[1]Deepak Mane

[2]Anand Magar

[3]Om Khode

[4]Sarvesh Koli

[5]Komal Bhat

[6]Prajwal Korade

Unlocking Machine Learning Model Decisions: A Comparative Analysis of LIME and SHAP for Enhanced Interpretability

Journal of Electrical Systems

Abstract: - XAI is critical for establishing trust and enabling the appropriate development of machine learning models. By offering transparency into how these models make judgements, XAI enables researchers and users to uncover potential biases, admit limits, and eventually enhance the fairness and dependability of AI systems. In this paper, we demonstrates two techniques, LIME and SHAP, used to improve the interpretability of machine learning models. Assessing Explainable AI (XAI) approaches is critical in searching for transparent and interpretable artificial intelligence (AI) models. Explainable AI (XAI) approaches are designed to provide insight into how complex models make decisions. This paper thoroughly analyzes two prominent XAI methods: Shapley Additive explanations (SHAP) and Local Interpretable Model-agnostic Explanations (LIME). This study aims to understand the decision made by a machine learning model and how the model came to that decision. We discuss the approaches and framework of both LIME and SHAP and assess their behavior in predicting the model's outcome.

Keywords: Machine Learning Interpretability, Explainable AI, Decision explanation, Deep Learning, Pattern classification, LIME, SHAP

I. INTRODUCTION

Diagnostics and imaging are essential to healthcare, education, and research. Each diagnosis helps determine the best course of action and how the therapy will proceed. The need for accurate and exact medical diagnosis might be a topic that takes some time to understand and accept. Various machine learning models are used to assess and predict the diagnosis of various illnesses. To aid in diagnosing various medical problems, machine learning models may assess images from medical imaging tests, including MRIs, CT scans, X-rays, and pathology slides. The models may identify various deformities, tumours, pneumonia, and other illnesses. But how accurate are these models in making forecasts? What parameters do these models use to make their final decisions, and how do they decide?

Transparency is required in machine learning models, particularly in medicine [11]. A set of techniques and frameworks known as "explainable AI" are intended to help interpret and evaluate the predictions produced by machine learning models. The two Explainable Artificial Intelligence Techniques (XAI), LIME and SHAP, are assessed and shown in this study, which also aids in comprehending the model's decision-making process. Furthermore, the differences between the methods and their approaches to elucidating machine learning models will be observed. The work contributes to understanding the model's predictions and their potential impact on medical images. Additionally, the study offers future developments in interpretability methodologies and their potential impact on patient treatment. What parameters are these models employing throughout the asking process? Index of paper:

[1] Vishwakarma Institute of Technology, Pune-411037, Maharashtra, India.

[1]deepak.mane@vit.edu

[2]anand.magar@vit.edu

[3]om.khode21@vit.edu

[4]sarvesh.koli21@vit.edu

[5]komal.bhat21@vit.edu

[6]prajwal.korade21@vit.edu

Correspondence author: deepak.mane@vit.edu

The study's objectives are:

- Examine and contrast the two XAI techniques, Local Interpretable Model-agnostic Explanations (LIME) and Shapley Additive explanations (SHAP).
- Boost the machine learning models' interpretability for medical diagnostics and imaging.
- Analyze how well XAI techniques work at identifying difficult model choices.
- In the medical area, emphasize interpretability and openness.
- Analyze SHAP and LIME methodologies and frameworks to understand better how machine learning models decide, particularly when analyzing medical pictures.

II. LITERATURE REVIEW

The paper introduces a method for accurate COVID-19 classification while ensuring understandable results. It combines multiple neural networks and XAI techniques for reliable classification and explanations. This is significant, especially in healthcare. The model becomes transparent by using SHAP values and attention mechanisms, highlighting feature impacts and important data areas. The paper offers an accurate classification model with clear insights into its decisions [1]. The paper presents a method for accurately detecting lung cancer while ensuring interpretability. It uses explainable AI techniques to make the model's decision-making process transparent. Integrating these techniques into lung cancer detection aims to offer accurate predictions and insights into the factors that drive those predictions. This is vital in medical contexts, where understanding prediction reasons is crucial for trust and clinical decisions [2]. The paper introduces an approach that combines accurate lung cancer diagnosis with interpretability. It uses a Convolutional Neural Network (CNN) architecture and techniques like feature visualization and Grad-CAM to identify essential regions in medical images for decision-making. Explanations are generated to provide insights into predictions, which are crucial for medical trust and decisions [3]. The paper employs machine learning algorithms for lung cancer detection and classification, utilizing medical images and clinical data. By training models on these datasets, the approach aids early diagnosis and accurate categorization of lung cancer cases [4].

The paper delves into the theoretical analysis of Explainable AI (XAI) using Case-Based Reasoning (CBR) to explain Neural Networks (NN). It draws insights from a survey of systems that combine Artificial Neural Networks (ANN) with CBR. This research investigates how CBR can provide post-hoc explanations for the decisions made by NNs, contributing to enhanced interpretability in AI systems [5]. The paper explores the concept of twin systems, combining Artificial Neural Networks (ANN) and Case-Based Reasoning (CBR) to enhance explainable AI (XAI). It conducts comparative tests of feature-weighting methods within ANN-CBR Twins, aiming to provide post-hoc explanations for ANN decisions. This research contributes to understanding how integrating CBR into twin systems can facilitate the interpretability of complex ANN-based models [6]. The paper focuses on expert-level evaluations of eXplainable AI (XAI) methods within the medical domain. It assesses the performance and effectiveness of various XAI techniques in providing transparent and interpretable insights into AI models' decisions. By conducting rigorous evaluations, the research aims to advance the understanding and adoption of XAI in medical applications [7]. The paper focuses on analyzing the performance of various Explainable Artificial Intelligence (XAI) methods in the context of medical image classification. The goal is to assess how effectively these methods provide interpretable insights for enhancing understanding and trust in AI-driven medical diagnoses [8].

III. PROPOSED METHODOLOGY

Understanding and assessing the decisions made by sophisticated models, such as deep neural networks, is essential to their application in medicine. This requirement is a result of the necessity to maintain patient safety and foster a feeling of confidence among healthcare practitioners. Because of their intrinsic complexity, deep neural networks can occasionally look opaque, contributing to the perception that decision-making processes are usually opaque[9]. This barrier must be removed to encourage more physician trust and the application of artificial intelligence (AI) models in therapeutic settings. Additionally, medical diagnostics have become more difficult due to the inherent unpredictability of the decisions and predictions made by machine learning models. To help physicians make educated judgments, it becomes necessary to communicate this ambiguity to them appropriately. Given the probabilistic nature of machine learning predictions and the complexity of medical diagnosis, it is imperative to develop specific procedures that specify the degrees of confidence and uncertainty associated with the diagnostic findings. By solving this issue, medical practitioners will be better equipped to decide on treatments based on a thorough comprehension of the underlying uncertainty in the prediction processes, improving the interpretability of AI models. When AI is used in medical settings, bias, and fairness concerns may surface. These concerns may show up as differences in recommended treatment regimens and diagnoses. Giving significant thought to the moral duty of providing impartial and equal healthcare results is essential. Various factors, such as unbalanced training data and algorithmic design decisions, can cause bias in AI models. Acknowledging uneven effects and putting fairness-promoting policies in place are crucial. Incorporating initiatives to improve medical AI models' interpretability, handle ambiguity, and reduce bias might help the healthcare industry provide more equitable, transparent, and dependable diagnostic services. The main objective is to improve patient care, which will contribute to that.

The usefulness of Shapley Additive exPlanations (SHAP) and Local Interpretable Model-agnostic Explanations (LIME) in elucidating the decision-making procedures incorporated into machine learning models is thoroughly evaluated in this research paper [10]. This work is particularly relevant to medical image analysis since certain models are ambiguously "black box." Utilizing LIME and SHAP becomes a strategic approach to lessen the opacity of these models and enhance their interpretability in the context of medical diagnosis, as they may offer localized explanations for model predictions. The study emphasizes the value of techniques such as LIME and SHAP in medical diagnostics, where machine learning predictions inherently involve uncertainty. These methods provide in-depth, context-specific insights into model forecasts, which significantly helps to lessen the problem of ambiguity. With these interpretability tools at their disposal, doctors may evaluate the level of confidence associated with specific diagnostic results by obtaining an in-depth understanding of the logic underlying the model's conclusions. This advanced knowledge encourages a more cautious and sensitive approach to patient care by assisting doctors in making well-informed judgments.

Furthermore, the paper's emphasis on interpretability techniques is essential for addressing bias and injustice in using artificial intelligence (AI) in healthcare. For ethical and fair healthcare practices to be implemented, biases in the decision-making process must be acknowledged and addressed. Due to the interpretability offered by LIME and SHAP, stakeholders can scrutinize the intricacies of model decisions and detect any biases. This understanding guarantees a more ethical and equitable implementation of AI technology in the complex field of medical diagnostics by empowering decision-makers and healthcare professionals to take corrective action. Combining interpretability techniques promotes AI's ethical and responsible use in the healthcare sector and makes machine-learning models more visible. As we delve into the implementation of two well-known interpretability techniques, Local Interpretable Model-agnostic Explanations (LIME) and Shapley Additive explanations (SHAP), across all the different models trained, the first step in the research was to extract Image features from the collected data. Histogram of Oriented Gradients (HOG) is a popular feature descriptor that extracts essential features from an image. HOG divides an image into small cells by first calculating each pixel's gradient magnitude and orientation. Implementing LIME and SHAP is done on three previously trained machine learning models: Decision Tree, Random Forest, and Support Vector Machine, along with a CNN (Convolutional Network Model) model training to understand and examine the outcome of the above models. Each model is trained using HOG-selected features. The models are then evaluated using parameters like accuracy, precision, etc. The medical image dataset is acquired from a trusted, reliable source and preprocessed to handle missing values, outliers, and other untrusted features.

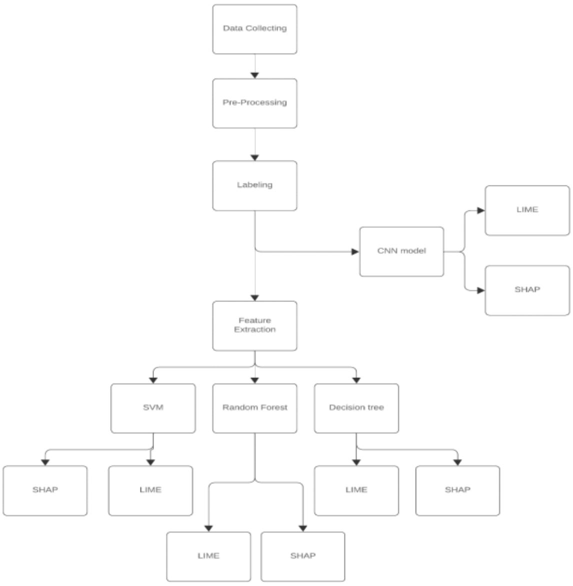


Figure 1 Overall flow diagram of this research.

First, a dataset of about 5860 lung X-ray pictures divided into two categories—normal and pneumonic—was gathered from Kaggle. The dataset was subjected to feature extraction in the initial phase of the research. The HOG (Histogram of Oriented Gradients) method was applied. This was required because simple machine learning (ML) models need numerical inputs in order to function; they cannot act directly on pictorial data.

Following feature extraction, three distinct machine learning models—Support Vector Machine, Decision Tree, and Random Forest—were trained using retrieved characteristics. Every model produced predictions at the end of the training period. The SHAP (SHapley Additive exPlanations) and LIME (Local Interpretable Model-agnostic Explanations) approaches were used on each model to comprehend the logic underlying these predictions. After that, visualization graphs from SHAP and LIME were generated.

In the project's second phase, the picture dataset was used directly instead of feature extraction. Since neural networks can process various input data types, including photographs, CNNs (Convolutional Neural Networks) were used this time to train the NN (Neural Network) model. Likewise, the NN model's predictions were subjected to LIME [12] and SHAP methods. But this time, the X-ray image's superpixels that made the predictions were highlighted, and the output format differed.

3.1Histogram of Oriented Gradients (HOG)

Let's say we have a black-and-white image of 352 pixels(22x16). The pixel value ranges from 0-255, where 0 is black and 255 is white. Humans can easily recognize this image and associate it with its class. However, the way it is stored will determine how computers classify it. Given that there is only one channel for Black and white images when it is stored in 2D matrix format, extracting features from the matrix is simple. In black and white images, the value of each image makes up the feature vector.

Similarly, for Colored images(RGB), we have three channels, and hence three matrices are used. The final matrix is obtained by combining all three channels and taking the mean of each pixel.HOG plots a histogram by computing pixel-wise gradients and orientations. The first step is to ensure that every image is the same size by resizing each to the exact dimensions. In this instance, we resized it to a 64x128 matrix. Assume for now that we have a 64x128 image with 2x2 cell sizes. Figure 1 shows a cell that we have selected, and Figure 2 shows the.

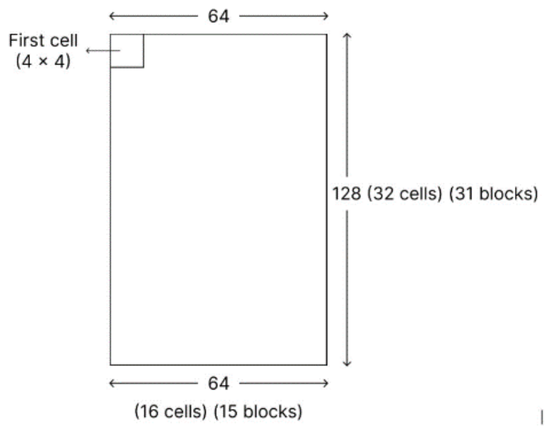
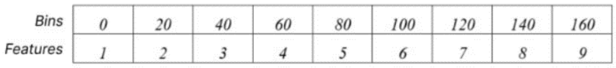


Figure 2 Assumed pixel values for the selected cell.

$$\begin{bmatrix} 121 & 10 & 5 & 4 \\ 3 & 20 & 8 & 9 \\ 27 & 45 & 75 & 80 \\ 36 & 82 & 2 & 4 \end{bmatrix}$$



Bins	0	20	40	60	80	100	120	140	160
Features	1	2	3	4	5	6	7	8	9

Figure3 A cell that we have selected.

The x and y gradients are created at the very beginning. The gradient for the indicated pixel is shown below, and subsequently, gradients for each pixel are calculated, and for the pixel at the edge, padding with 0 is done. The next step involves the calculation of orientation (Slope){Θ=tan-1 (Gy= Gx)}. For this, Θ=tan-1(5/35) ~ 8, and the magnitude is also calculated to get a 1xGmatrix for each cell.

G_x=|8-3|=5

$\{G_x=|P_r-P_l|\}$ (1)

G_y=|45-10|=35

$\{G_x=|P_b-P_t|\}$ (2)

$G_{mag}= \surd(G_x^2) + (G_y^2)$

Where Pr is the pixel to the right of the selected pixel, Pl is the pixel to the left, Pt is the pixel to the top of the selected pixel, and Pb is the pixel to the bottom of the selected pixel.

$G_{mag}= \surd(G_x^2) + (G_y^2) = \surd\ (125 + 1225) = \surd 1350$

$\therefore G_{mag} = 35.07$

Further, Normalization is done while combining multiple cells to form 1 block. Each cell is used as one block, signifying four cells (2x2). Then, a 2x2 kernel will move on to the features, normalizing it. Therefore, since each cell in this instance is 1x9, the normalized matrix created from the cells, √also known as blocks, will measure 1x36.

To find normalized vector, k= √(a1)2 +(a2)2 +(a3)2 +…(a36)2.

Normalized vector =√(a1/k) +(a2/k) +(a3/k) +…. +(a36/k), which will be the output for Block 1, and HOG will calculate the following value for all the blocks. Two matrices comprising values G magnitude and theta are maintained from gradients (Gx & Gy), and the required histogram is created from this.

Note: - Theta may vary from 0-180°. As for the histogram, HOG prepares bins of 20 each.

Figure3 Shows the calculated bins.

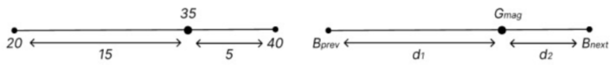


Figure4 Shows the example of weight and deviation from each method.

W_{bprev} = (d_2 x G_{mag})/ (B_{next} - B_{prev}) (3)

W_{bnext}= (d_1 x G_{mag})/(B_{next} - B_{prev}) (4)

3.2 LIME(Local Interpretable Model-Agnostic Explanations)

In LIME, we generate a local approximation of our complex model.Assume that f represents the complex model and that set represents the simple model that will serve as the local model g.(g Ɛ G).G is the set of interpretable models, also known as the family of interpretable model . It is made up of linear models and their variations (like a decision tree).

The first loss term would be:

$L\ (f,g,\pi_x)$ (5)

It means that we seek a model approximation of f about data point x (denoted by pix). Simply put, we use [a] to obtain a good approximation in the vicinity to reduce the complexity of the simple model.

The second last class term would be:

$\Omega(g)$ (6)

It is used to regularize the complexity of simple surrogate models. The final function including both the loss functions will be

$\varepsilon(x)\ = \sum_{g\in G}(f,g,x)\ +\ \Omega(g)$ (7)

In summary, [c] states that we seek the simple model g, which is the argument in argmin, that minimizes the loss term [a] and tries to reduce the complexity of the simple model in [b]. In conclusion, it approximates the complex model in the local area.

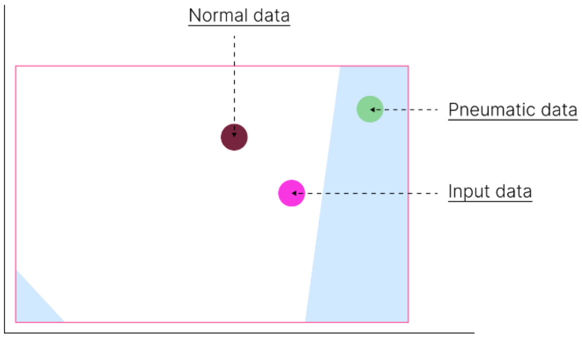


Figure 6

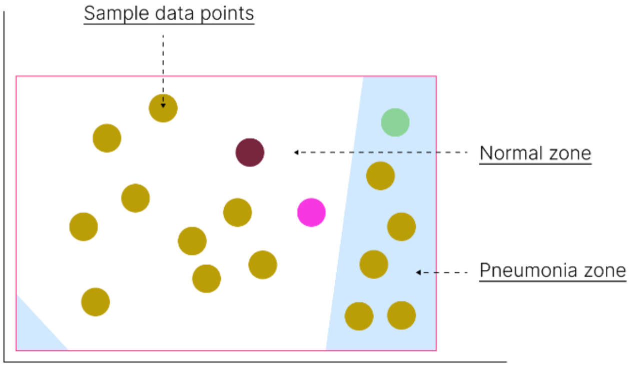


Figure 7 Random sample points which are generated near the input data which are predicted using the complex model f.

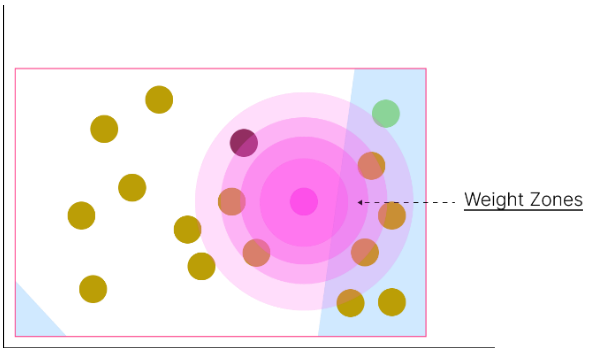


Figure 8 Represents concentric weighted zones that are generated around the input data.

In the first step, random sample points are generated near the input data, which are predicted using the complex model f. We now have a new dataset with features and labels (tables are complex model predictions). A simple model, g, is now used to predict the values of generated samples. In short, a new, simple, linear model is trained nearby.

The first loss function can be defined by [a]

$$\mathrm{L}(f, g, \pi_{\mathrm{x}}) \ = \sum_{z,z\prime \in Z} \pi_{\mathrm{x}}(z)(f(z) - g(z'))^2 \ (8)$$

Here f(z) is the actual label for generated data. g(z`) is the output predicted by the simple model g which is trained on sample data. $\Omega\Pi_{x}$(z) is the weight of the sample, meaning how far or close the sample point is from the generated sample. The closer the point is, the more weight it has. Ω(z) function is used to create as many zero weights are.

3.3SHAP (Shapley Additive exPlanations)

The shapely value for the feature 'i' is being calculated. Let's call the complex model that was trained here 'f' over the input data x. 'x' will be a single row of size 1x16740.

As we have an image feature descriptor of size 1x16740, each image will be stored as 1x16740.

We assume that 'z' represents all possible subsets. Subsets are formed by all possible permutations of different features in this case. Let 'x' represent the simplified data input after feature extraction.

Obtaining output/predictions on a complex model f with and without the target feature (feature to be excluded) on the subset is shown below.

$f_{\mathrm{x}}(\frac{z\prime}{i})$is without target feature ‘I’ and
$f_{\mathrm{x}}(z')$ is with target feature ‘I’

Calculating the individual importance of the target feature.

$$f_{\text{x}}(z') - f_{\text{x}}(\frac{z\prime}{i}) \qquad (9)$$

Equation [d]

The output of [d] can also be called as marginal value. Marginal Value is then calculated for each possible permutation of subset and for each possible target feature.

The weights for each subsets are calculated by

$$\frac{(|z\prime|!(M - |z\prime| - 1)!)}{M!} \qquad (10)$$

Equation [e]

Where M is the total number of feature in particular subset.

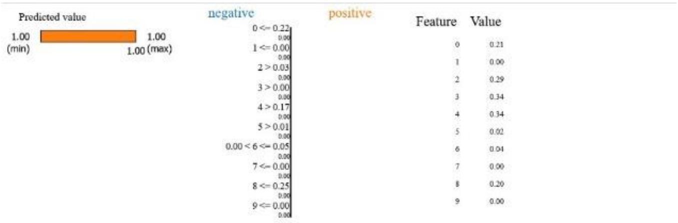
Combining [d] and [e], we get

$$\phi_{\text{i}}(f, x) = \sum_{z' \subseteq x'} \frac{(|z'|!(M - |z'| - 1)!)}{M!} \left[f_{\text{x}}(z') - f_{\text{x}}\left(\frac{z'}{i}\right) \right] \qquad (11)$$

As we make subsets, how do we exclude some features from all of them? Removing some features will change the dimension of the input and will cause an error. Instead of completely removing a specific feature, the actual values are changed to some random values. And, because random data is unpredictable, the output will eventually favor the data that falls under the subset. There are two ways for each feature to be selected in a subset. One will be chosen, and one will not. As a result, if there are n features, there will be 2n sub-sets that consume so much of a computational problem. We can approximate the shapely values using linear regression using Kernal Shap, which samples feature subsets.

IV. EXPERIMENTATION AND RESULTS

RANDOM FOREST LIME:



Feature	Value
0	0.21
1	0.00
2	0.29
3	0.34
4	0.34
5	0.02
6	0.04
7	0.00
8	0.20
9	0.00

Figure 9 represents the readings of the LIME implemented random forest Model.

Negative Class:

- **Intercept:** The intercept for the negative class is 0.22.
- **Feature Weights:**
    - Feature 0 has a weight of 0.00.
    - Feature 1 has a weight of 0.00.
    - Feature 2 has a weight of 0.03.
    - Feature 3 has a weight of 0.00.
    - Feature 4 has a weight of 0.17.
    - Feature 5 has a weight of 0.01.
    - Features 6 through 9 have weights between 0.00 and 0.05.

Positive Class:

- **Intercept:** The intercept for the positive class is 0.21.

- **Feature Values:**
  - Feature 0 has a value of 0.21.
  - Feature 1 has a value of 0.00.
  - Feature 2 has a value of 0.20.
  - Feature 3 has a value of 0.34.
  - Feature 4 has a value of 0.34.
  - Features 5 through 9 have values between 0.02 and 0.25.

Interpretation:

- **Baseline Prediction:** The baseline prediction for the negative class is 0.22, and for the positive class is 0.21.
- **Feature Importance (Negative Class):** Features 2 and 4 seem to have a relatively higher impact on pushing the prediction towards the negative class, as they have non-zero weights.
- **Feature Importance (Positive Class):** Features 3 and 4 seem to have a relatively higher impact on pushing the prediction towards the positive class, as they have higher values.
- **Local Fidelity:** The fidelity of the local models for both classes depends on how well the weights/values approximate the behavior of the underlying black-box model in the vicinity of the instance.

RANDOM FOREST SHAP:

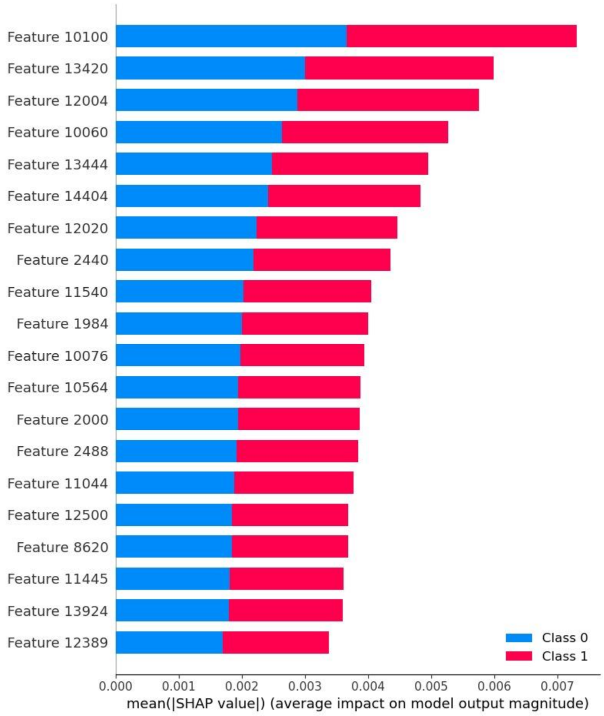


Figure 10 represents the readings from SHAP applied on random forest model.

The provided graph is a SHAP (SHapley Additive exPlanations) summary plot designed to elucidate each feature's average influence on the magnitude of the model output across two distinct classes. The blue line corresponds to the average impact of each feature on the model output magnitude for class 0, while the red line illustrates the average impact for class 1. Features are delineated along the x-axis, and the mean absolute value of SHAP (|SHAP value|) is represented on the y-axis. A higher mean (|SHAP value|) for a specific feature indicates a more pronounced impact on the model output magnitude. For instance, in class 0, feature 10100 exhibits the highest mean (|SHAP value|), signifying its preeminent influence on the model output magnitude for this class. In class 1, feature 13420 attains the highest mean (|SHAP value|), emphasizing its paramount role in determining the model output magnitude for this class. It is imperative to underscore that SHAP values are relative, and direct interpretation of their absolute values is cautioned against. The SHAP summary plot facilitates interpretation by discerning the relative magnitudes of SHAP values for different features. For instance, if feature 10100 has a mean (|SHAP value|) of 0.005, and feature 13420 has a mean (|SHAP value|) of 0.003, feature 10100 is deemed to exert a more substantial influence on the model output magnitude than feature 13420.

Furthermore, the SHAP summary plot aids in identifying the pivotal features crucial for the model's predictive accuracy. Features with the highest mean (|SHAP values|) are identified as pivotal for the model to make precise predictions. In scholarly contexts, the SHAP summary plot emerges as an invaluable tool for comprehending the functioning of a machine learning model and pinpointing features pivotal for accurate predictions.

SVM LIME:

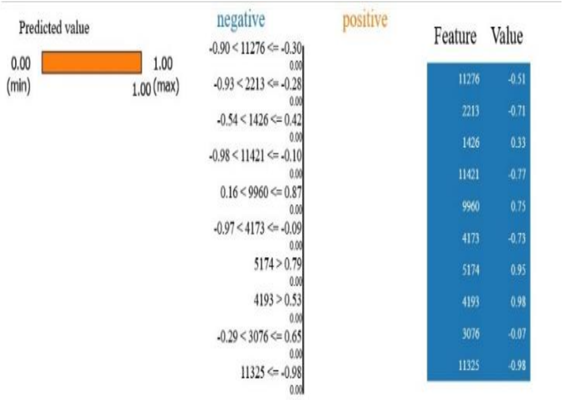


Figure 11 represents the outputs from LIME implemented on SVM model.

Negative Class:

- **Feature Weights:**
  - −0.90<11276≤−0.30−0.90<11276≤−0.30 has a weight of 0.00.
  - −0.93<2213≤−0.28−0.93<2213≤−0.28 has a weight of 0.00.
  - −0.54<1426≤0.42−0.54<1426≤0.42 has a weight of 0.00.
  - −0.98<11421≤−0.10−0.98<11421≤−0.10 has a weight of 0.00.
  - 0.16<9960≤0.870.16<9960≤0.87 has a weight of 0.00.
  - −0.97<4173≤−0.09−0.97<4173≤−0.09 has a weight of 0.00.
  - 5174>0.795174>0.79 has a weight of 0.00.
  - 4193>0.534193>0.53 has a weight of 0.00.
  - −0.29<3076≤0.65−0.29<3076≤0.65 has a weight of 0.00.
  - 11325≤−0.9811325≤−0.98 has a weight of 0.00.

Positive Class:

- **Feature Values:**
  - Feature 0 (11276) has a value of -0.51.
  - Feature 1 (2213) has a value of -0.71.

- Feature 2 (1426) has a value of 0.33.
- Feature 3 (11421) has a value of -0.77.
- Feature 4 (9960) has a value of 0.75.
- Feature 5 (4173) has a value of -0.73.
- Feature 6 (5174) has a value of 0.95.
- Feature 7 (4193) has a value of 0.98.
- Feature 8 (3076) has a value of -0.07.
- Feature 9 (11325) has a value of -0.98.

Interpretation:

- **Baseline Prediction:** The predicted value for the instance falls within the range of 0.00 to 1.00.
- **Feature Importance (Negative Class):** None of the specified ranges for features in the negative class have contributed to the prediction (weights are all 0.00).
- **Feature Importance (Positive Class):** The positive class prediction is influenced by the specific values of features, with higher absolute values having a greater impact.

These outputs suggest that, for the given instance, the positive class prediction is influenced by the values of features rather than specific ranges. The weights associated with the negative class features are all zero, indicating that those feature ranges did not contribute to the negative class prediction. The interpretation is specific to this instance and may vary for other instances.

SVM SHAP:

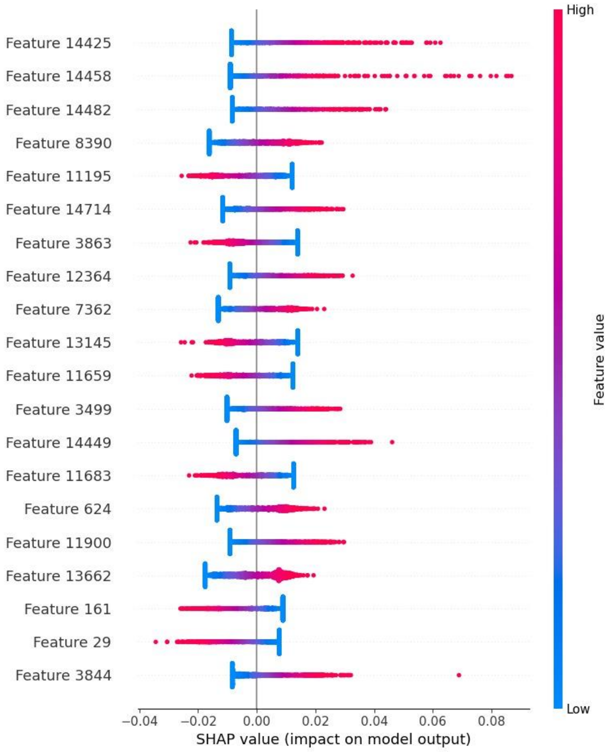


Figure 12 represents the outputs from SHAP implemented on SVM model.

The presented image constitutes a SHAP beeswarm plot, where color is utilized to denote feature values when available. This visualization articulates the impact of individual features on the model output across instances within the dataset.

The x-axis portrays SHAP values for each feature, while the y-axis denotes the respective feature names. Each dot's color corresponds to the value of the feature for the specific instance. A higher SHAP value for a given feature suggests a more pronounced influence on the model output, whereas lower SHAP values indicate a diminished impact. This beeswarm plot is instrumental in identifying influential features for the model output and understanding how variations in feature values contribute to the model's predictions.

Interpreting the plot involves scrutinizing the distribution of SHAP values for each feature. Notably, consistently positive SHAP values for a feature imply a positive impact on the model output, while consistently negative values suggest a negative impact. The SHAP beeswarm plot aids in pinpointing crucial features for accurate predictions, with features exhibiting a broader spread of SHAP values considered most important for the model.

Specific observations from the SHAP beeswarm plot include:

- **Feature 14425:** Evidence of a substantial positive impact on the model output, indicating that instances with higher values of feature 14425 tend to yield higher model outputs.
- **Feature 14458:** Displays a notable negative impact on the model output, suggesting that instances with higher values of feature 14458 tend to yield lower model outputs.
- **Feature 11195:** Manifests a positive impact on the model output for instances with low values of the feature but a negative impact for instances with high values. This underscores that the effect of feature 11195 on the model output is contingent on the feature's value.
- **Feature 11659:** Reflects a negative impact on the model output for instances with low feature values and a positive impact for instances with high feature values. This suggests that the influence of feature 11659 on the model output depends on the feature's value.

In summary, the SHAP beeswarm plot underscores the model's intricate nature, revealing that each feature's impact on the model output is intricately linked to the values of other features within the dataset.

DECISION TREE LIME:

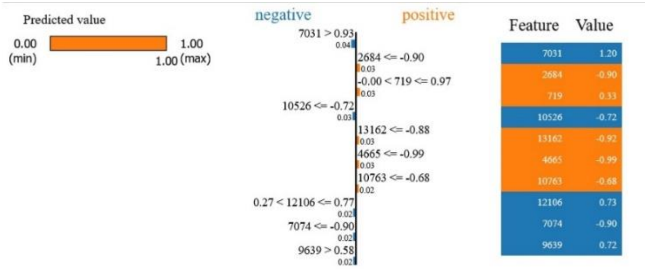


Figure 13 represents the outputs from LIME implemented on Decision Tree model.

Predicted Value:

The predicted value for the instance falls between 0.00 (min) and 1.00 (max).

Negative Class:

- **Feature Values:**
  - 7031>0.937031>0.93 has a weight of 0.04.
  - 2684≤−0.902684≤−0.90 has a weight of 10.03.
  - −0.00<719≤0.97−0.00<719≤0.97 has a weight of 10.03.

Positive Class:

- **Feature Values:**
    - 10526≤−0.7210526≤−0.72 has a weight of 0.03.
    - 0.27<12106≤0.770.27<12106≤0.77 has a weight of 0.02.
    - 7074≤−0.907074≤−0.90 has a weight of 0.02.
    - 9639>0.589639>0.58 has a weight of 0.02.
    - 13162≤−0.8813162≤−0.88 has a weight of 10.03.
    - 4665≤−0.994665≤−0.99 has a weight of 0.03.
    - 10763≤−0.6810763≤−0.68 has a weight of 0.02.

Interpretation:

- **Baseline Prediction:** The predicted value for the instance falls within the range of 0.00 to 1.00.
- **Feature Importance (Negative Class):** Feature 2684 has a substantial negative impact on the negative class prediction and features 7031 and 719 also contribute but with positive weights.
- **Feature Importance (Positive Class):** Features 10526, 12106, 7074, 9639, 4665, and 10763 contribute to the positive class prediction. The specific values of these features, as well as their weights, are listed.

It's important to note that the weights associated with feature values are specific to this instance, and the interpretation is based on the local behavior of the model around this data point. This local interpretation might not be generalizable to the entire dataset or other instances.

DECISION TREE SHAP:

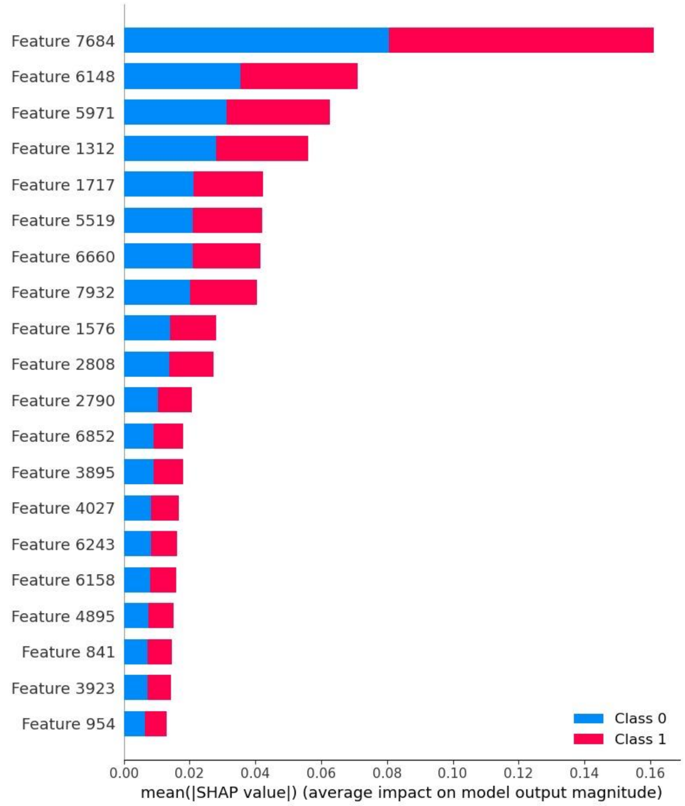


Figure 14 represents the outputs from SHAP implemented on Decision Tree model.

The vertical axis elegantly captures the nomenclature of the features, while the horizontal axis imparts the mean (|SHAP value|), signifying the average influence on the model output magnitude. Notably, the SHAP values are judiciously normalized to a cumulative sum of 1, underscoring their relative rather than absolute interpretability.

The plot notably highlights feature 7684 as the preeminent contributor by our preceding analysis, boasting the highest mean (|SHAP value|) within this class. Additional features, namely 6148, 5971, 1312, and 1717, exhibit noteworthy mean (|SHAP values|), suggesting their substantive impact on the model output magnitude.

Extending beyond mere identification of influential features, the SHAP summary plot enriches our understanding by ascertaining the directional impact of these features. Features registering positive mean (|SHAP values|) contribute positively to the model output magnitude, while those with negative mean (|SHAP values|) exert a discernible negative influence.

From a technical standpoint, SHAP values are derived from a meticulous formula, capturing the average alteration in the model output magnitude consequent to a modification in the value of a specific feature, with all other features held constant. This intricate formulation facilitates a granular comprehension of the nuanced contributions of individual features to the overall model predictions.

In the academic milieu, the SHAP summary plot emerges as an invaluable analytical instrument, elucidating the intricacies of a machine learning model, revealing features pivotal for precision in predictions, and unraveling their respective impacts on the model output magnitude.

Our research uses various techniques to increase the interpretability of the inherent meaning associated with the recovered attributes, aiming to make the intricacies comprehensible to the human observer. We employ the techniques of super-pixels and Local Interpretable Model-agnostic Explanations (LIME) to offer more in-depth justifications for identifying the fundamental features in the context of our study. These methods give profound insight into the feature space by bridging the gap between complex feature representations and human interpretability.

Our study employs many approaches to improve the interpretability of the intrinsic meaning linked to the retrieved characteristics to make the complexities understandable to the human observer. Specifically, we use the methods of super-pixels and Local Interpretable Model-agnostic Explanations (LIME) to provide refined explanations for identifying the underlying characteristics in the framework of our investigation. These techniques provide a more detailed understanding of the feature space by bridging the gap between intricate feature representations and human interpretability.

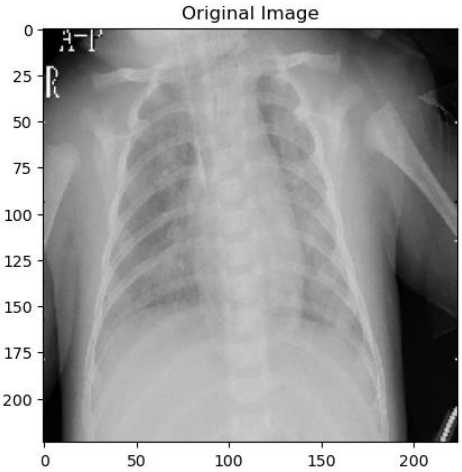


Figure 15 represents a lung X-ray depicting pneumonia symptoms.

The first image is a lung X-ray depicting pneumonia symptoms, and the model correctly identified it as representative of pneumonic circumstances. However, thoroughly examining the relevant super-pixels is necessary to determine the exact reasoning for this categorization. Shapley additive explanations (SHAP)[14] and Local Interpretable Model-agnostic Explanations (LIME)[12] approaches were used to aid in this investigation. The following pictures represent the results obtained by the LIME model. The first picture is the output retrieved using the LIME library's get_image_and_mask () function within the explainer class. The subsequent image clarifies the particular pixels from the first image that substantially impact the final decision made by the model. This thorough method illuminates the key elements influencing the categorization of pneumonia in the X-ray and enables a detailed investigation of the model's decision-making process.

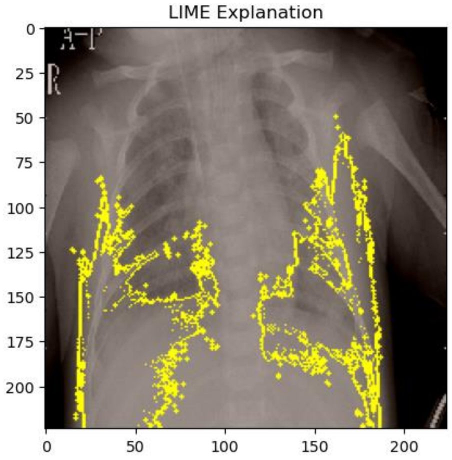


Figure 16 represents the highlighted area, used for the decision-making process from LIME.

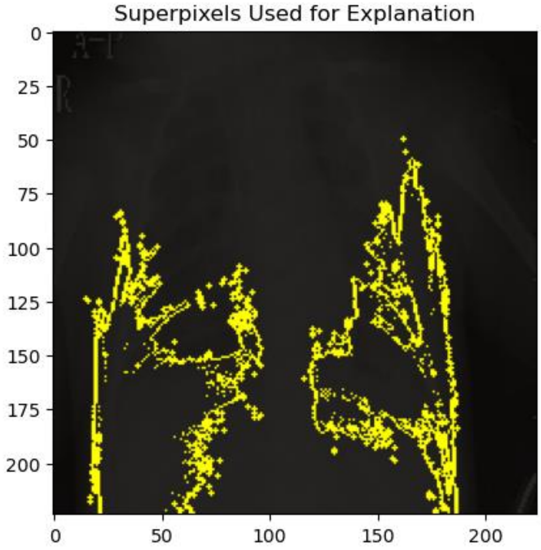


Figure 17 represents the pixels from image are responsible for the specific conclusion.

The first image shows the lung X-ray containing pneumonia; the model has classified it as pneumonitis lungs. However, to get the reason behind that classification, we need to study the super pixels involved in classifying them. For that purpose, we used LIME and SHAP. So, the following images are the outputs from LIME. The first image shows the actual output from LIME using the function get_image_and_mask() from the explainer class from the LIME library. The second image shows which pixels from image are responsible for the specific conclusion.

V. CONCLUSION AND FUTURE WORK

At the end of our study project, the importance of eXplainable Artificial Intelligence (XAI)—especially LIME and SHAP—takes center stage. By carefully combining these methodologies with SVM, random forest, and decision tree models, we were able to get a comprehensive understanding of their underlying complexity. The capacity of LIME to be interpreted locally faithfully and SHAP to provide feature significance through its Shapley values was crucial in revealing the decision-making processes of these models. The focus on XAI increases the openness of our machine-learning models. It provides essential resources to examine and understand their complex workings, enabling us to take a wise and educated approach to using artificial intelligence.

REFERENCES

[1] Shakib Mahmud Dipto, Irfana Afifa, Mostofa Kamal Sagor, Md. Tanzim Reza & Md. Ashraful Alam. Interpretable COVID-19 Classification Leveraging Ensemble Neural Network and XAI.2021.

[2] Md. Sabbir Ahmed,Khondoker Nazia Iqbal,Md. Golam Rabiul Alam.Interpretable Lung Cancer Detection using Explainable AI Method.2021.

[3] Li, Y., Gu, D., Wen, Z., Jiang, F., & Liu, S.. Lung Cancer Detection and Classification using Machine Learning Algorithm.2020.

[4] Meraj Begum Shaikh Ismail.Lung Cancer Detection and Classification using Machine Learning Algorithm.2021.

[5] Mark T. Keane and Eoin M. Kenny. How Case-Based Reasoning Explains Neural Networks: A Theoretical Analysis of XAI Using Post-HocExplanationby-Example from a Survey of ANN-CBR Twin-Systems.2019.

[6] Eoin M. Kenny and Mark T. Keane. Twin-Systems to Explain Artificial Neural Networks using Case-Based Reasoning: Comparative Tests of Feature-Weighting Methods in ANN-CBR Twins for XAI.2019.

[7] Muddamsetty, Satya Mahesh; Jahromi, Mohammad Naser Sabet; Moeslund, Thomas B.. Expert level evaluations for explainable AI (XAI) methods in the medical domain.2021.

[8] Joy Purohit, Ishaan Shivhare, Vinay Jogani and Seema C Shrawne. Analysis of Explainable Artificial Intelligence Methods on Medical Image Classification.2022.

[9] Alejandro Barredo Arrieta, Natalia Díaz-Rodríguez, Javier Del Ser, Adrien Bennetot, Siham Tabik, Alberto Barbado, Salvador Garcia, Sergio Gil-Lopez, Daniel Molina, Richard Benjamins, Raja Chatila, Francisco Herrera,Explainable Artificial Intelligence (XAI): Concepts, taxonomies, opportunities and challenges toward responsible AI, Information Fusion, Volume 58, 2020, Pages 82-115, ISSN 1566-2535, https://doi.org/10.1016/j.inffus.2019.12.012. (https://www.sciencedirect.com/science/article/pii/S1566253519308103)

[10] Evaluation of Explainable Artificial Intelligence: SHAP, LIME, and CAM Hung Truong Thanh Nguyen1, 2 1FSO.QNH.QAI.AIC FPT Software 2Department of Computer Science Frankfurt University of Applied Sciences Frankfurt am Main, Germany HungNTT@fsoft.com.vn Hung Quoc Cao FSO.QNH.QAI.AIC FPT Software Binh Dinh, Vietnam HungCQ3@fsoft.com.vn Khang Vo Thanh Nguyen FSO.QNH.QAI.AIC FPT Software Binh Dinh, VietnamKhangNVT1@fsoft.com.vn Nguyen Dinh Khoi Pham1, 2 1FSO.HCM.FHO.AIC FPT Software 2Northfield Mount Hermon Massachusetts, USA NguyenPDK@fsoft.com.vn

[11] Slack, D., Hilgard, S., Jia, E., Singh, S., &Lakkaraju, H. (2020). Fooling LIME and SHAP. Proceedings of the AAAI/ACM Conference on AI, Ethics, and Society. doi:10.1145/3375627.3375830

[12] What does LIME really see in images? Garreau, Damien and Mardaoui, DinaProceedings of the 38th International Conference on Machine Learning

[13] XAI for Image Captioning using SHAP. DEWI, CHRISTINE; RUNG-CHING CHEN; HUI YU; XIAOYl JIANG

[14] N. Laopracha, T. Thongkrau, K. Sunat, P. Songrum and R. Chamchong, "Improving vehicle detection by adapting parameters of HOG and kernel functions of SVM," 2014 International Computer Science and Engineering Conference (ICSEC), Khon Kaen, Thailand, 2014, pp. 372-377, doi: 10.1109/ICSEC.2014.6978225.