Explanation of Multi-Label Neural Networks with Layer-Wise Relevance Propagation

Peer-reviewed author version

Bello, Marilyn; NAPOLES RUIZ, Gonzalo; VANHOOF, Koen; Garcia, Maria M. & Bello, Rafael (2022) Explanation of Multi-Label Neural Networks with Layer-Wise Relevance Propagation. In: 2022 INTERNATIONAL JOINT CONFERENCE ON NEURAL NETWORKS (IJCNN), IEEE,.

DOI: 10.1109/IJCNN55064.2022.9892239
Handle: http://hdl.handle.net/1942/39310

Explanation of Multi-Label Neural Networks with Layer-Wise Relevance Propagation

1st Marilyn Bello
Andalusian Research Institute
in Data Science & Computational Intelligence
Granada University
Granada, Spain
mbgarcia@ugr.es

2nd Gonzalo Nápoles
Department of Cognitive Science & Artificial Intelligence
Tilburg University
Tilburg, The Netherlands
G.R.Napoles@tilburguniversity.edu

3rd Koen Vanhoof
Faculty of Business Economics
Hasselt University
Hasselt, Belgium
koen.vanhoof@uhasselt.be

4th María M. García
Department of Computer Science
Central University of Las Villas
Santa Clara, Cuba
mmgarcia@uclv.edu.cu

5th Rafael Bello
Department of Computer Science
Central University of Las Villas
Santa Clara, Cuba
rbellop@uclv.edu.cu

Abstract—Neural networks are considered a black-box model as their strength in modeling complex interactions makes its operation almost impossible to explain. Still, neural networks remain very interesting tools as they have shown promising performance in various classification tasks. Layer-wise relevance propagation is a technique that, based on a propagation approach, is able to explain the predictions obtained by a neural network. In this work, we propose four adaptations of this technique to operate on multi-label neural networks. The proposed methods provide new ways of distributing the relevance between the output layer and the preceding ones. The efficacy of these adaptations is demonstrated after an experimental study. The study is carried out based on existing evaluation criteria in the literature that measure the explanation's quality. These methods are applied to a case study in which a neural network is used to detect secondary coinfections in patients infected with SARS-CoV-2. Overall, the proposed methods provide a post-hoc interpretability stage of the results.

Index Terms—explanation, layer-wise relevance propagation, neural networks, multi-label scenarios

I. Introduction

Neural networks have demonstrated impressive performance in complex machine learning tasks [1], [2]. However, due to their multilayer nonlinear structure, they are considered black-box models [3], [4]. As their strength in modeling complex interactions also makes their performance almost impossible to explain. In particular, it is not easy to intuitively and quantitatively understand the result of their inference, i.e., for a single input data point that caused the trained neural model to arrive at a given output. This is especially important in applications such as medicine where the model's confidence must be guaranteed.

One of the most important challenges of Artificial Intelligence (AI) is the construction of effective and interpretable computational models, which has given rise to the so-called Explainable AI (XAI) [5], [6]. XAI is defined as systems with the ability to explain their rationale for making decisions. If an intelligent system resulting from a machine learning process is able to solve a problem and explain its solution, the confidence of its user's increases, which contributes to the credibility of AI [7]. This can be achieved by developing more transparent models or including a post-hoc interpretability stage.

Several approaches [8]–[10] have been proposed to understand and interpret the reasoning embedded in a neural network. Some of them attempt to build more interpretable models from a trained neural network [11], while others supplement the neural network model with an interpretation stage [8]. The Layer-wise relevance propagation (LRP) [8], [12] technique is an example of the latter, which has been shown to provide insightful explanations in the form of the input space's relevance for understanding the classification decisions of feed-forward neural networks. This method allows extracting a significant subset of inputs as the most influential for making a prediction.

LRP explains the classifier's decisions by decomposition, i.e., it redistributes the obtained prediction backward using a local redistribution rule until a relevance score is assigned to each input variable. The local decomposition rule starts from an initial top-level relevance (i.e., output layer relevance) whose associated value is the neuron's activation value representing the decision class. However, this approach, as it is, cannot be applied to scenarios where an input object has a vector of outputs (set of labels) associated with it instead of a single value, such as multi-label classification problems [13].

In this work, an adaptation of the LRP method is proposed to improve the interpretation of the results obtained by a multi-label neural network. For this purpose,

four approaches that redistribute the activation values associated with each label towards the input values are presented. The first approach is based on redistributing all the activation degrees at once. The second redistributes the activation degrees of those neurons activated (predicted labels), while the third does it for each predicted label independently. In this third approach, an explanation is generated for each label in the application domain. The last method performs an aggregation process of the inferred labels' activation degrees, resulting in a granular label from which the redistribution process starts. The explanation of these methods is based on determining the relevant inputs in the inference of a label (3rd approach) or a set of labels (1st, 2nd, and 4th approach). The effectiveness of our methods (in terms of explanation quality) on multi-label scenarios is evaluated based on two different criteria proposed in [9], [10].

These methods are also applied to explain a multi-label neural network's output that detects secondary coinfections in patients infected with SARS-CoV-2. Coinfections associated with the infection SARS-CoV-2 are classified into bacterial infections and fungal infections. A patient may develop one, both, or neither [14]. This case study aims to combine the neural network proposed for its solution with a post-hoc stage, including the four approaches presented in this research. This stage's inclusion made it possible to identify the input variables that influence whether a patient is coinfected with one or more than two infections simultaneously.

The paper is organized as follows. Section II describes the LRP technique while Section III introduces four adaptations of this technique to operate on multi-label neural networks. Section IV presents the experimental study carried out to evaluate the quality of our proposal's explanation. A case study applying the proposed methods is described in Section V. Finally, some concluding remarks are given in Section VI.

II. LRP

LRP [8] is a technique that provides neural networks the ability to explain themselves. It belongs to a class of explanation methods that explain the neural network's output for a specific example, x, giving a score for each input variable to be ranked.

This technique explains the classifier's decisions by decomposition. Mathematically, it redistributes the classifier output $f(x)$ backward using local redistribution rules until it assigns a relevance score R_i to each input variable. This rule fulfills an important property, namely conservation of relevance, defined by Equation (1),

$$\sum_i R_i^0 = \cdots = \sum_j R_j^{T-2} = \sum_k R_k^{T-1} = \cdots = f(x). \quad (1)$$

It ensures that the network's output is fully redistributed to the input domain. In other words, no relevance is lost, and no additional relevance is generated. The relevance scores R_i of each input variable determine how much this variable has contributed to the prediction. If the degree calculated for an input variable is positive ($R_i > 0$), it indicates that it supports that output, but if it is negative ($R_i < 0$), it goes against that prediction.

The local redistribution rule redistributes relevance from layer T to layer $T-1$ in the following way:

$$R_i^{T-1} = \sum_j \frac{a_i w_{ij}}{\sum_i a_i w_{ij} + \epsilon} R_j^T \quad (2)$$

where a_i is the neuron activations at layer $T-1$, R_j is the relevance scores associated to the neurons at layer T and w_{ij} is the weight connecting neuron i to neuron j. A small stabilization term ϵ is added to prevent division by zero.

A downside of this propagation rule (at least if $\epsilon = 0$) is that the denominator may tend to zero if lower-level contributions to neuron j cancel each other out. The numerical instability can be overcome by setting $\epsilon > 0$. However, in that case, the conservation idea is relaxed to gain better numerical properties. A way to achieve exact conservation is by separating the positive and negative activations in the relevance propagation formula, as it does by the $\alpha\beta$-rule given by Equation (3),

$$R_i^{T-1} = \sum_j (\alpha \frac{a_i w_{ij}^+}{\sum_i a_i w_{ij}^+} - \beta \frac{a_i w_{ij}^-}{\sum_i a_i w_{ij}^-}) R_j^T \quad (3)$$

where $()^+$ and $()^-$ are the positive and negative weights connecting i to j, respectively. The parameters α and β are chosen subject to the constraint $\alpha + \beta = 1$ [15].

Intuitively, both rules redistribute relevance proportionally from layer T to each neuron in layer $T-1$ based on two criteria. First, the neuron activation a_i, i.e., more activated neurons receive a larger share of relevance. Secondly, the strength of the connection w_{ij}, i.e., more relevance flows through more prominent connections. However, in the output layer's particular case, the LRP procedure is started on a single neuron whose relevance is set to $R_j = f(x)$. This is not applicable in all scenarios, specifically in multi-label scenarios.

III. Extending LRP to Multi-label Neural Networks

In multi-label scenarios, each object has associated a vector of outputs instead of being associated with a single value [13], [16]. Let us suppose that $\mathcal{U}$ is an N-dimensional object space called the universe, and $\mathcal{L} = \{l_1, l_2, \ldots, l_K\}$ denotes the label space with K possible class labels. The task of multi-label learning is to learn a function $f : \mathcal{U} \longrightarrow 2^{\mathcal{L}}$ from the multi-label training set, where $x \in \mathcal{U}$ is a M-dimensional attribute vector and $\mathcal{L}_i \subseteq \mathcal{L}$ is the set of labels associated with x.

Then, each label $l \in \mathcal{L}$ is mapped by a neuron in the neural network's output layer. Therefore, it is necessary to define how to redistribute the classifier's output $f(x)$. For

this purpose, the following four approaches are proposed. The difference between them lies in how the relevance values are propagated from the output layer (T) to its preceding layer ($T-1$) and the initial top-level relevances in the redistribution process.

A. LRPmlV1

The idea of this approach is to propagate backward the activation values of all output neurons. This means that the initial top-level relevance is defined as $\{R_1 = a_1^{(T)}, R_2 = a_2^{(T)}, \ldots, R_K = a_K^{(T)}\}$. The redistribution rule that distributes the relevance in the i-th neurons of layer $T-1$ is defined by the Equation (4),

$$R_i^{T-1} = \sum_{1 \leq j \leq K} (\alpha \frac{a_i w_{ij}^+}{\sum_i a_i w_{ij}^+} - \beta \frac{a_i w_{ij}^-}{\sum_i a_i w_{ij}^-}) R_j^T. \quad (4)$$

Figure 1 shows the process of relevance redistribution of LRPmlV1 for a sample multi-label neural network. This neural network has four layers: an input layer, two hidden layers, and an output layer. The input layer has M neurons, one for each attribute in $\{f_1, f_2, \ldots, f_M\}$, and the output layer has K labels, one for each label in $\{l_1, l_2, \ldots, l_K\}$. The input is first propagated forward through the network. The last layer activations $\{a_1^{(4)}, a_2^{(4)}, \ldots, a_K^{(4)}\}$ are set as the relevance scores for the last layer and used as the base for relevance redistribution. Using the redistribution rules in Equation (3) and (4), relevance is redistributed back along the network, layer by layer, until the input layer relevance scores $\{R_1^{(1)}, R_2^{(1)}, \ldots, R_M^{(1)}\}$ are obtained.

B. LRPmlV2

This approach aims to propagate backward only the activation values of those output neurons whose activation value is greater than a threshold (i.e., in the output layer, only the neurons' activations corresponding to the inferred labels for the input are considered). The initial top-level relevance is defined as $\{R_j = a_j^{(T)} : a_j^{(T)} > \xi\}$ where $j = 1, \ldots, K$. The redistribution rule that distributes the relevance in the i-th neurons of layer $T-1$ is defined by the Equation (5),

$$R_i^{T-1} = \sum_{\substack{1 \leq j \leq K \\ a_j^{(T)} > \xi}} (\alpha \frac{a_i w_{ij}^+}{\sum_i a_i w_{ij}^+} - \beta \frac{a_i w_{ij}^-}{\sum_i a_i w_{ij}^-}) R_j^T. \quad (5)$$

Figure 2 shows the relevance redistribution process of LRPmlV2 for an input labeled by the neural model with (l_1) and (l_K) as the activation values associated with those labels exceeded the ξ threshold.

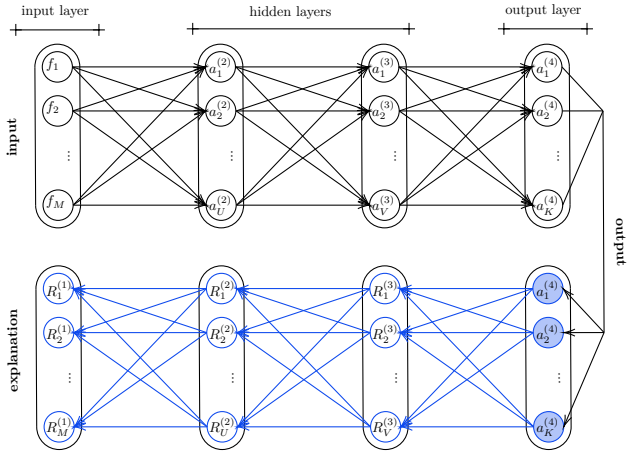

Figure 1: Relevance flow across a multi-label neural network after applying LRPmlV1. The figure's top part represents the inference process from the $\{f_1^{(1)}, f_2^{(1)}, \ldots, f_M^{(1)}\}$ activation values of the input layer to the $\{a_1^{(4)}, a_2^{(4)}, \ldots, a_K^{(4)}\}$ activation values in the output layer. In contrast, the bottom part represents the explanation process from the $\{a_1^{(4)}, a_2^{(4)}, \ldots, a_K^{(4)}\}$ activation values of the output layer to the $\{R_1^{(1)}, R_2^{(1)}, \ldots, R_M^{(1)}\}$ activation values of the input layer.

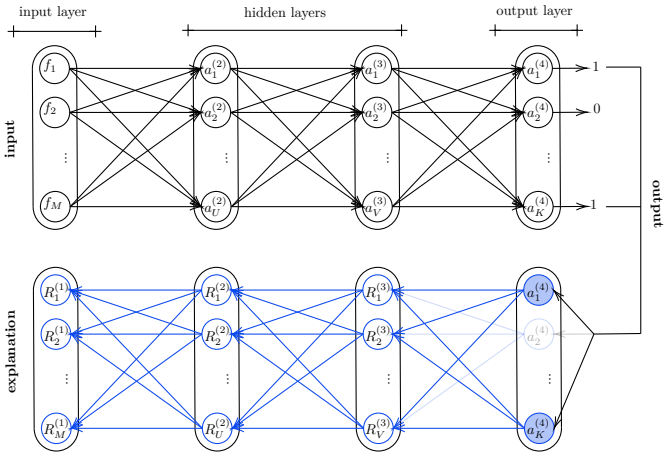

Figure 2: Relevance flow across a multi-label neural network after applying LRPmlV2. The figure's top part represents the inference process from the $\{f_1^{(1)}, f_2^{(1)}, \ldots, f_M^{(1)}\}$ activation values of the input layer to the $\{a_1^{(4)}, a_2^{(4)}, \ldots, a_K^{(4)}\}$ activation values in the output layer. In contrast, the bottom part represents the explanation process from the $\{a_1^{(4)}, a_K^{(4)}\}$ activation values of the output layer to the $\{R_1^{(1)}, R_2^{(1)}, \ldots, R_M^{(1)}\}$ activation values of the input layer.

C. LRPmlV3

The explanations given by this approach differ from the others proposed. In this case, the intuition is to provide an independent explanation for each label, i.e. to find to what

extent each input variable contributed to the prediction of a given label.

In this sense, the idea is to propagate the neuron's activation value associated with the j-th label to be explained. Accordingly, the redistribution rule that distributes the relevance in the i-th neurons of layer $T-1$ is defined by the Equation (6),

$$R_i^{T-1} = (\alpha \frac{a_i w_{ij}^+}{\sum_i a_i w_{ij}^+} - \beta \frac{a_i w_{ij}^-}{\sum_i a_i w_{ij}^-}) R_j^T. \quad (6)$$

Figure 3 shows the relevance redistribution process of LRPmlV3 when the initial top-level relevance is $R_1 = a_1^{(4)}$. The $\{R_1^{(1)}, R_2^{(1)}, \ldots, R_M^{(1)}\}$ values indicate the relevance of the $\{f_1^{(1)}, f_2^{(1)}, \ldots, f_M^{(1)}\}$ attributes in the inference of the l_1 label.

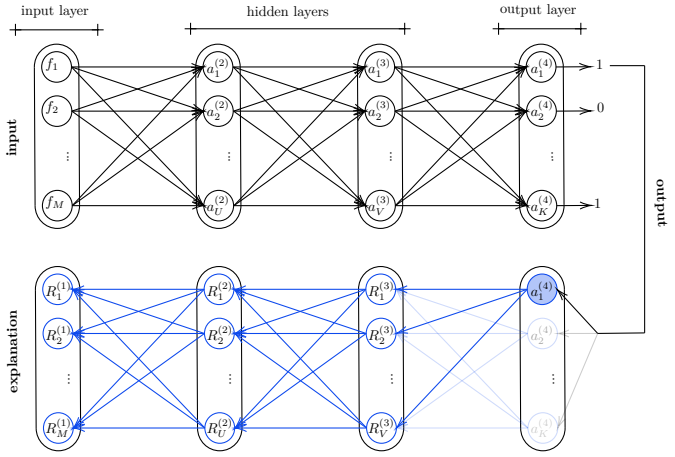

Figure 3: Relevance flow across a multi-label neural network after applying LRPmlV3. The figure's top part represents the inference process from the $\{f_1^{(1)}, f_2^{(1)}, \ldots, f_M^{(1)}\}$ activation values of the input layer to the $\{a_1^{(4)}, a_2^{(4)}, \ldots, a_K^{(4)}\}$ activation values in the output layer. In contrast, the bottom part represents the explanation process from the $\{a_1^{(4)}\}$ activation values of the output layer to the $\{R_1^{(1)}, R_2^{(1)}, \ldots, R_M^{(1)}\}$ activation values of the input layer.

D. LRPmlV4

This approach aims to turn the multi-label neural network into a single-label network, but only at the time of explanation. The idea is to determine the inputs that allowed inferring that combination of labels, aggregating all the corresponding neurons' activation values so that the relevance value to be redistributed backward is higher.

In this way, the relevance redistribution process is performed similarly to the classical LRP in terms of the initial top-level relevance since the decomposition process is based on a single output value. The idea is to build a neural granule with all the neurons activated for a given input. After building the granule, an aggregation process of the activation values for each neuron belonging to the granule is carried out. In this way, the relevance of the granule is $R_{g_r} = \bigoplus_{j=1,\ldots,K} a_j^{(T)} : a_j^{(T)} > \xi$. Accordingly, the redistribution rule that distributes the relevance in the i-th neurons of layer $T-1$ is defined by the Equation (7),

$$R_i^{T-1} = (\alpha \frac{a_i w_{ig_r}^+}{\sum_i a_i w_{ig_r}^+} - \beta \frac{a_i w_{ig_r}^-}{\sum_i a_i w_{ig_r}^-}) R_{g_r}. \quad (7)$$

In addition, all connections between the $T-1$ layer and the T layer are removed, and new connections between the $T-1$ layer and the T' layer (i.e., layer associated with the granular neuron) are made. For example (see Figure 2), if neuron $n_1^{(3)}$ is connected to $n_1^{(4)}$ and $n_1^{(K)}$, with a weight $w_{11}^{(3)}$ and $w_{1K}^{(3)}$, respectively, then a connection is set between $n_1^{(3)}$ and n_{g_r} whose associated weight is $w_{1g_r}^{(3)} = w_{11}^{(3)} \oplus w_{1K}^{(3)}$. Figure 2 shows the relevance redistribution process of LRPmlV4 for an input labeled with the labels (l_1) and (l_K), as $a_1^{(4)} > \xi$ and $a_K^{(4)} > \xi$. In this case, the initial top-level relevance is $R_{g_r} = a_1^{(4)} \oplus a_K^{(4)}$.

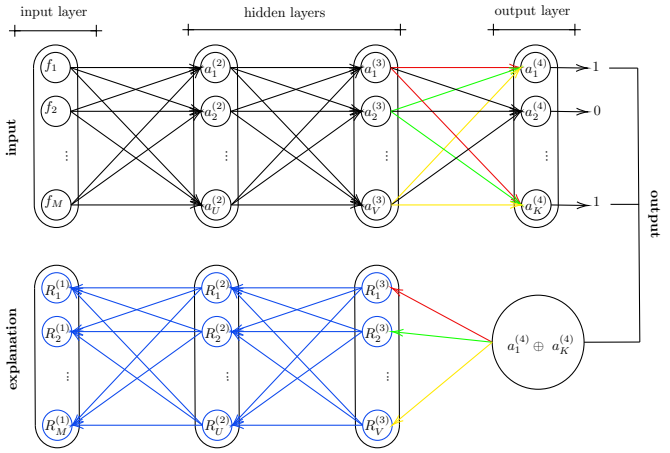

Figure 4: Relevance flow across a multi-label neural network after applying LRPmlV4. The figure's top part represents the inference process from the $\{f_1^{(1)}, f_2^{(1)}, \ldots, f_M^{(1)}\}$ activation values of the input layer to the $\{a_1^{(4)}, a_2^{(4)}, \ldots, a_K^{(4)}\}$ activation values in the output layer (T). In contrast, the bottom part represents the explanation process from the $a_1^{(4)} \oplus a_K^{(4)}$ activation values of the T' layer to the $\{R_1^{(1)}, R_2^{(1)}, \ldots, R_M^{(1)}\}$ activation values of the input layer.

This method's advantage over LRPmlV1 and LRPmlV2 is that it achieves a relevance redistribution process similar to that obtained by classical LRP since it compacts all the relevance information from the output neurons into a single neuron. In this way, the information received by the input layer is more compact, which is difficult to achieve, for example, in datasets with many labels.

IV. Numerical Experiments and Discussion

This section evaluates the quality of the explanations of the four LRP methods for a multi-label scenario. To do this, we first describe the multi-label datasets involved in this study and detail the multi-label neural model employed. Then, we discuss results from two different evaluation criteria existing in the literature.

A. Characterization of datasets

We use five multi-label datasets taken from the RUMDR repository [17]. In these problems (see Table I), the number of objects ranges from 502 to 43,807, the number of attributes goes from 72 to 294, and the number of labels from 6 to 174. More details about these datasets are given next:

- cal500 [18]: It is a music dataset, composed by 502 songs. Each one was manually annotated by at least three human annotators, who employ a vocabulary of 174 tags concerning to semantic concepts. These tags span 6 semantic categories: instrumentation, vocal characteristics, genres, emotions, acoustic quality of the song, and usage terms.
- emotions [19]: It is a small dataset to classify music into emotions that it evokes according to the Tellegen-Watson-Clark model of mood: amazed-suprised, happy-pleased, relaxing-calm, quiet-still, sad-lonely, and angry-aggresive. It consists of 593 songs with 6 classes.
- mediamill [20]: It is a multimedia dataset for generic video indexing, which was extracted tom the TRECVID 2005/2006 benchmark. This dataset contains 85 hours of international broadcast news data categorized into 100 labels, and each video instance is represented as a 120-dimensional feature vector of numeric features.
- scene [21]: It is an image dataset that contains 2407 images, annotated in up to 6 classes: beach, sunset, fall foliage, field, mountain, and urban. Each image is described with 294 visual numeric features corresponding to spatial color moments in the LUV space.
- yeast [22]: This dataset contains micro-array expressions and phylogenetic profiles for 2417 yeast genes. Each gen is annotated with a subset of 14 functional categories (e.g., metabolism, energy, etc.) of the functional catalog's top level.

Table I: Characterization of datasets.

	Objects	Attributes	Labels
cal500	502	68	174
emotions	593	72	6
mediamill	43807	120	101
scene	2407	294	6
yeast	2417	103	14

B. Multi-label neural networks

The proposed architecture involves a fully-connected neural network with four layers: an input layer, two hidden layers, and an output layer. The number of hidden neurons is equal to $2 \times M$ and $2 \times K$, where M and K are the numbers of attributes and labels of the problem. This model operates with scaled exponential linear units [23].

Also, we adopted a squared hinge loss function to increase the margins between positive and negative labels in terms of the learning algorithm [24]. The weights associated with the multi-layer networks are adjusted using the Adam optimization algorithm with the number of epochs set to 200.

On the other hand, Hamming Loss (HL) in Equation (8), is adopted to measure the performance of the multi-label neural model. HL is probably the most widely used performance metric in multi-label scenarios that quantifies the proportion of incorrectly predicted labels [13]. The closer its value is to zero, the more accurate the model is.

$$HL = \frac{1}{N}\frac{1}{K}\sum_{i=1}^{N} |\mathcal{L}_i \Delta Y_i| \tag{8}$$

where Δ operator returns the symmetric difference between $\mathcal{L}_i$ (the real label set of the ith object) and Y_i (the predicted one).

C. Evaluating the quality of explanations

The authors of [3] and [9] proposed an explanation quality criterion based on perturbation analysis. They state that: The perturbation of input variables, which are highly important for the prediction, leads to a steeper decline of the prediction score than the perturbation of input dimensions, which are of lesser importance.

The proposed explanation methods provide a score for each input variable. Thus, according to this relevance score, the input variables can be sorted, obtaining a ranking of attributes. Therefore, it is possible to iteratively perturb input variables (starting from the most relevant ones) and track the prediction score after every perturbation step. The decrease in prediction accuracy (i.e., the increase in the HL value) can be used as an objective measure of the explanation's quality since a large increase indicates that the explanation method was successful in identifying the truly relevant input variables.

Figures 5, 6, and 7 show the HL value when the attributes with the highest relevance value according to the LRPmlV1, LRPmlV2, and LRPmlV4 algorithms are perturbed. This perturbation is based on replacing the input values with random values in the application domain using a uniform distribution. We set the threshold $\xi = 0$, although other values are possible. Later, we will study the effect of this parameter on the performance of the LRPmlV2 method.

These figures illustrate an increase in the HL value when the most relevant attribute (first in the ranking)

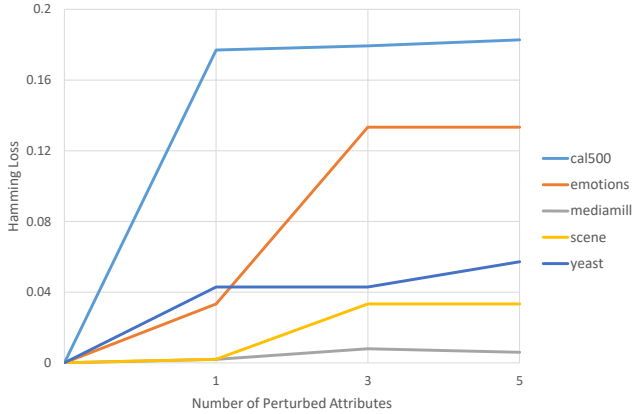

Figure 5: Error of classification (in terms of Hamming Loss) when the values of the five most relevant attributes, according to the ranking resulting from applying LRPmlV1, are replaced by a random value in their application domain.

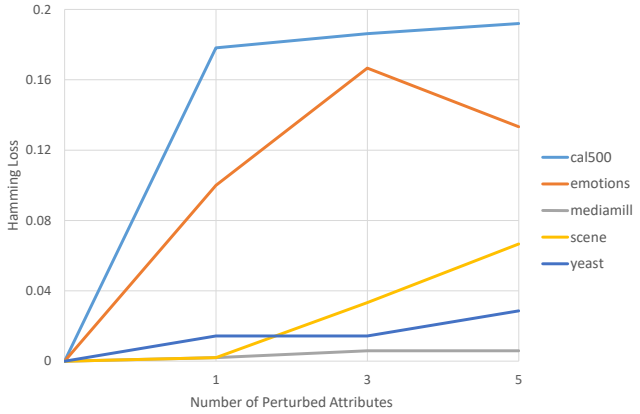

Figure 6: Error of classification (in terms of Hamming Loss) when the values of the five most relevant attributes, according to the ranking resulting from applying LRPmlV2, are replaced by a random value in their application domain.

is perturbed, particularly in the cal500 dataset when the LRPmlV4 algorithm is employed. Moreover, in most cases, as the number of perturbed attributes increases, so does the HL value. However, in the mediamill dataset, more attributes need to be perturbed for this increase to be evident. This is because mediamill has several attributes (more than five) with a high relevance value that influence the result.

Figure 8 shows a similar experiment to the previous ones but based on the relevance ranking obtained by LRPmlV3 for a particular object (in the emotions dataset) when the activation value of the l_3 label is propagated. In this case, it is explored to what extent the perturbation of the most relevant attribute affects a label's value. Note in the figure how after the perturbation, the object that should

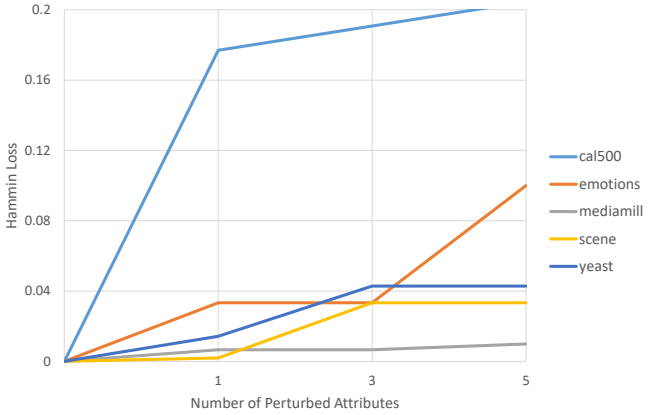

Figure 7: Error of classification (in terms of Hamming Loss) when the values of the five most relevant attributes, according to the ranking resulting from applying LRPmlV4, are replaced by a random value in their application domain.

be labeled with l_3 turns out not to be.

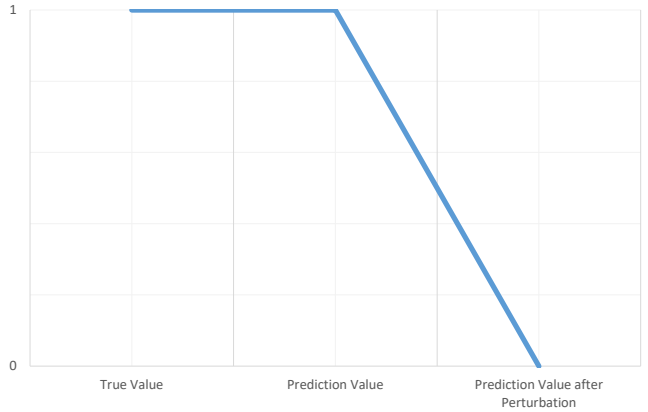

Figure 8: Hamming Loss achieved when perturbing the values of the five most relevant attributes (according to the ranking resulting from applying LRPmlV3) in predicting the l_3 label.

A second desirable property of an explanation technique is that it produces a continuous explanation function [10]. This means that: If two data points are nearly equivalent, then the explanations of their predictions should also be nearly equivalent. According to [10], the continuity of the explanation (or lack of it) can be quantified by the Equation (9),

$$\mathcal{QE} = \max_{\substack{\forall x \in \mathcal{U} \\ x \neq x'}} \frac{\left\|\sigma(x) - \tau(x')\right\|_1}{\left\|x - x'\right\|_2} \tag{9}$$

where $\sigma(x)$ and $\tau(x')$ are the rankings of the attributes according to their relevance associated with the object's output x and x', respectively. Also, $\left\|.\right\|_1$ is the normalized Spearman distance [25], and $\left\|.\right\|_2$ is the L2 norm (i.e.,

Euclidean norm). A value close to zero means that the quality of the explanation is better.

Table II shows the performance of LRPmlV1, LRPmlV2, LRPmlV3 and LRPmlV4 according to Equation (9). Each column represents the quality of explanation ($\mathcal{QE}$) obtained for each of these approaches on a set of objects. In each case, we measure the extent to which similar objects in a dataset have similar relevance values associated with their attributes. For example, in a dataset, if an object x is similar to an object x', then the relevance rankings of their attributes $\sigma(x)$ and $\tau(x')$ should also be similar. Therefore, the more this assumption is fulfilled in a dataset, the closer the value of $\mathcal{QE}$ is to 0, i.e., the better the quality of the method's explanation.

Table II: The $\mathcal{QE}$ of LRPmlV1, LRPmlV2, LRPmlV3, and LRPmlV4. LRPmlV3 is applied by propagating the label l_3 backward.

	LRPmlV1	LRPmlV2	LRPmlV3	LRPmlV4
cal500	0.33	0.34	0.36	0.32
emotions	0.30	0.23	0.19	0.32
mediamill	0.43	0.36	0.38	0.13
scene	0.05	0.05	0.04	0.05
yeast	0.18	0.14	0.15	0.11

The results show that LRPmlV4 provides more accurate explanations in datasets with many labels (i.e., cal500 and mediamill) or many attributes (i.e., scene). However, in those datasets with fewer labels, such as emotions, LRPmlV2 achieves better explanations. In this comparison, LRPmlV3 is not considered since the reported values merely indicate the explanations' quality in terms of the l_3 label and not of a global decision as the other methods do. Note that l_3 is adopted to choose a particular label for experimentation, i.e., only for experimentation purposes without following a particular criterion.

Figure 9 shows the performance of the LRPmlV2 method (in terms of $\mathcal{QE}$) when the value of the threshold ξ is varied in the range $[-1, 1)$. This range is adopted based on the activation function's characteristics, i.e., the scaled linear exponential unit [23]. The idea of this experiment is to show the effect of the threshold ξ on the method's performance. It shows how as the threshold ξ approaches zero, the quality of the method's explanation is better.

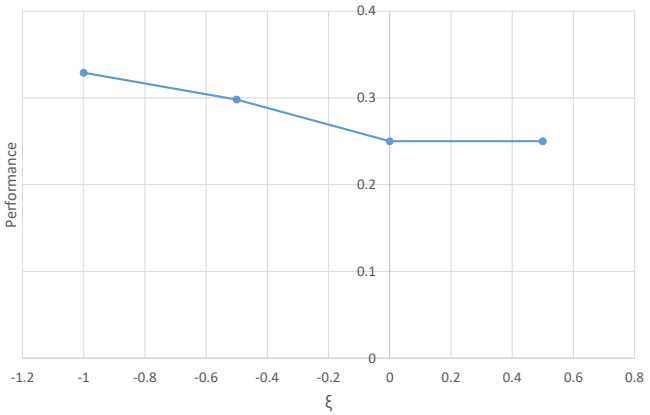

Figure 9: Performance of the LRPmlV2 method when the value of the threshold ξ is varied in the range $[-1, 1)$. In this case, the emotions dataset is used.

V. Case Study: SARS-CoV-2 Associated Coinfections

The methods proposed in this research are applied as a post-hoc interpretability stage to explain the results obtained by a neural network used to detect secondary coinfections in patients infected with SARS-CoV-2.

A. Description of the problem

COVID-19 has been affected worldwide since the end of 2019. Clinical studies have shown that a factor that increases its lethality is secondary infections [14]. In the first stage of the ongoing research at the "Comandante Manuel Fajardo Rivero" Hospital in Santa Clara, Cuba, it is evidenced that 60% of the patients with secondary infections coexisting with the SARS-CoV-2 virus died.

Coinfections associated with the infection SARS-CoV-2 are classified into bacterial infections and fungal infections, i.e., patients may develop one, both, or neither of them. From a machine learning point of view, this is considered a multi-label classification problem. One of the most effective multi-label classification methods is neural networks, which have also been successfully applied in the diagnosis of COVID-19 [1], [26]. Section IV describes the multi-label neural network used to solve this problem. However, one aspect to be taken into account in the construction of a neural model, especially in medical applications [27], is their interpretability [6].

B. Dataset under consideration

A dataset of 42 patients is available. Although the number of cases is low since this problem is relatively new, specialists in this medical field consider this information sufficient. They usually perform their analyses on similar patient samples. Also, tests performed on the neural model with well-known cases have been effective.

Each patient in a dataset has three possible labels associated with it: the patient has no coinfection (l_1), has bacterial coinfection (l_2), has fungal coinfection (l_3). The variables that characterize it are divided into five large groups: epidemiological (G1), clinical (G2), radiological (G3), clinical laboratory (G4), and microbiological (G5)—the latter group including antimicrobial susceptibility and all information related to coinfection. More details about these variable groups are given next:

G1: age ($A0$), sex ($A1$), stay in hospital ($A2$), status at admission ($A3$), hospitalization room ($A4$), personal pathological history ($A18 - A29$).

G2: clinical diagnosis ($A5 - A17$), clinical condition ($A30 - A42$), heart rate ($A43$), respiratory rate ($A44$),

evacuation status ($A48 - A50$), medications used ($A51 - A58$), invasive procedures ($A59 - A61$).

G3: x-ray report ($A45 - A47$).

G4: global leukocyte count ($A62$), neutrophil nuclear polymorphs ($A63$), lymphocytes ($A64$), platelets ($A65$), hemoglobin ($A66$), hematocrit ($A67$), creatinine ($A68$), tgp ($A69$), tgo ($A70$), d-dimer ($A71$), ggt ($A72$), ldh ($A73$), fa ($A74$), lactate ($A75$), urea ($A76$), cholesterol ($A77$), triglycerides ($A78$), uric acid ($A79$), glycemia ($A80$).

G5: number of laboratory cultures ($A81$), isolated microorganism ((escherichia.coli ($A82$), candida.spp ($A83$), psuedomona aeruginosa ($A84$), coagulase negative staphylococcus ($A85$), staphylococcus aureus ($A86$), acinetobacter baumannnii calcoaceticus complex ($A87$), klebsiella pneumoniae ($A88$), moraxella.spp ($A89$), enterobacter aerogenes ($A90$)), antimicrobial resistance ($A91$), multidrug resistance ($A92$), type of laboratory culture ((stool culture ($A93$), urine culture ($A94$), endotracheal tube culture ($A95$), central venous catheter culture ($A96$), blood culture ($A97$), tracheostomy culture ($A98$)).

C. Additional details on the neural network learning process

At the data preparation stage, nominal attributes (i.e., non-numeric attributes) are coded from a one-hot encoding. A mean value replaces missing values in the attribute's normal value range, which is done using the expert's knowledge. Finally, all attributes are normalized.

The average HL value associated with the classifier is 0.1545 after performing a leave-one-out cross-validation process. It is a particular case of cross-validation where the number of folds equals the number of objects in the dataset. Thus, the learning algorithm is applied once for each object, using all other objects as a training set and using the selected object as a single-item test set.

D. Identifying influential input variables in a patient with bacterial and fungal coinfections

Figures 10, 11, and 14 show the attribute relevance heatmap (resulting from applying the LRPmlV1, LRPmlV2 and LRPmlV4 methods) for a patient X presenting bacterial (l_2) and fungal (l_3) coinfection at the same time. Likewise, Figures 12 and 13 show this for the LRPmlV3 method as the activation values of labels l_2 and l_3 are propagated backward. Each attribute (represented by a cell in the heatmap) has its associated relevance value. Each color's intensity represents the influence that the attribute has on the output predicted by the neural model. The proposed methods explain the neural model's result for a specific case (in this case, patient X). This means that the attributes that influence patient X output do not necessarily influence other patients' decisions with different characteristics. The methods report that,

- $A91$ (with $R_{LRPmlV1}(A91) = 1.23$, $R_{LRPmlV2}(A91) = 0.7$, $R_{LRPmlV4}(A91) = 0.6$), $A84$ (with $R_{LRPmlV1}(A84) = 0.90$, $R_{LRPmlV2}(A84) = 0.49$, $R_{LRPmlV4}(A84) = 0.43$), and $A92$ (with $R_{LRPmlV1}(A92) = 0.87$, $R_{LRPmlV2}(A92) = 0.49$, $R_{LRPmlV4}(A92) = 0.43$) are the three most relevant attributes for a patient to have both coinfections.
- $A79$ (with $R_{LRPmlV1}(A79) = -0.77$, $R_{LRPmlV2}(A79) = -0.4$), and $A12$ (with $R_{LRPmlV4}(A12) = -0.35$) have a negative relevance.
- Similar results are obtained with the LRPmlV3 method, where the attributes $A91$, $A84$, and $A92$ (with $R_{LRPmlV3}(A91) = 0.24$, $R_{LRPmlV3}(A84) = 0.16$, $R_{LRPmlV3}(A92) = 0.15$) have a high influence on l_2, and l_3 (with $R_{LRPmlV3}(A91) = 0.45$, $R_{LRPmlV3}(A84) = 0.34$, and $R_{L}RPmlV3(A92) = 0.33$). Also, $A79$ has a negative impact on l_2 (with $R_{LRPmlV3}(A79) = -0.14$), and l_3 (with $R_{LRPmlV3}(A79) = -0.26$).
- Several attributes have a near-zero relevance in the presence of any of these coinfections.
- The differences between these results lie in that LRPmlV1, LRPmlV2, and LRPmlV4 show higher relevance values than LRPmlV3. This is expected since the relevance resulting from using LRPmlV3 is distributed individually for each label.

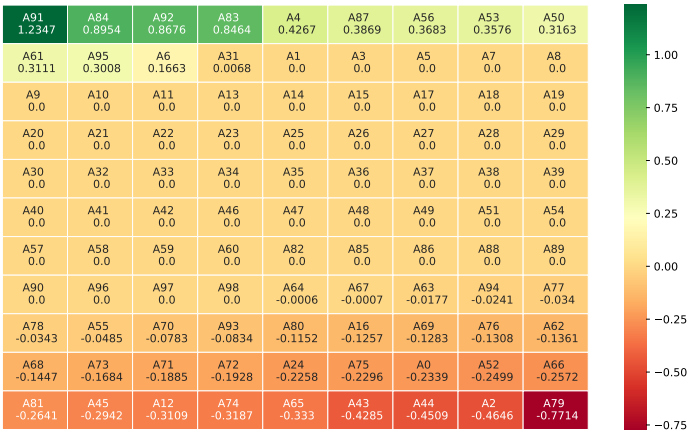

Figure 10: LRPmlV1 output. The cell in the upper left corner, attribute $A91$ (in dark green), represents the attribute most influences the patient X output. While the cell in the lower right corner, attribute $A79$ (in dark red), represents the attribute that goes most against that prediction. Tracking this heatmap as a ranking, the first attributes are $A91, A84, A92$, and the last attributes are $A44, A2, A79$.

This result was evaluated using expert criteria as suggested in [28]. The experts assessed explanations obtained from cases already known to them. They estimated that the most relevant attributes when a patient presents both coinfections coincide with those obtained by the proposed methods.

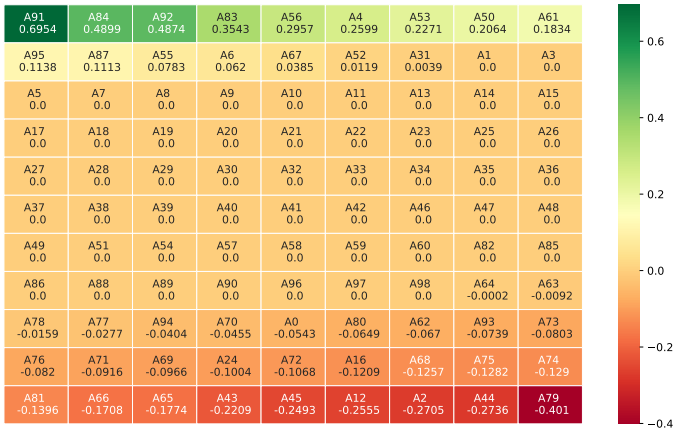

Figure 11: LRPmlV2 output. The cell in the upper left corner, attribute $A91$ (in dark green), represents the attribute most influences the patient X output. While the cell in the lower right corner, attribute $A79$ (in dark red), represents the attribute that goes most against that prediction. Tracking this heatmap as a ranking, the first attributes are $A91, A84, A92$, and the last attributes are $A2, A44, A79$.

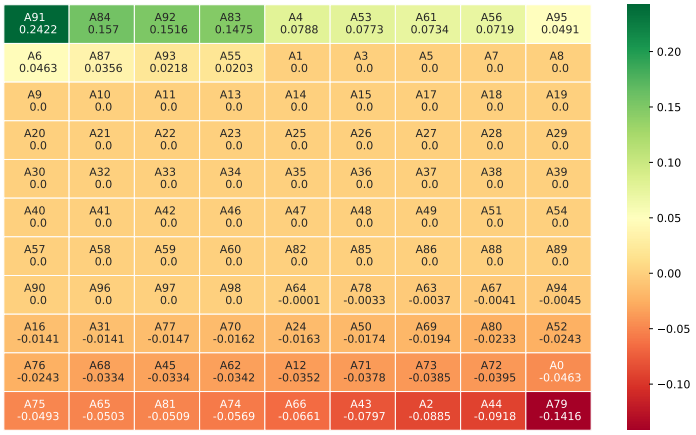

Figure 12: LRPmlV3 output when l_2 label is propagated. The cell in the upper left corner, attribute $A91$ (in dark green), represents the attribute most influences the patient X output. While the cell in the lower right corner, attribute $A79$ (in dark red), represents the attribute that goes most against that prediction. Tracking this heatmap as a ranking, the first attributes are $A91, A84, A92$, and the last attributes are $A2, A44, A79$.

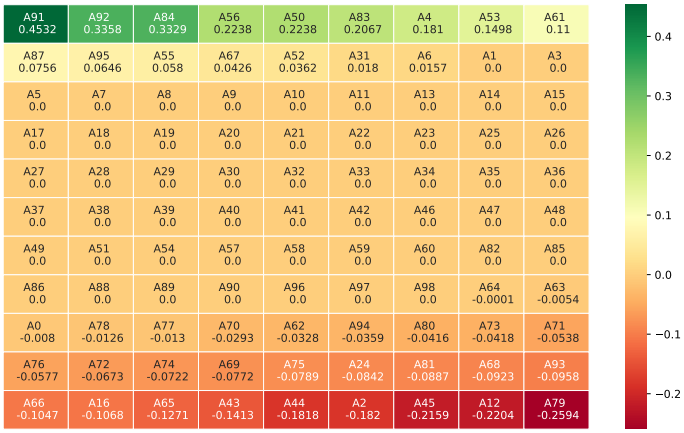

Figure 13: LRPmlV3 output when l_3 label is propagated. The cell in the upper left corner, attribute $A91$ (in dark green), represents the attribute most influences the patient X output. While the cell in the lower right corner, attribute $A79$ (in dark red), represents the attribute that goes most against that prediction. Tracking this heatmap as a ranking, the first attributes are $A91, A92, A84$, and the last attributes are $A45, A12, A79$.

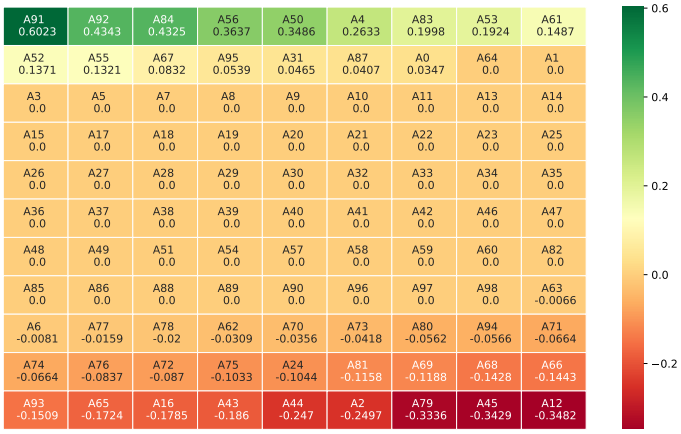

Figure 14: LRPmlV4 output. The cell in the upper left corner, attribute $A91$ (in dark green), represents the attribute most influences the patient X output. While the cell in the lower right corner, attribute $A12$ (in dark red), represents the attribute that goes most against that prediction. Tracking this heatmap as a ranking, the first attributes are $A91, A92, A84$, and the last attributes are $A79, A45, A12$.

VI. Concluding Remarks

The risk that intelligent systems may represent for human beings in some applications that are very sensitive to human life, e.g., those intended for medicine, has led to the need to develop systems capable of solving problems whose solutions can be explained. This is especially relevant when intelligent systems have been created using approaches such as neural networks. Different techniques have been developed in the so-called XAI. LRP falls into the category of interpretability post-hoc methods since it is applied to explain a solution inferred by the intelligent system for a given object.

This research presents the adaptation of the LRP method for multi-label classification scenarios. Four alternatives of redistribution of activation levels are proposed, developed from the fact that a multi-label solution may include the activation of more than one label at the

same time. Experimental studies (in terms of explanation quality) developed from international multi-label datasets show that three of the proposed methods (LRPmlV1, LRPmlV2, LRPmlV4) are effective in globally interpreting the results in a multi-label neural network. While LRPmlV3 is effective in cases where a local interpretation is needed, i.e., based on a single label's output. However, the disadvantage of the LRPmlV2 method is the need to use a threshold, which could affect the method's performance. On the other hand, LRPmlV4 is the most suitable method for those datasets with many labels, which is very common in multi-label problems.

The proposed methods are applied to a real problem as a post-hoc stage in predicting coinfections associated with SARS-CoV-2. The interpretation of the neural model results provided the Cuban medical community dedicated to COVID-19 studies with an intelligent system that satisfies the clinical requirements necessary for its use.

Acknowledgment

The authors would like to sincerely thank the doctors of the Hospital "Comandante Manuel Fajardo Rivero" in the city of Santa Clara, Cuba, who assisted us in the description and medical terminology associated with the case study under consideration. Likewise, in the validation of the results obtained as an expert in the field. This study is supported by the Special Research Fund of Hasselt University.

References

[1] Z. Car, S. Baressi Šegota, N. Anđelić, I. Lorencin, and V. Mrzljak, "Modeling the spread of covid-19 infection using a multilayer perceptron," Computational and Mathematical Methods in Medicine, vol. 2020, 2020.

[2] M. Desai and M. Shah, "An anatomization on breast cancer detection and diagnosis employing multi-layer perceptron neural network (mlp) and convolutional neural network (cnn)," Clinical eHealth, 2020.

[3] W. Samek, A. Binder, G. Montavon, S. Lapuschkin, and K.-R. Müller, "Evaluating the visualization of what a deep neural network has learned," IEEE Transactions on Neural Networks and Learning Systems, vol. 28, no. 11, pp. 2660–2673, 2016.

[4] Y. Liang, S. Li, C. Yan, M. Li, and C. Jiang, "Explaining the black-box model: A survey of local interpretation methods for deep neural networks," Neurocomputing, vol. 419, pp. 168–182, 2021.

[5] A. Adadi and M. Berrada, "Peeking inside the black-box: a survey on explainable artificial intelligence (xai)," IEEE Access, vol. 6, pp. 52 138–52 160, 2018.

[6] A. B. Arrieta, N. Díaz-Rodríguez, J. Del Ser, A. Bennetot, S. Tabik, A. Barbado, S. García, S. Gil-López, D. Molina, R. Benjamins et al., "Explainable artificial intelligence (xai): Concepts, taxonomies, opportunities and challenges toward responsible ai," Information Fusion, vol. 58, pp. 82–115, 2020.

[7] A. HLEG, "High-level expert group on artificial intelligence: Ethics guidelines for trustworthy ai," European Commission, 09.04, 2019.

[8] S. Bach, A. Binder, G. Montavon, F. Klauschen, K.-R. Müller, and W. Samek, "On pixel-wise explanations for non-linear classifier decisions by layer-wise relevance propagation," PloS One, vol. 10, no. 7, p. e0130140, 2015.

[9] W. Samek, T. Wiegand, and K.-R. Müller, "Explainable artificial intelligence: Understanding, visualizing and interpreting deep learning models," arXiv preprint arXiv:1708.08296, 2017.

[10] G. Montavon, W. Samek, and K.-R. Müller, "Methods for interpreting and understanding deep neural networks," Digital Signal Processing, vol. 73, pp. 1–15, 2018.

[11] G. Bologna, "A study on rule extraction from several combined neural networks," International Journal of Neural Systems, vol. 11, no. 03, pp. 247–255, 2001.

[12] Y. Yang, V. Tresp, M. Wunderle, and P. A. Fasching, "Explaining therapy predictions with layer-wise relevance propagation in neural networks," in IEEE International Conference on Healthcare Informatics (ICHI). IEEE, 2018, pp. 152–162.

[13] F. Herrera, F. Charte, A. J. Rivera, and M. J. Del Jesus, "Multilabel classification," in Multilabel Classification. Springer, 2016, pp. 17–31.

[14] Y. Aguilera Calzadilla, Y. Díaz Morales, L. A. Ortiz Díaz, O. L. Gonzalez Martínez, O. A. Lovelle Enríquez, and M. d. L. Sánchez Álvarez, "Infecciones bacterianas asociadas a la covid-19 en pacientes de una unidad de cuidados intensivos," Revista Cubana de Medicina Militar, vol. 49, no. 3, 2020.

[15] M. Kohlbrenner, A. Bauer, S. Nakajima, A. Binder, W. Samek, and S. Lapuschkin, "Towards best practice in explaining neural network decisions with lrp," in International Joint Conference on Neural Networks (IJCNN). IEEE, 2020, pp. 1–7.

[16] E. Gibaja and S. Ventura, "Multi-label learning: a review of the state of the art and ongoing research," Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery, vol. 4, no. 6, pp. 411–444, 2014.

[17] F. Charte, D. Charte, A. Rivera, M. J. del Jesus, and F. Herrera, "R ultimate multilabel dataset repository," in International Conference on Hybrid Artificial Intelligence Systems. Springer, 2016, pp. 487–499.

[18] D. Turnbull, L. Barrington, D. Torres, and G. Lanckriet, "Semantic annotation and retrieval of music and sound effects," IEEE Transactions on Audio, Speech, and Language Processing, vol. 16, no. 2, pp. 467–476, 2008.

[19] G. Tsoumakas, I. Katakis, and I. Vlahavas, "Effective and efficient multilabel classification in domains with large number of labels," in Workshop on Mining Multidimensional Data (MMD'08), vol. 21, 2008, pp. 53–59.

[20] C. G. Snoek, M. Worring, J. C. Van Gemert, J.-M. Geusebroek, and A. W. Smeulders, "The challenge problem for automated detection of 101 semantic concepts in multimedia," in Proceedings of the 14th ACM International Conference on Multimedia, 2006, pp. 421–430.

[21] M. R. Boutell, J. Luo, X. Shen, and C. M. Brown, "Learning multi-label scene classification," Pattern Recognition, vol. 37, no. 9, pp. 1757–1771, 2004.

[22] A. Elisseeff, J. Weston et al., "A kernel method for multi-labelled classification." in NIPS, vol. 14, 2001, pp. 681–687.

[23] G. Klambauer, T. Unterthiner, A. Mayr, and S. Hochreiter, "Self-normalizing neural networks," in 31st International Conference on Neural Information Processing Systems, ser. NIPS'17. Curran Associates Inc., 2017, pp. 972–981.

[24] G. Nápoles, M. Bello, and Y. Salgueiro, "Long-term cognitive network-based architecture for multi-label classification," Neural Networks, 2021.

[25] L. P. Dinu and F. Manea, "An efficient approach for the rank aggregation problem," Theoretical Computer Science, vol. 359, no. 1-3, pp. 455–461, 2006.

[26] M. Bello, Y. Aguilera, G. Nápoles, M. M. García, R. Bello, and K. Vanhoof, "Layer-wise relevance propagation in multi-label neural networks to identify covid-19 associated coinfections," in International Workshop on Artificial Intelligence and Pattern Recognition. Springer, 2021, pp. 3–12.

[27] R. Caruana, Y. Lou, J. Gehrke, P. Koch, M. Sturm, and N. Elhadad, "Intelligible models for healthcare: Predicting pneumonia risk and hospital 30-day readmission," in Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2015, pp. 1721–1730.

[28] R. R. Selvaraju, M. Cogswell, A. Das, R. Vedantam, D. Parikh, and D. Batra, "Grad-cam: Visual explanations from deep networks via gradient-based localization," in Proceedings of the IEEE International Conference on Computer Vision, 2017, pp. 618–626.