUCTION

Lung cancer is one of the most lethal kinds of cancer in the globe. Cancer is difficult to diagnose, and its symptoms appear only in the late stages. Although early discovery and effective treatment for patients might reduce the fatality rate from this malignancy. Lung cancer often begins in the lungs; however, it can occasionally manifest as early signs before spread [1]. Numerous techniques have been developed in recent years, and research is ongoing to detect lung cancer effectively. CT scan images are the best imaging method for the early detection of lung cancer, but they can be difficult for medical professionals to interpret and detect cancer. [2]. Figure 1 displays projected statistics for a few cancer types in 2019. We utilized statistics to create this figure. American Cancer Society (ACS) statistics [3] According to the American Cancer Society, lung malignancy has the highest death rate of any cancer, accounting for around 0.13 million deaths worldwide. Every year, many new cases are reported, with an expected 0.237 million cases in 2019. In the absence of effective therapy, the mortality rate is large since this disease is only identified in its advanced stages. The ratio of new cases to death rate is higher than in any other cancer [4].

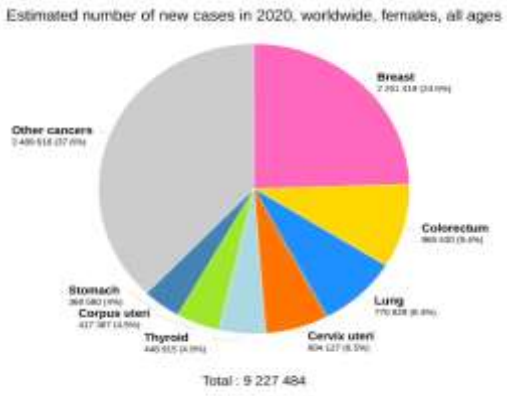


Figure 1 Cancer in 2019 Lung cancer cells

BACKGROUND

On a very large chest x-ray picture collection, Ausawalaithong et al. employed a convolutional neural network (CNN) to detect anomalies in chest x-rays. The authors evaluated the accuracy, specificity, and sensitivity of the model's performance using three retrained models and diverse datasets. Using the ChestX-ray14 dataset, Model A identified lung nodules. Model C recognized lung cancer using both ChestX-ray14 and JSRT, and although having a lower standard deviation in all assessment measures, it revealed an accurate location of the lung cancer. Model B, which employed the Japanese Society of Radiological Technology (JSRT) dataset, displayed greater specificity but lowered accuracy and sensitivity than Model C. They advocated retraining the model numerous times as an approach.

The CNN configuration developed by the PSO algorithm was utilised by Da Silva et al. [6]. It was trained and verified on identical sets to enable precise particle comparison. The authors gathered information from the LIDC-IDRI dataset. The findings were obtained using five test subgroups. With 17,870 samples, Test-1 produced results of 96.54% accuracy, 87.79% sensitivity, 98.215% specificity, and 0.931 AUC. One of the five test subsets, Test-4, produced the best results, with 97.62% accuracy, 92.20% sensitivity, 98.64% specificity, and 0.955 AUCStacked Autoencoder + Softmax were utilised by Naqi et al. [7]. The authors propose that nodules be classified using a mix of 2D and 3D information. Deep learning is utilised for feature reduction and nodule classification. The experiment makes use of the LIDC-IDRI data collection, which is freely available to the public. The performance features of this investigation, which include sensitivity, specificity, accuracy, and the number of FPs/scans, are the primary assessment criteria for this study. The study includes 888 CT images with 777 sized 3 mm nodules that were noticed by all four expert radiologists. The proposed method resulted in low false positive rates of 2.8/scan, 95.6% sensitivity, 96.9% accuracy, and 97.0% specificity, significantly improving the results.

Deep autoencoder was utilised by Shaffie et al. Click or tap here to enter text.. It presents a novel automated noninvasive clinical diagnostic technique for the early detection of lung cancer by determining whether the observed lung nodule is benign or malignant. The authors' method produced encouraging results using the LIDC-IDRI data set. This study's performance parameters were sensitivity, specificity, accuracy, and AUC. According to the categorization data produced from a collection of 727 nodules collected from 467 individuals, the suggested framework has the potential for early lung cancer detection with an accuracy of 91.20%, specificity of 95.88%, sensitivity of 85.03%, and AUC of 95.73.

The suggested CNN network was built by Kaur et al. [9] using CNN, three sets of rectified linear unit (ReLU) layers, and convolutional layers, followed by a fully connected layer. At each convolutional layer, 64 filters recover the representative features. The Japanese Society of Radiological Technology (JSRT) dataset is utilised for validation and training. The average accuracy was 98.05%, the overlap was 93.4%, the sensitivity was 96.25%, and the specificity was 98.80%.

Xie et al. [10] suggested a deep neural network model for benign-malignant lung nodule classification on chest CT using a multi-view knowledge-based collaborative (MV-KBC). Using the reference LIDC-IDRI data set, they compared their methodology against the five cutting-edge classification algorithms. The MV-KBC model correctly classified lung nodules, according to their findings.

Zhang et al. [11] used a deep belief network to solve their problem (DBN). The authors also made use of the freely accessible dataset LIDC-IDRI. When utilised to detect large nodules larger than 30 mm, the algorithm's accuracy, sensitivity, and specificity were all more than 90%. The articles' sensitivity ranged between 84.2% and 87.1%, and their accuracy ranged between 89.0% and 89.5%.

Causey et al. [12] provide NoduleX, an effective framework based on deep learning convolutional neural networks for predicting lung nodule malignancy from CT images of patients (CNN). The authors examined the LIDC/IDRI cohort nodules for training and validation and discovered that NoduleX can achieve 0.99 AUC on the independent validation test with an accuracy of 94.6%, sensitivity of 94.8%, and specificity of 94.3%

METHODOLOGY

The procedure starts with a picture dataset taken from a publically accessible source. After that, the picture dataset is pre-processed. The suggested CNN model, as well as additional deep learning models such as ResNet-50, Inception V3, and Xception, are then trained, tested, and validated using the usual hold-out-validation approach on the CT scan dataset[14]. The outcomes are computed and examined to establish the optimum deep learning-based model for detecting lung tumours such as adenocarcinoma, large cell carcinoma, and squamous cell carcinoma, as well as normal lung tissue (not lung cancer). CNN is a model that has been trained from scratch, whereas ResNet-50, Inception V3, and Xception are pre-trained transfer learning models**.**

Collection of datasets:

The lung cancer Dataset (CT scan Images) used in this study was obtained from the publically available "Kaggle" internet source [15]. The photos were hand-collected from multiple sources, according to the dataset source, with each label validated. To match the model, photos are in JPG or PNG format rather than DCM. The data set includes 967 CT scan pictures. The dataset contains four classes for diagnosing lung cancer: adenocarcinoma, large cell carcinoma, squamous cell carcinoma, and normal (not lung cancer).Dataset pre-processing:

The images were pre-processed using feature extraction, which included reading the images, resizing them, removing noises (de-noise), image segmentation, and morphology (smoothing edges). This processing system is essential for analyzing deep learning models for image classification or detection.

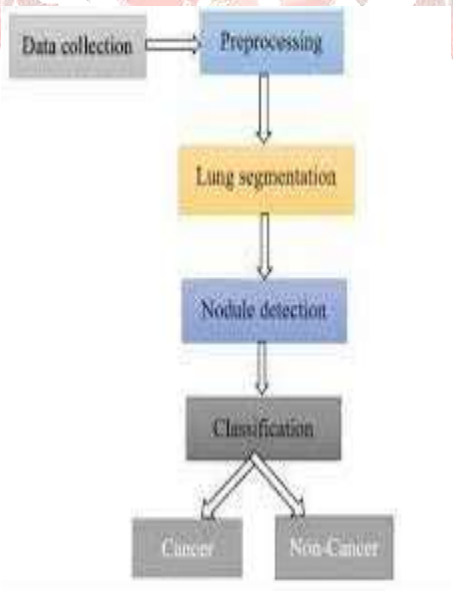


Figure 2 Overview of the study

VALIDATION PROCESS:

For large image datasets, it is critical to choose the best validation procedure. We used a hold-out validation process, keeping 70% of the data for training, 15% for testing, and 15% for validating. The hold-out validation

technique is the most commonly used method and produces effective results [16]. For all the deep learning models, we selected the epochs value of 50 and batch size value of 13. We also used a random seed value of 1000 while implementing all the models, so that we can get the re-producible implemented results, or else the results would change in every iteration[17].

PROPOSED CNN ARCHITECTURE:

The 64x64 input image was first sent to a first convolution layer in the proposed CNN, which has a value of 16 filters and 62x62 feature maps to look for the most fundamental features. The convolutional layer was the main building block of CNN. The output of the convolutional layer was then passed on to a max pooling layer with feature maps of 31x31 in order to reduce the size of the spatial data for the subsequent layer by half. Max pooling selects the maximum elements or pixels from the area of the feature map covered by the filter. Then, for additional processing, this output was sent to a second convolution layer with a value of 32 filters and 29x29 feature maps. The output of this layer was then passed on to a max pooling layer with 14x14 feature maps in order to decrease the amount of spatial data for the subsequent layer in half. Another set of convolution and pooling layers was added in the third step. In this case, the pooling layer consisted of 5x5 feature maps and the convolution layer was consisting of 64 filters with 10x10 feature maps. Then, the final output was flattened and moved to the 260-dimensional fully connected dense layer. After that, it is routed to the activation function layer which was softmax. Softmax activation function is generally used for multiple classifications. Except for the final layer, all layers used a ReLU activation function with no dropout. Figure 3 depicts the proposed CNN architecture's above-mentioned layout. With a learning rate of 0.01, 50 epochs, and 13 batch sizes, the model was trained, validated, and tested. The model was compiled using the Adam optimizer. Using the Keras Python library, a categorical cross-entropy based loss function and other metrics such as accuracy, recall, and AUC were achieved.

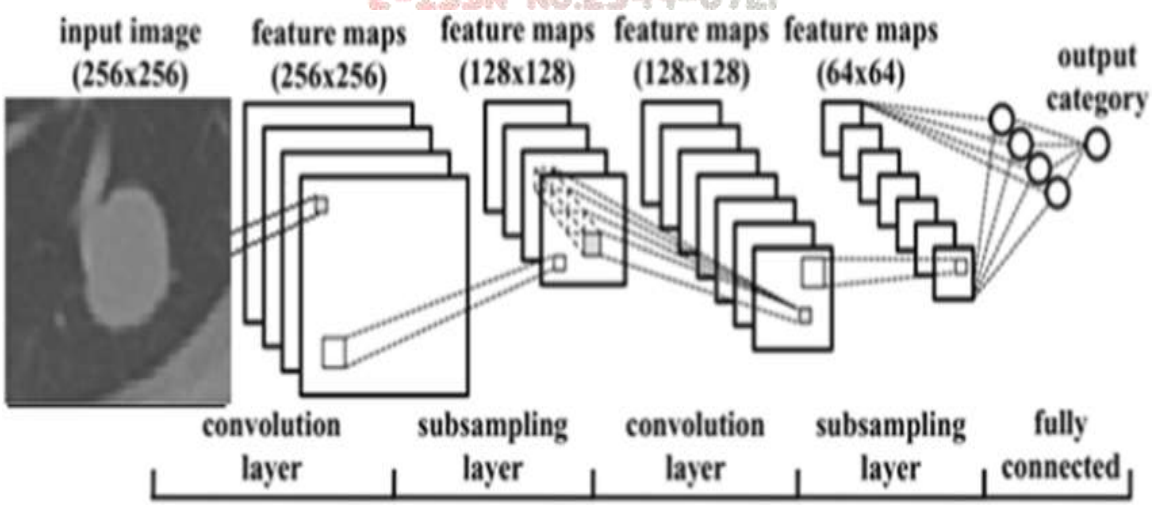


Figure 3 CNN architecture

RESULTS AND DISCUSSION

The findings of four types of deep learning models - i.e. CNN, ResNet-50, Inception V3, and Xception classification algorithms on the Lung cancer CT scan image dataset have been computed in table I and

comparisons have been provided in Figure 4. CNN considered the proposed model for lung cancer detection by CT scan images. The CNN achieved a testing accuracy of 92%, a testing AUC of 98.21%, testing recall of 91.72%, and a testing loss of 0.328.

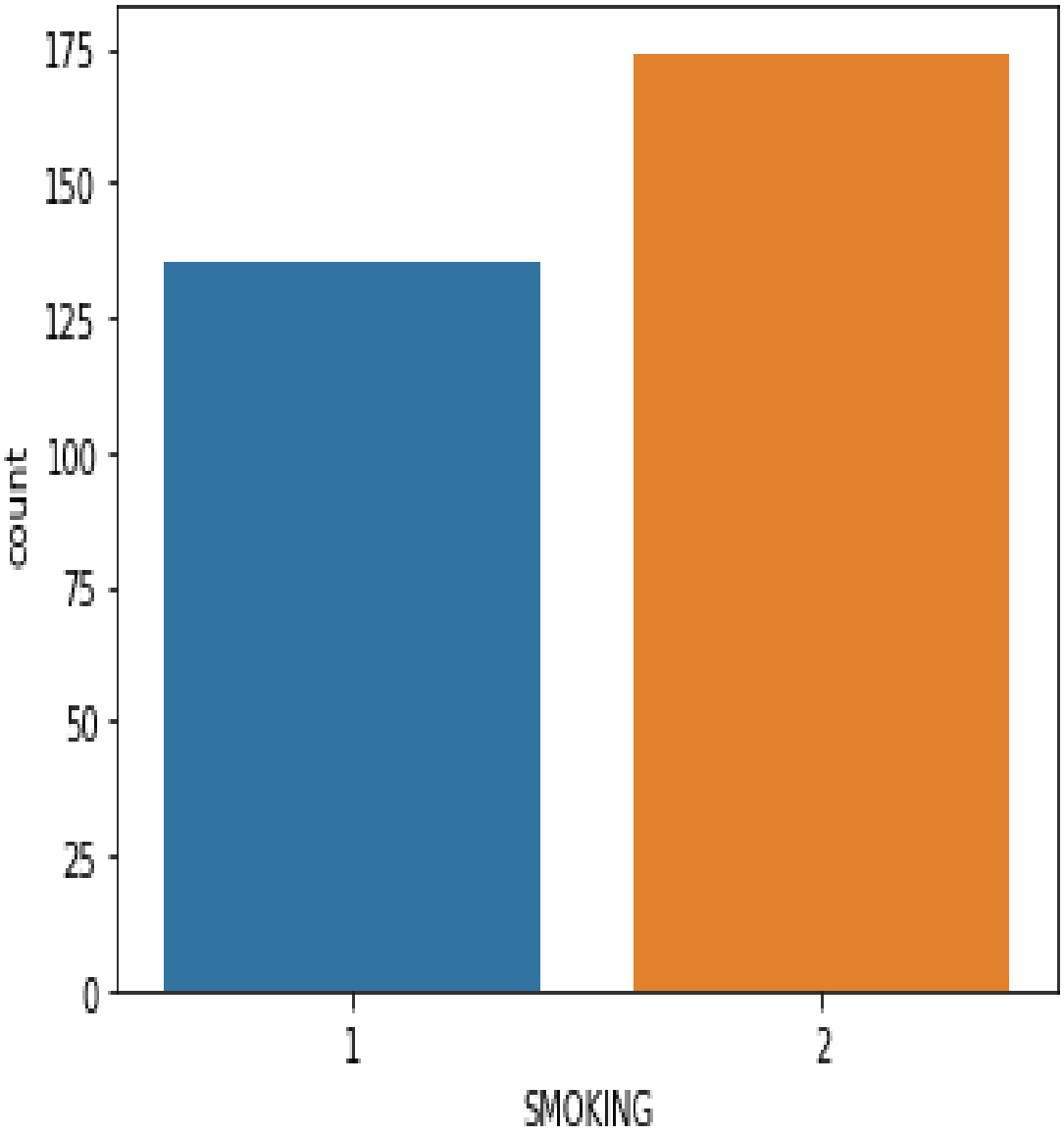


Figure 4 Number of people in dataset who smoking

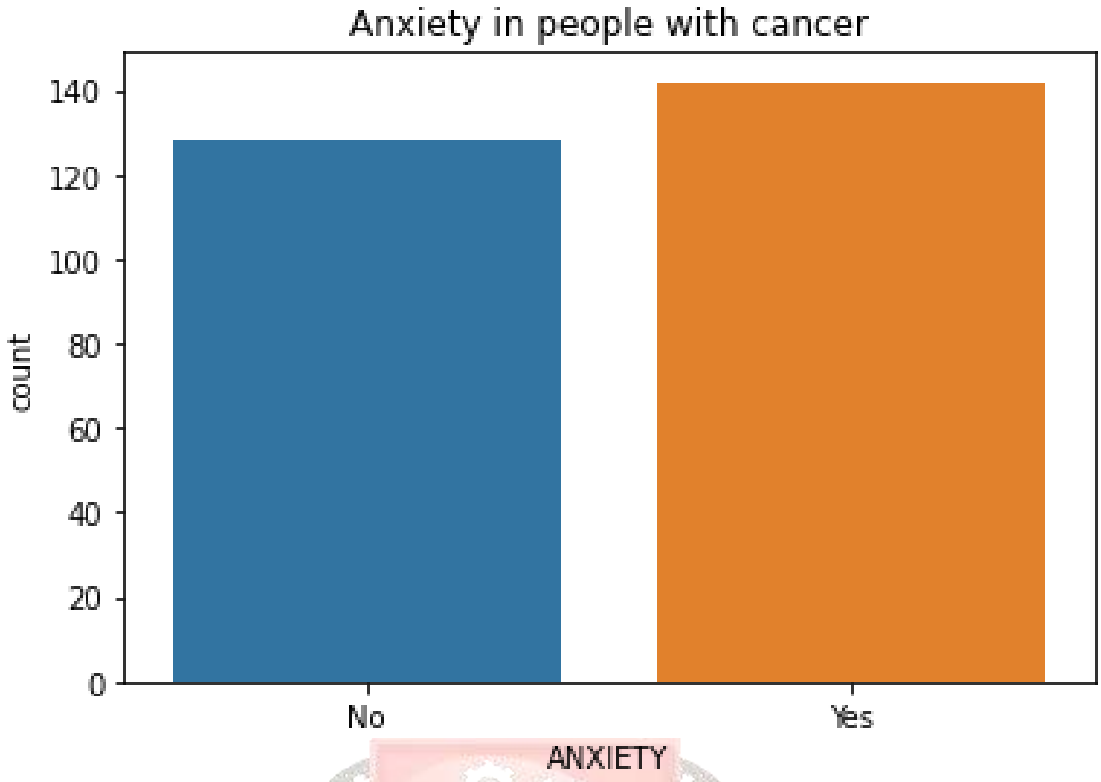


Figure 5 anxiety in people with cancer

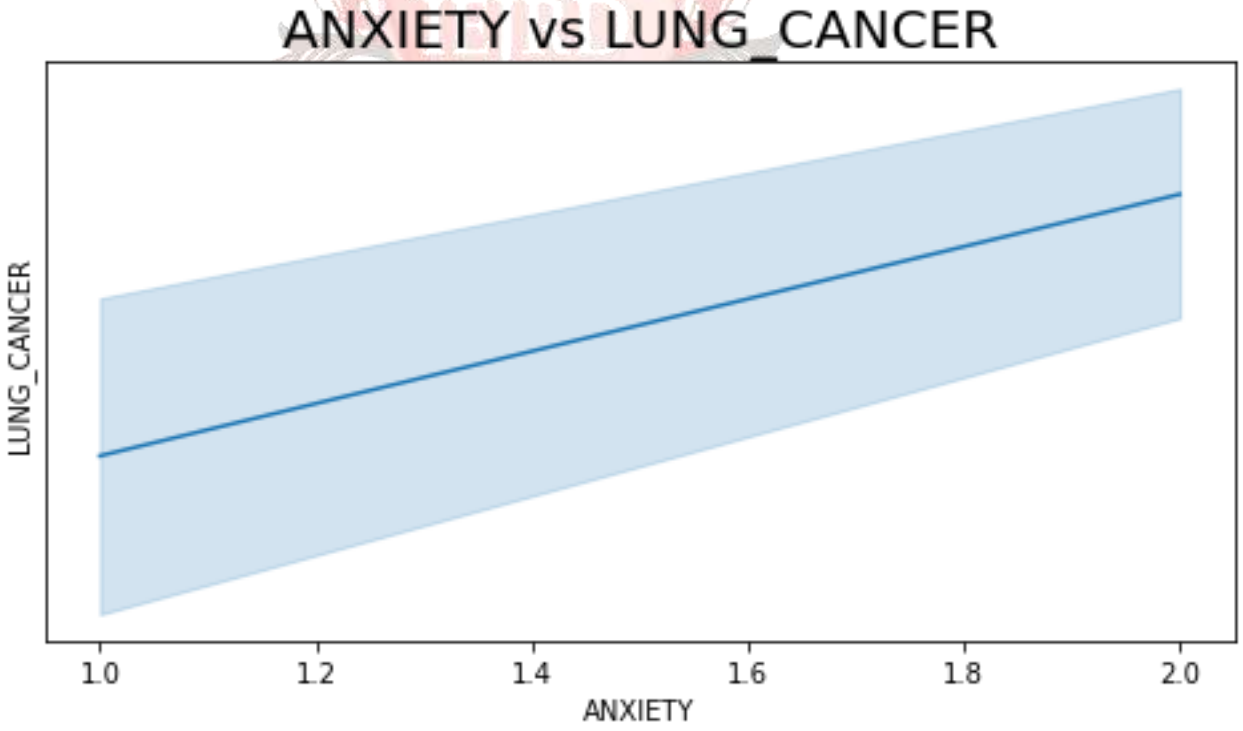


Figure 6 Anxiety vs lung cancer

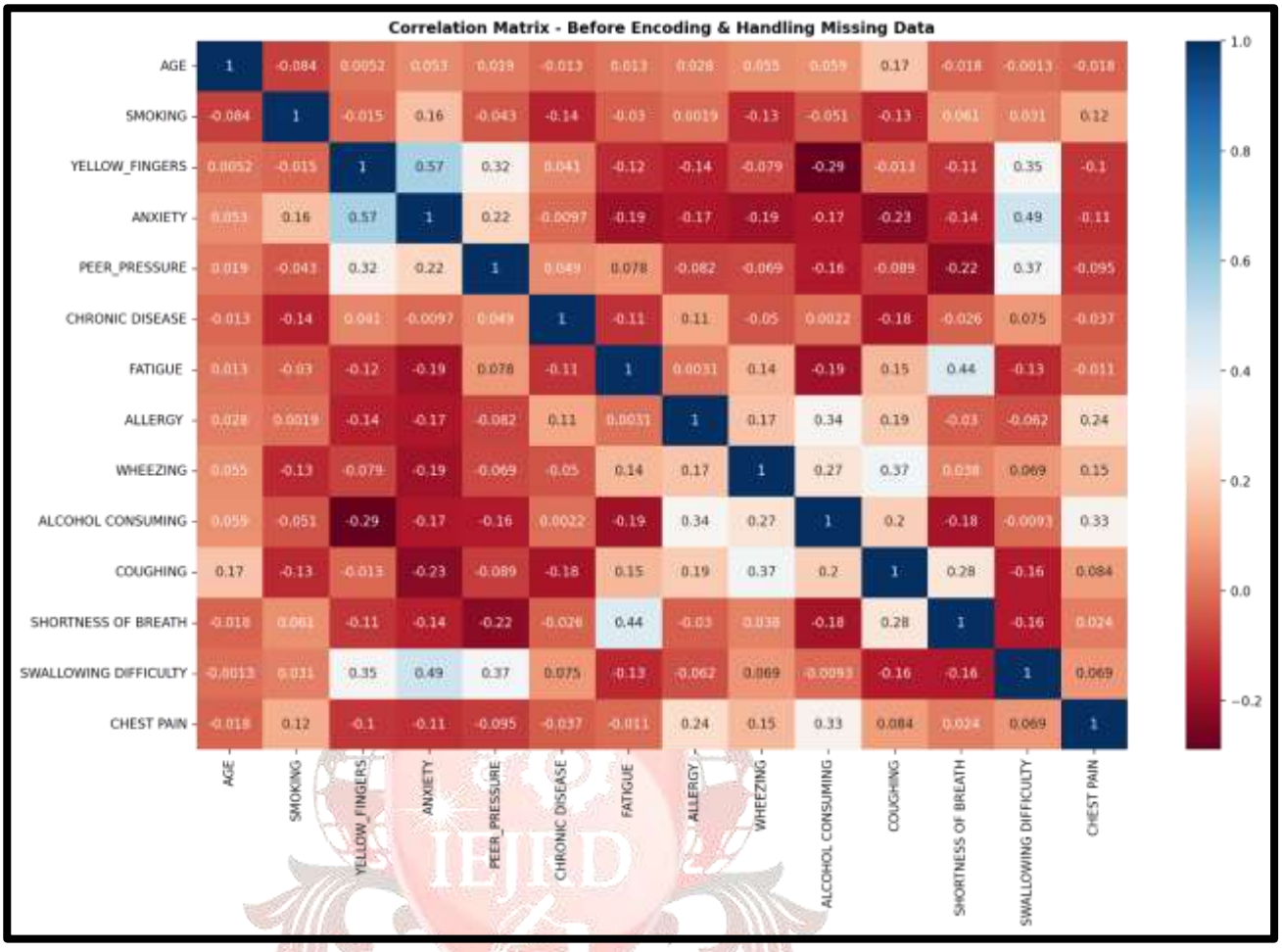


Figure 7 Correlation Matrix before encoding and handling missing data

Table I. Training results for different deep learning models for detecting Lung cancer.

Models	Training Accuracy	Training AUC	Training Recall	Training Loss
CNN	**99.20%**	**99.9%**	**99.40%**	**0.002**
Res-50	**98.56%**	**98.99%**	**94.60%**	**0.046**
Ince. V3	**93.25%**	**95.40%**	**93.23%**	**1.760**
Xce.	**92.10%**	**97.70%**	**92.08%**	**1.350**

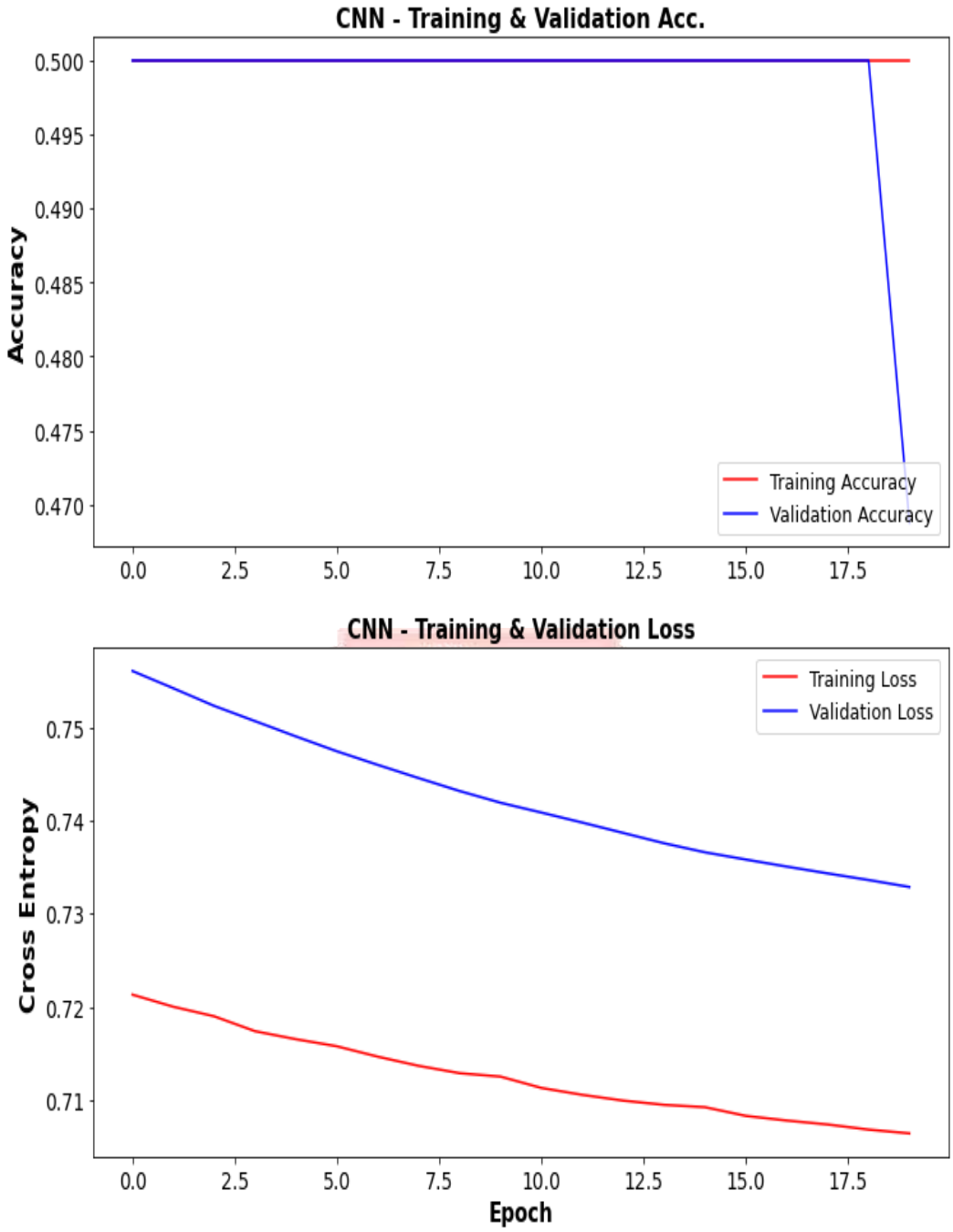


Figure 8 CNN Training and Validation Loss

Figure 4 illustrates the comparison of the model's accuracy, AUC, and loss. Accuracy, AUC, and loss have all been taken into consideration when evaluating the models' performance. Figure 4 shows that compared to other models, CNN had the highest testing accuracy which is 92%. ResNet-50, Inception V3, and Xception achieved testing accuracy of 84.13%, 82.07%, and 82.10% respectively.

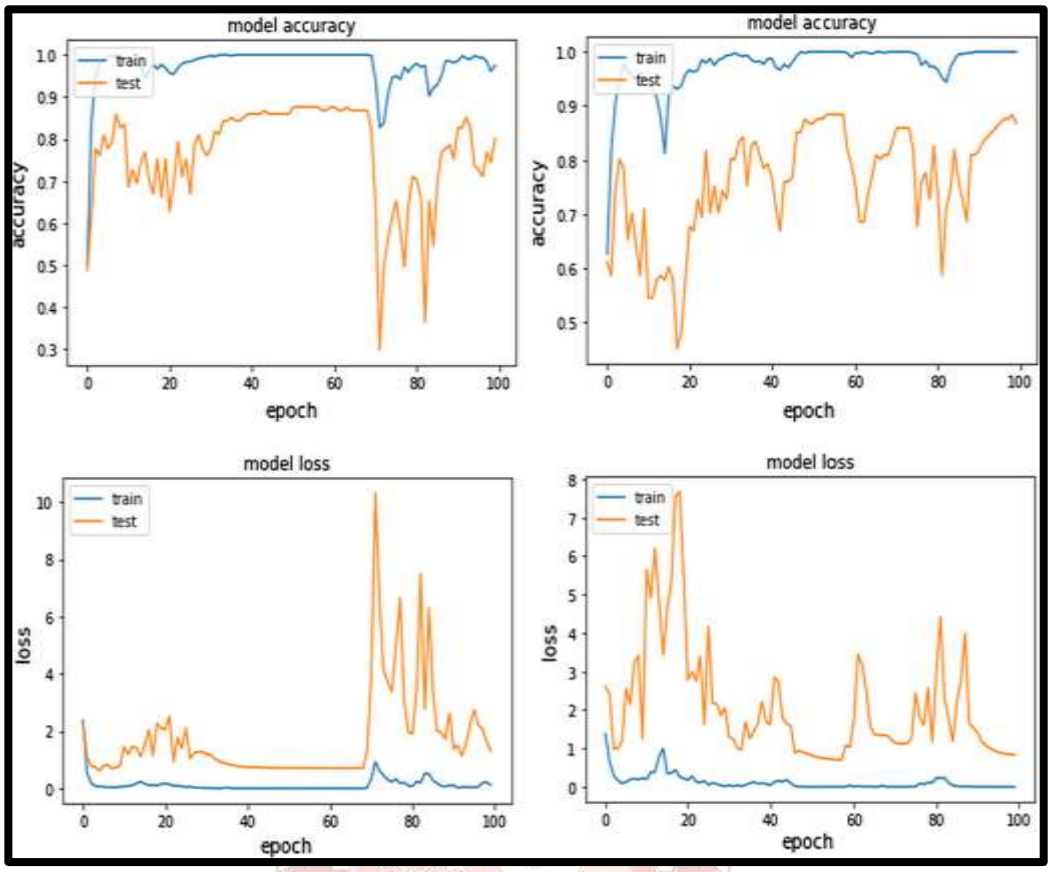
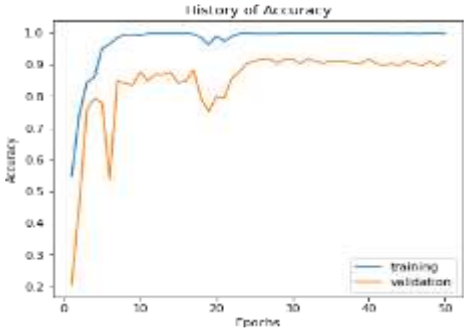


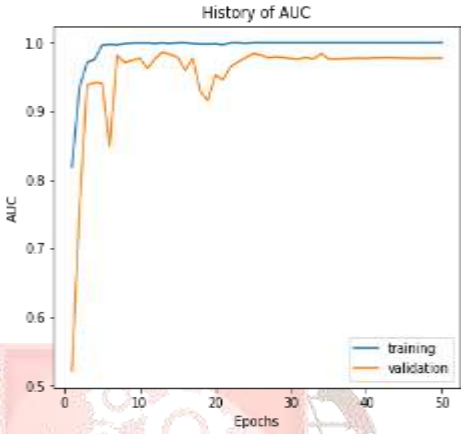
Figure 9 Deep Learning model's performance analysis in terms of accuracy, AUC, and loss.

AUC is also a crucial metric for evaluating the model's performance. AUC determines the model's performance and assesses a model's ability to differentiate between classes. The AUC measures how well the model differentiates between positive and negative classes. The higher the AUC value, the better the model's performance. The value range is 0 to 1, with 0 representing an incorrect test and 1 representing an accurate test. In general, an AUC of 0.5 indicates no discrimination (i.e., the ability to classify lung cancer), 0.7 to 0.8 is considered acceptable, 0.8 to 0.9 is considered great performance, and greater than 0.9 is considered outstanding performance . Based on figure 4, CNN not only achieved the highest testing accuracy only but also achieved the highest test AUC score which is 98.21%. In addition, ResNet-50, Xception, and Inception V3 achieved testing AUC scores of 94.85%, 90%, and 88.5% respectively. However, the loss is another important metric for considering the model's performance. Loss is a number that indicates how inaccurate the model's prediction is at each epoch. If the loss is zero the model's prediction is perfect; otherwise, the greater the loss worse the model's performance. To calculate the loss in the detecting process, we

used the categorical cross-entropy loss function. Categorical cross-entropy is a loss function that is mostly used in multi- class classification tasks . Here, CNN achieved the lowest loss value of 0.328 and ResNet-50 achieved the loss value of

0.598. On the contrary, Xception and Inception V3 models achieved very high loss values which are 8.27 and 15.7 respectively. . From the above analysis, the custom CNN model outperforms other deep learning models in detecting different types of lung cancer using the CT scan image dataset and considered the proposed model. We provided the validation accuracy, validation AUC and loss function curve with respect to training accuracy, training AUC, and training loss in every epoch for CNN in Figures 5, 6, and 7, respectively.





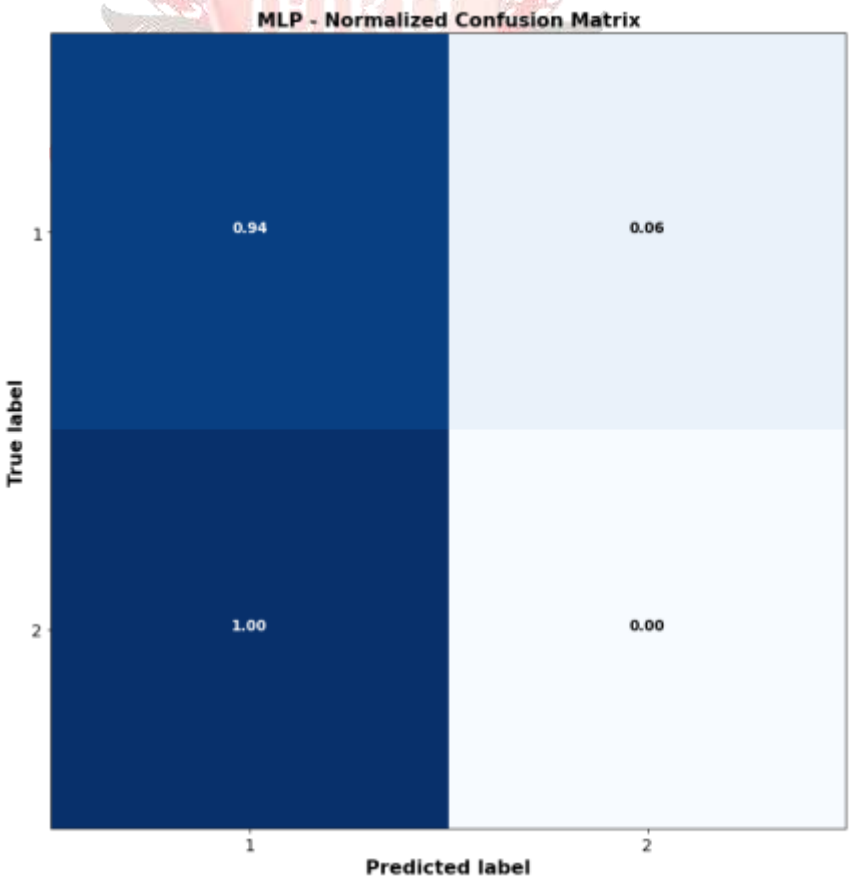


Figure 13 Normalized Confusion Matrix

CONCLUSION AND FUTURE WORK

Figure 1 depicts the death rate for lung cancer, indicating that it is one of the most common and leading tumors globally. Although it cannot be avoided, early detection can help the patient survive longer than predicted. Lung cancer is the leading cause of cancer-related death in North America and other developed countries. Lung cancer is at the top of the priority list since it is typically diagnosed late in its progression. As a result, despite significant progress in recent years, early diagnosis remains unreliable. In this study, we suggested a CNN-based deep learning algorithm for detecting lung cancer early utilizing CT scan pictures. After assessing our deep learning model using CT scan pictures, we discovered that CNN outperformed other models with an accuracy of 92%, AUC of 98.21%, recall of 91.72%, and loss of 0.328. To improve our ability to detect lung cancer early, we may use more datasets and different machine learning and deep learning models in the future.

REFERENCES

[1] Survey of Intelligent Methods for Brain Tumor Detection-IJCSI International Journal of Computer Science, vol. 11, no. 5, September 2014.

[2] Zehra Karhan and Taner Tunç, "Lung Cancer Detection and Classification with Classification Algorithms", IOSR Journal of Computer Engineering (IOSR-JCE), vol. 18, no. 6, pp. 71-77, Nov.-Dec. 2016.

[3] Survey of Intelligent Methods for Brain Tumor Detection-IJCSI International Journal of Computer Science Issues, vol. 11, no. 5, September 2014.

[4] Lung Cancer detection and Classification by using Machine Learning & Multinomial Bayesian-IOSR Journal of Electronics and Communication Engineering (IOSR-JECE), vol. 9, no. 1, pp. 2278-8735, Jan. 2014.

[5] H.R.H Al-Absi, B. B. Samir, K. B. Shaban and S. Sulaiman, "Computer aided diagonosis system based on machine learning techniques for lung cancer", 2012 International Conference on Computer and Information Science(ICCIS), pp. 295-300, 2012.

[6] D. Vinitha, Deepa Gupta and S. Khare, "Exploration of Machine Learning Techniques for Cardiovascular Disease", Applied Medical Informatics, vol. 36, pp. 23-32, 2015.

[7] J. Isaac and S. Harikumar, "Logistic regression within DBMS", Proceedings of the 2016 2nd International Conference on Contemporary Computing and Informatics IC3I 20167918045, pp. 661-666, 2016.

[8] Haralick RM, Shanmugam K, Dinstein I. Textural Features for Image Classification. IEEE Trans Syst Man Cybern Syst 1973;3:610-21. 10.1109/TSMC.1973.4309314

[9] U. Tomar, N. Chakroborty, H. Sharma, and P. Whig, “AI based Smart Agricuture System,” Transactions on Latest Trends in Artificial Intelligence, vol. 2, no. 2.

[10] Lee TS. Image Representation Using 2D Gabor Wavelets. IEEE Trans Pattern Anal Mach Intell 1996;18:1-13

[11] Pickup L, Declerck J, Munden R, et al. MA 14.13 Nodule Size Isn't Everything: Imaging Features Other Than Size Contribute to AI Based Risk Stratification of Solid Nodules. J Thorac Oncol 2017;12:S1860-1. 10.1016/j.jtho.2017.09.582

[12] Aerts HJ, Velazquez ER, Leijenaar RT, et al. Decoding tumour phenotype by noninvasive imaging using a quantitative radiomics approach. Nat Commun 2014;5:4006. 10.1038/ncomms5006

[13] McWilliams A, Tammemagi MC, Mayo JR, et al. Probability of Cancer in Pulmonary Nodules Detected on First Screening CT. N Engl J Med 2013;369:910-9. 10.1056/NEJMoa1214726

[14] Gould MK, Ananth L, Barnett PG, et al. A Clinical Model To Estimate the Pretest Probability of Lung Cancer in Patients With Solitary Pulmonary Nodules. Chest 2007;131:383-8. 10.1378/chest.06-1261

[15] Bartlett EC, Walsh SL, Hardavella G, et al. Interobserver Variation in Characterisation of Incidentally-Detected Pulmonary Nodules: An International, Multicenter Study.

[16] P. Whig, "Exploration of Viral Diseases mortality risk using machine learning," International Journal of Machine Learning for Sustainable Development, vol. 1, no. 1, pp. 11–20, 2019.

[17] R. R. Nadikattu, S. M. Mohammad, and P. Whig, "Novel economical social distancing smart device for covid-19," International Journal of Electrical Engineering and Technology (IJEET).

Appendix

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns

data = pd.read_csv('../input/lung-cancer-detection/survey lung cancer.csv')
df = pd.DataFrame(data)
```

Data Visualization

```
df_lung_cancer = df[df["LUNG_CANCER"]=='YES']
sns.countplot(df_lung_cancer["ANXIETY"])
plt.title("Anxiety in people with cancer")
plt.xticks([0, 1], ('No', 'Yes'))
plt.show()
print("YES = 142")
print("NO = 128")


plt.figure(figsize=(8,4))
sns.lineplot(x=df["ANXIETY"], y=df["LUNG_CANCER"])
plt.title("ANXIETY vs LUNG_CANCER", size=20)
plt.show()
```

Convert categorical value to numeral value

```
colum_cat = ['GENDER','LUNG_CANCER']
for i in colum_cat:
    print('-------------------------')
    print(df[i].value_counts())
    print('-------------------------')

df_clean = df.copy()
for i in colum_cat:
    print(f'Catagory of {i}')
    catlist = df_clean[i].unique()
    for j, val in enumerate(catlist):
        dftobjfinal = df_clean[i].replace({val:j},inplace=True)
```

```
        #print(dftobjfinal)
        print(j,val)
    print('---------------------------------')
```

#Data Visualization

```
plt.figure(figsize=(10,6))
sns.countplot(df["SMOKING"])
plt.title("SMOKING", size=15)
plt.show()
plt.figure(figsize = (16,10), dpi=200)
ax = plt.axes()
sns.heatmap(df.corr(), annot = True, cmap='RdBu', ax=ax)
ax.set_title('Correlation Matrix - Before Encoding & Handling Missing Data', weight='bold')
plt.show()
```

Detect Outliers

```
from sklearn.neighbors import LocalOutlierFactor
clf = LocalOutlierFactor()
y_pred = clf.fit_predict(df_clean)
x_score = clf.negative_outlier_factor_
outlier_score = pd.DataFrame()
outlier_score["score"] = x_score

#threshold
threshold2 = -1.5
filtre2 = outlier_score["score"] < threshold2
outlier_index = outlier_score[filtre2].index.tolist()
g = df_clean.groupby('LUNG_CANCER')
df_balanced = g.apply(lambda x: x.sample(g.size().min()).reset_index(drop=True))
df_balanced = df_balanced.reset_index(drop=True)
x = df_balanced[['SMOKING','PEER_PRESSURE','ALCOHOL CONSUMING']]
y = df_balanced['LUNG_CANCER'].map({'YES': 1, 'NO': 0})
```

Random Forest Classifier

```
from sklearn.ensemble import RandomForestClassifier
r_forest = RandomForestClassifier(criterion = 'entropy', max_depth = 20, n_estimators = 10000)
r_forest.fit(x_train,y_train)
predicted = r_forest.predict(x_test)
score = r_forest.score(x_test, y_test)
rf_score_ = np.mean(score)

print('Accuracy : %.3f' % (rf_score_))
```

Logistic Regression

```
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LogisticRegression
from sklearn.metrics import classification_report
```

```
logmodel= LogisticRegression()
logmodel.fit(x_train,y_train)

predictions=logmodel.predict(x_test)
print(classification_report(y_test,predictions))
```

Deep learning

```
from tensorflow.keras.layers import Conv1D, Dense, Input, MaxPooling1D, Dropout, Flatten
from tensorflow.keras.models import Sequential


model_cnn = Sequential()

#model_cnn.add(Conv1D(32, kernel_size=3, padding='same', activation='relu',
input_shape=(16, 1)))
#model_cnn.add(MaxPooling1D(pool_size=2))

#model_cnn.add(Conv1D(64, kernel_size=3, padding='same', activation='relu'))
#model_cnn.add(MaxPooling1D(pool_size=2))

#model_cnn.add(Conv1D(32, kernel_size=3, padding='same', activation='relu'))
#model_cnn.add(MaxPooling1D(pool_size=2))

#model_cnn.add(Dropout(0.25))
#model_cnn.add(Flatten())
#model_cnn.add(Dense(16, activation='relu'))

#model_cnn.add(Dense(5, activation='softmax'))

model_cnn.add(Dense(3, activation='relu'))
model_cnn.add(Dense(10, activation='relu'))
model_cnn.add(Dense(1, activation='sigmoid'))

model_cnn.compile(loss='binary_crossentropy', optimizer='adam', metrics='accuracy')

#model_cnn.build(input_shape=[3])

import tensorflow as tf
from tensorflow import keras
#es=tf.keras.callbacks.EarlyStopping(
#    min_delta=0.001,
#    patience=10,
#    restore_best_weights=True
#)
h_cnn = model_cnn.fit(x_train, y_train, validation_data=(x_test, y_test), epochs=20)
model_cnn.summary()

acc = h_cnn.history['accuracy']
val_acc = h_cnn.history['val_accuracy']

loss = h_cnn.history['loss']
val_loss = h_cnn.history['val_loss']
```

```
plt.figure(figsize=(12, 12))

plt.subplot(2, 1, 1)
plt.plot(acc, label='Training Accuracy', color='r')
plt.plot(val_acc, label='Validation Accuracy', color='b')
plt.xticks(fontsize=14)
plt.yticks(fontsize=14)
plt.legend(loc='lower right', fontsize=13)
plt.ylabel('Accuracy', fontsize=16, weight='bold')
plt.title('CNN - Training & Validation Acc.', fontsize=16, weight='bold')


plt.subplot(2, 1, 2)
plt.plot(loss, label='Training Loss', color='r')
plt.plot(val_loss, label='Validation Loss', color='b')
plt.xticks(fontsize=14)
plt.yticks(fontsize=14)
plt.legend(loc='upper right', fontsize=13)
plt.ylabel('Cross Entropy', fontsize=16, weight='bold')
plt.title('CNN - Training & Validation Loss', fontsize=15, weight='bold')
plt.xlabel('Epoch', fontsize=15, weight='bold')

plt.show()

import tensorflow as tf
from tensorflow import keras
es=tf.keras.callbacks.EarlyStopping(
    min_delta=0.001,
    patience=10,
    restore_best_weights=True
)
```