

# Local Multi-Label Explanations for Random Forest

Nikolaos Mylonas<sup>1</sup>[0000-0002-5733-543X], Ioannis Mollas<sup>1</sup>[0000-0002-7765-7903],  
Nick Bassiliades<sup>1</sup>[0000-0001-6035-1038], and Grigorios  
Tsoumakas<sup>1</sup>[0000-0002-7879-669X]

Aristotle University of Thessaloniki, Thessalonik, Greece  
{myloniko, iamollas, nbassili, greg}@csd.auth.gr

**Abstract.** Multi-label classification is a challenging task, particularly in domains where the number of labels to be predicted is large. Deep neural networks are often effective at multi-label classification of images and textual data. When dealing with tabular data, however, conventional machine learning algorithms, such as tree ensembles, appear to outperform competition. Random forest, being a popular ensemble algorithm, has found use in a wide range of real-world problems. Such problems include fraud detection in the financial domain, crime hotspot detection in the legal sector, and in the biomedical field, disease probability prediction when patient records are accessible. Since they have an impact on people’s lives, these domains usually require decision-making systems to be explainable. Random Forest falls short on this property, especially when a large number of tree predictors are used. This issue was addressed in a recent research named LionForests, regarding single-label classification and regression. In this work, we adapt this technique to multi-label classification problems, by employing three different strategies regarding the labels that the explanation covers. Finally, we provide a set of qualitative and quantitative experiments to assess the efficacy of this approach.

**Keywords:** Explainable Artificial Intelligence · Interpretable Machine Learning · Random Forest · Multi-Label Learning

## 1 Introduction

Multi-label classification is a popular machine learning task, concerned with assigning multiple different labels to a single sample [28]. There are plenty of applications employing multi-label classification, such as semantic indexing [21] and object detection [10]. Multi-label classification has also proven useful in the predictive maintenance [16] and financial services [3] sectors, where tabular data are mainly used. When this sort of data is available, ensemble methods are typically outperforming other families of methods [25,29]. Ensembles, however, are intrinsically not explainable. This is an important weakness, as explainability is useful for the vast majority of ML applications, and a necessity when they impact human lives or incur economic costs [1,12].

This paper focuses on the explainability of random forest (RF) [4] models in the context of multi-label classification. There is a lot of work on the explainability of RF for regression and single-label classification tasks [6,13,14,17]. However, adapting these methods to multi-label tasks, where RF models find frequent use [15,26,30], is not straightforward. There are also techniques that have been specifically designed for multi-label tasks [20,27]. These are, however, independent of the explained model’s architecture, and therefore cannot exploit the specific properties of RF models to their benefit.

To address the lack of RF-specific explainability techniques for multi-label classification in the literature, we propose an extension of LionForests [17] towards explaining multi-label classification decisions. We introduce three different strategies concerning the scope of the provided explanation (single-label, predicted labelset, label subsets). We compare these strategies against similar state-of-the-art techniques, through a set of quantitative and qualitative experiments. The results highlight the conciseness of the explanations of the proposed approach.

The rest of this paper is organized as follows. Section 2 discusses relevant research, while Section 3 introduces important concepts of the LionForests method. Section 4 presents the three novel strategies for explaining multi-label RFs. The experimental procedure, along with the data sets used, and the results, are mentioned in Section 5. Finally, we conclude and propose future steps for this research in Section 6.

## 2 Related work

Explainability techniques can be classified into two categories, depending on their applicability to different types of models. *Model-agnostic* techniques ignore model structure and are therefore applicable to any ML model, whereas *model-specific* techniques are designed to interpret a certain family of models. The latter can either alter the model’s structure to achieve explainability, or simply leverage information from the architecture without affecting it. Another distinction is between *global* and *local* explainability techniques, with the former explaining the entire model and the latter focusing on particular predictions of instances.

We first discuss model-agnostic explainability methods, which could be applied to multi-label classification, with some modifications. The use of simpler surrogate models that mimic the behavior of more complex ones, while also being more interpretable, is a topic studied in the literature and can be applied in a multi-label setting as well. Surrogates are built based on the input and the produced output of the model, and can be used to provide both global and local interpretations.

LIME [22], one of the most well-known explainability technique, provides local interpretations for all types of models by estimating feature importance using perturbation methods. Similarly, Anchors [23] extracts rule-based interpretations using a slightly different perturbation approach. Another interesting technique, LORE [11], uses a genetic algorithm to generate neighbors, which are then used

to train a decision tree that produces rule-based explanations. By constructing a decision tree to approximate the performance of complicated models, single tree approximations are used to simplify the prediction pattern of complex black-box models. TREPAN [5] approaches this task as an inductive learning problem, aiming to represent a complex model, such as a neural network, with symbolic knowledge. This is a rule learning task, according to RuleFit [9], in which each rule covers a small portion of the input space. RuleFit extracts rules from each decision tree to build a sparse linear model that incorporates both the original features and the retrieved rules, using decision trees as base learners for the various input variables. The final interpretation is based on feature importance.

We continue with explainability methods designed specifically for RF. A model-specific approximation technique for RF called DefragTrees [13] formulates the simplification of tree ensembles as a model selection problem. The aim of this work is to derive the simplest model possible that has similar performance to the whole ensemble. To do so, they employ a Bayesian model selection method to optimize the simplified model. InTrees [6], on the other hand, approaches the same problem by providing a framework for selecting and pruning specific rules from the entire ensemble, effectively summarizing the relevant rules into a new simpler and more interpretable learner that can be used for future predictions.

The RF explainability techniques discussed so far concern global explanations covering the whole model. We now move to local explanation methods. [18] introduces a local explainability method, which provides rule-based explanations exploiting feature importance. CHIRPS [14] extracts the relevant paths for an instance from each decision tree and filters them to reduce the complexity of the explanation. LionForests (LF) [17] provides explanations for the decisions of a random forest in the form of rules. A key advantage of LF is that it distills the interpretation from the knowledge already present in RF, while also providing complete explanations. This in turn means that these explanations are provided without any demerits in the model’s performance or complexity. LF can be used in binary or multi-class classification and regression problems.

Finally, we discuss RF explainability methods with a focus on visualization. iForest [31] supports the visualization of relevant paths by multi-dimensional projection. It further allows the summarization of those paths into a final one that can be used as an explanation. Another visualization tool that can provide a global overview of a random forest model in conjunction with local explanations is ExMatrix [19]. Both the global and local explanations provided by this method come in the form of a table.

We close the related work section with a method that has been designed specifically for multi-label classification, which is applicable to RF as well. MARLENA [20] is a model-agnostic approach that can provide local interpretations for black-box models by creating a neighborhood of similar instances to the one to be explained and training a decision tree. This approach is applicable to any black-box model and was evaluated mainly on health applications.

### 3 LionForests

This section introduces the fundamental concepts of LF. LF does not affect the performance of the RF model, because it is applied post-hoc and only explains decisions. The main step behind the interpretation extraction process of the technique is the estimation of the minimum number of paths across the different estimators of the RF model that cover the examined instance. From each tree estimator, we extract one path responsible for the instance’s prediction. Then, the set of paths that positively vote for that instance is identified. Through feature and path reduction, as well as feature-range formulation upon those paths, LF builds the interpretation. The estimation of the minimum number of paths is not a straightforward task, especially since it needs to comply with LF’s main property, namely *conclusiveness*. This property requires the rules produced by an explainability technique to be free of misleading or erroneous elements.

Since a multi-label classification task can be decomposed into multiple binary classification tasks, one for each label, we will further discuss how LF computes the minimum number of paths for binary classification tasks. To better understand the following statements, we first need to mention Proposition 1 introduced in the original paper, which is pivotal in LF’s procedure.

**Proposition 1.** *An RF model with a set of trees ( $T$ ), casting  $|T|$  votes, always predicts class  $M$  if and only if class  $M$  has at least a quorum of votes or more, where  $quorum = \lfloor \frac{|T|}{2} + 1 \rfloor$  out of  $|T|$  votes.*

Proposition 1 states that the minimum number of paths needed for the RF model to maintain its original prediction is  $\lfloor \frac{|T|}{2} + 1 \rfloor$ . The validity of this statement is proved in the original paper. With that in mind, the minimum number of paths which cover the examined instance that LF tries to compute is actually the quorum based on Proposition 1.

Given the minimum number of paths, LF reduces the paths extracted from each RF decision tree using a sequence of procedures, obtaining the reduced set of paths extracted from the reduced trees ( $T'$ ). In this order, they are a) reduction through association rules, b) reduction through clustering, and c) reduction by random selection. Each procedure serves a different purpose. Reduction by association rules and by clustering aim to reduce features by selecting paths with similar feature sets, whereas reduction by random selection reduces the number of paths to the quorum.

Finally, after obtaining the reduced paths, LF identifies the common features and their ranges, merging them to obtain a single range for each feature, which we call *feature-range*. The lower (upper) bound of the combined range is the maximum (minimum) of the lower (upper) bounds found across all ranges. This step is called feature aggregation. When categorical features are present, LF performs OneHot encoding on them, effectively obtaining OneHot features equal to the number of possible values the initial one had.

These new OneHot features are handled based on the following principle: if a OneHot encoded feature is present in at least one path, with a value of 1,

then the categorical feature it originates from is added to the final rule with the encoded feature as its value. For example, in a categorical feature regarding *Country*, if the examined instance has the value *Greece*, then the OneHot feature would be  $Country\_Greece = 1$ . In this case, we would include in the final rule this statement  $Country = Greece$ . The reduced paths cannot contain any other OneHot encoded feature of *Country* with a value of 1, as that would mean that the path would not cover the instance. In contrast, if it is absent from all paths, this value of the categorical feature does not affect the outcome. In that case, LF searches for the other OneHot features originating from the same categorical one that have value of 0 and adds them to the rule as values of the original feature, which can affect the outcome. Following our example, if  $Country\_Japan = 0$ ,  $Country\_United-States = 0$  (or any other), appears in the reduced paths, and given that  $Country\_Greece = 1$  does not exist in any, these features are included in the final rule in the following form  $Country \neq [Japan, United States, \dots]$ . This procedure is called categorical feature handling.

## 4 LionForest Multi-Label Explainability

In multi-label classification, predictions come in vectors of size  $|L|$ , with  $L$  denoting the set of all labels. In this case, an explanation could concern: i) the set of positively predicted labels,  $L_p \subseteq L$ , ii) subsets of it,  $L'_p \subseteq L_p$ , or iii) each one of its individual labels,  $l \in L_p$ . We propose three corresponding strategies that allow LF to output multi-label explanations, each calculating the quorum in a different way, based on the subset of  $L_p$  that we want our explanation to cover. As in the original, the three strategies do not meddle with the performance of the RF model.

We also introduce an example, which will be used for all strategies. Consider an RF model with  $|T| = 9$  estimators, that outputs its prediction concerning  $|L| = 5$  labels for a given input. Based on the theory presented before, the quorum equals  $\lfloor \frac{9}{2} + 1 \rfloor = \lfloor 5.5 \rfloor = 6$ . For a given instance, RF predicts the following labelset  $[0, 1, 0, 1, 1]$ . From each  $t \in T$  tree estimator, we extract the path and the prediction for this instance. Then, based on the strategy, we proceed to the appropriate reduction and eventually the formulation of the final rule interpretation. In Figure 1, the predicted labelsets from each  $t$  tree are visible.

### 4.1 Explaining each predicted label separately

The first step of this strategy (LF-1) is the extraction of all paths regarding a decision from the tree predictors comprising the RF model. Then, for each predicted label  $l \in L_p$ , a multi-stage process takes place, which first identifies the paths that vote for its prediction. The next step is the reduction of  $T$ , as explained in Section 3, to the number denoted by the quorum, obtaining the minimum number of paths from  $T'$  trees. The rule building steps remain the same as those in the original technique, namely feature aggregation and handling

	<u>Prediction:</u>	<u>Per Label:</u>			<u>All:</u>	<u>Frequent Set:</u>
RF:	[0, 1, 0, 1, 1]	[0, 1, 0, 1, 1]	[0, 1, 0, 1, 1]	[0, 1, 0, 1, 1]	[0, 1, 0, 1, 1]	[0, 1, 0, 1, 1]
T	[0, 1, 0, 1, 1]	[0, 1, 0, 1, 1]	[0, 1, 0, 1, 1]	[0, 1, 0, 1, 1]	[0, 1, 0, 1, 1]	[0, 1, 0, 1, 1]
	[1, 1, 1, 1, 1]	[1, 1, 1, 1, 1]	[1, 1, 1, 1, 1]	[1, 1, 1, 1, 1]	[1, 1, 1, 1, 1]	<u>[1, 1, 1, 1, 1]</u>
	[0, 0, 0, 0, 0]	[0, 0, 0, 0, 0]	[0, 0, 0, 0, 0]	[0, 0, 0, 0, 0]	[0, 0, 0, 0, 0]	[0, 0, 0, 0, 0]
	[0, 1, 0, 0, 1]	[0, 1, 0, 0, 1]	[0, 1, 0, 0, 1]	[0, 1, 0, 0, 1]	[0, 1, 0, 0, 1]	[0, 1, 0, 0, 1]
	[0, 1, 0, 1, 1]	<u>[0, 1, 0, 1, 1]</u>	[0, 1, 0, 1, 1]	<u>[0, 1, 0, 1, 1]</u>	<u>[0, 1, 0, 1, 1]</u>	[0, 1, 0, 1, 0]
	[0, 1, 0, 0, 0]	<u>[0, 1, 0, 0, 0]</u>	[0, 1, 0, 0, 0]	[0, 1, 0, 0, 0]	[0, 1, 0, 0, 0]	[0, 1, 0, 0, 0]
	[0, 1, 0, 1, 1]	<u>[0, 1, 0, 1, 1]</u>	[0, 1, 0, 1, 1]	<u>[0, 1, 0, 1, 1]</u>	[0, 1, 0, 1, 1]	[0, 1, 0, 1, 1]
	[1, 1, 0, 1, 1]	[1, 1, 0, 1, 1]	<u>[1, 1, 0, 1, 1]</u>	[1, 1, 0, 1, 1]	[1, 1, 0, 1, 1]	[1, 1, 0, 1, 1]
	[0, 1, 0, 1, 1]	[0, 1, 0, 1, 1]	[0, 1, 0, 1, 1]	[0, 1, 0, 1, 1]	[0, 1, 0, 1, 1]	[0, 1, 0, 1, 1]
		8→5	6→5	7→5	6→5	6→5

**Fig. 1.** Running example. Greyed out predictions do not cover the examined label/labelset, whereas black and red (underlined) do. The predictions whose paths were decreased are shown in red

of categorical features. After formulating a rule for each predicted label  $l$ , we use these rules as an explanation for the examined instance.

In columns 2 to 4 (*Per Label*) in Figure 1, we can see how LF selects and reduces the paths to the quorum. It identifies the paths that voted for each of the three predicted labels (black and red font). If the number of paths exceeds the quorum, the LF reduction strategies are used to decrease them to the bare minimum (black font). Treating each label separately can result in smaller feature sets in the final interpretation. This is because LF has a greater number of possible paths to reduce to the minimum.

## 4.2 Explaining all the predicted labelset

This strategy is largely similar to the previous one, with the main difference being that instead of an iterative process for each label (*LF-a*), this time a single process is executed for the whole predicted labelset. This in turn means that LF must now identify the paths from the  $T$  trees that include the whole predicted labelset in its prediction, greatly limiting the number of available paths to be reduced in the following step, if possible. Furthermore, during the path reduction step, each produced path set must cover the whole prediction, restricting the number of paths LF can safely remove to obtain  $|T'|$ .

It is worth noting that, due to the above conditions, the final rule obtained after applying the rule-building steps is very specific to the examined instance. There is a possibility that the number of recognized paths covering all predicted

labels will be less than the quorum. This prevents us from further decreasing them, but also prohibits us from using them alone to form the final rule. In this scenario, regardless of their vote, we use all the paths.

Connecting this strategy with the running example of Figure 1, we focus on the third column, *All*. Only 6 paths include all the predicted labels at the same time. LF will use the reduction strategies to decrease those pathways to 5 (quorum). However, because there is so little room for reduction, the strategy’s effectiveness is limited, and therefore, we might not observe the desired feature reduction. This strategy is more effective in confident classifiers, where the number of individual estimators voting the whole predicted labelset is larger.

### 4.3 Explaining frequent label subsets

This strategy provides explanations for subsets ( $LF-p$ ) of the predicted labelset that frequently appear inside the examined data set. These subsets are identified with the use of association rules and specifically the *fpgrowth* algorithm. Then, a process comparable to the one present in the first strategy is performed. For each subset, the paths that vote for all the labels present inside it are identified and then reduced to  $|T'|$ , before the rule building steps that formulate the final rule for this subset are implemented. The final explanation for the frequent subsets is an aggregation of the rules built by the aforementioned process.

In case of larger labelsets, as well as a large set of predicted labels, the number of activated subsets can be very high. Therefore, the end-user is given an option to limit the number of subsets. Hence, if the activated subsets are  $X$  and the user asks for  $N < X$ , the first  $N$  subsets and their explanation will be provided, ordered based on the support of the subset across the labelsets of the training data set.

In the example (Figure 1), the last column presents the explanation of one identified subset  $[0, 1, 0, 1, 0] \subset [0, 1, 0, 1, 1]$ , the paths which cover this set, and the removed path.

## 5 Experiments

This section summarizes the experiments that we carried out to compare the performance of our strategies to state-of-the-art techniques frequently used in the literature. To further the reliability of our results we performed a 10-fold cross validation and present the standard deviation of our showcased results in each table. We performed three distinct sets of experiments. The first compares our various strategies to each other, in order to gain insight into their effectiveness with multi-label data. The second focuses on techniques that explain the entire predicted labelset, pitting our second strategy against MARLENA, due to its relevance to our task, and two baselines: local (LS) and global (GS) tree surrogates. The third and final set compares our first strategy to Anchors and CHIRPS. Anchors was selected for its prominence in the literature, and CHIRPS for its similarity to our approach, concerning the per label experiments. Both

techniques have been adjusted to provide explanations for each of the predicted labels. The code for these experiments is available in GitHub<sup>1</sup> and DockerHub<sup>2</sup>.

## 5.1 Data sets

Here, we present the data sets used and the pre-processing steps we followed for each one of them. We selected four multi-label tabular data sets, whose main statistics and performance<sup>3</sup> can be found in Table 1. The values inside the parentheses concern the numbers after the pre-processing steps for each data set.

**Table 1.** Data set information

ID	Dataset	Performance	Instances	Features	Labels	Cardinality
D1	Food Truck	51.73%	407	21 (29)	12	2.29
D2	Water Quality	51.79%	1060	16	14	5.07
D3	Flags	76.82%	194	19	7	3.39
D4	AIAI	88.77%	10K (339)	7 (6)	6 (4)	1.04

*D1: Food Truck.* This data set was compiled from the replies of 407 survey participants, concerning their food preferences, personal information, time of meal (input variables) and types of food trucks they eat from (target variables) [24]. We replaced missing values with the mean value across their column. Furthermore, four categorical features were present. As such, two of them were handled as ordinal (gender and time of the day) and the rest were one-hot encoded (motivation and marital status). Finally, the data were min-max scaled.

*D2: Water Quality.* This scientific data set was used for modeling the quality of water in Slovenian rivers. It contains features like the water’s temperature, pH, and concentration of different chemicals. The target variables correspond to different water quality indicators [2]. The pre-processing was minimal only including min-max scaling.

*D3: Flags.* The third data set used in our experiments contains information about certain countries and their culture. The target variables concern the colors that are present on the flag of each country [7]. No pre-processing took place for this data set other than min-max scaling.

<sup>1</sup> <https://tinyurl.com/c5y8uxm4>

<sup>2</sup> <https://tinyurl.com/2nh4zyj3>

<sup>3</sup> Performance was estimated following a grid search, and is estimated based on micro averaged  $F_1$  score



*D4: AI4I.* The last data set employed in our experiments is a synthetic predictive maintenance data set concerning the prediction of certain types of machine failures [16]. Aiming to emulate real predictive maintenance problems encountered in the industry, the data set contains features like temperature, torque, rotational speed of different machines to predict possible failures. There is a single categorical feature (type) inside the data set which we handle as ordinal, and a feature without information (product ID). We then min-max scale all features. Regarding the labels, there is a strong dependency between the first denoting whether there is a failure or not and the rest of the labelset, which are the types of failures. As such, we only keep the examples exhibiting some kind of failure, while removing the first. Finally, we also remove the final label which denotes a random failure, of which very few instances exist in the data set.

## 5.2 Quantitative experiments

In order to perform our quantitative experiments, we made use of four different metrics, each one covering a different aspect of a produced rule. Rule length (L) denotes the number of feature-ranges present inside the rule. Smaller lengths are easier for the end-user to comprehend but can also indicate that the explanation is problematic. Furthermore, expert users tend to prefer larger rules, due to their richer information [8]. It is worth noting that for techniques providing rules for each label or label subset separately, the final rule length is the sum of the lengths of those individual rules. Coverage (C) describes the average number of instances each rule satisfies. Precision (P) refers to the fraction of correctly covered instances among the ones covered by each rule. Higher values on both metrics correspond to better performance. Time response (T) describes the run time of the technique in seconds.

**Table 2.** Comparison between the different strategies of LF

Datasets	Algorithm	L	T
D1	LF-a	20.33 $\pm$ 0.70	6.25 $\pm$ 0.96
	LF-l	30.06 $\pm$ 3.47	10.30 $\pm$ 0.86
	LF-p	44.71 $\pm$ 9.04	11.33 $\pm$ 1.16
D2	LF-a	16.37 $\pm$ 0.40	0.94 $\pm$ 0.04
	LF-l	57.64 $\pm$ 5.71	1.37 $\pm$ 0.11
	LF-p	153.34 $\pm$ 16.76	2.87 $\pm$ 0.29
D3	LF-a	18.24 $\pm$ 0.49	1.05 $\pm$ 0.10
	LF-l	62.58 $\pm$ 5.24	5.66 $\pm$ 0.40
	LF-p	111.72 $\pm$ 7.32	7.38 $\pm$ 0.48
D4	LF-a	4.92 $\pm$ 0.46	1.38 $\pm$ 0.30
	LF-l	5.04 $\pm$ 0.48	1.41 $\pm$ 0.31
	LF-p	5.04 $\pm$ 0.48	1.41 $\pm$ 0.30

We first discuss the results of the comparison among our proposed strategies that can be seen in Table 2. The goal of this setup was to compare the length of

the rules produced by each strategy, along with the time needed to produce them. As expected, the strategy providing one rule for the whole predicted labelset results in shorter rules and smaller run times. This can be seen in all four data sets, albeit the differences are minimal in the last one. The frequent pair strategy seems to be the one with the longest run times and lengthiest rules. Such an outcome was anticipated, given that in most cases the number of frequent subsets present in a labelset is higher than the number of distinct labels comprising it.

The comparison regarding the techniques providing explanations for the whole predicted labelset can be seen in Table 3. Exploring each metric separately reveals that all four techniques evaluated have rather poor Coverage, with MARLENA having the best performance and LF-a having the worst. In terms of Precision, LF-a always achieves perfect precision due to its conclusiveness property, while both surrogate models surpass MARLENA. In all datasets, LF-a produces the lengthiest rules, while MARLENA produces the shortest. Finally, save for LF-a’s poor performance on the first dataset, there is no clear winner among the other local techniques in terms of time response, except for GS which is a global technique trained only once and has zero inference time.

**Table 3.** Comparison between techniques explaining the whole predicted labelset

Dataset	Algorithms	C	L	P	T
D1	LF-a	0.02 $\pm$ 0.00	20.32 $\pm$ 0.65	1.00 $\pm$ 0.00	6.25 $\pm$ 0.98
	GS	0.06 $\pm$ 0.03	7.81 $\pm$ 1.39	0.77 $\pm$ 0.08	0.00 $\pm$ 0.00
	LS	0.07 $\pm$ 0.02	6.96 $\pm$ 0.59	0.75 $\pm$ 0.07	1.52 $\pm$ 0.02
	MARLENA	0.15 $\pm$ 0.04	4.42 $\pm$ 0.27	0.73 $\pm$ 0.05	2.40 $\pm$ 0.02
D2	LF-a	0.01 $\pm$ 0.00	16.36 $\pm$ 0.40	1.00 $\pm$ 0.00	0.94 $\pm$ 0.04
	GS	0.02 $\pm$ 0.00	8.78 $\pm$ 0.36	0.66 $\pm$ 0.05	0.00 $\pm$ 0.00
	LS	0.03 $\pm$ 0.00	6.67 $\pm$ 0.20	0.65 $\pm$ 0.03	1.41 $\pm$ 0.03
	MARLENA	0.05 $\pm$ 0.01	5.97 $\pm$ 0.33	0.62 $\pm$ 0.05	2.04 $\pm$ 0.01
D3	LF-a	0.05 $\pm$ 0.00	18.24 $\pm$ 0.48	1.00 $\pm$ 0.00	1.05 $\pm$ 0.10
	GS	0.08 $\pm$ 0.01	5.22 $\pm$ 0.65	0.90 $\pm$ 0.05	0.00 $\pm$ 0.00
	LS	0.08 $\pm$ 0.02	5.31 $\pm$ 0.62	0.91 $\pm$ 0.03	3.41 $\pm$ 0.06
	MARLENA	0.13 $\pm$ 0.03	4.41 $\pm$ 0.20	0.85 $\pm$ 0.05	0.86 $\pm$ 0.00
D4	LF-a	0.03 $\pm$ 0.00	4.92 $\pm$ 0.46	1.00 $\pm$ 0.00	1.38 $\pm$ 0.30
	GS	0.24 $\pm$ 0.15	3.62 $\pm$ 0.57	0.92 $\pm$ 0.07	0.00 $\pm$ 0.00
	LS	0.24 $\pm$ 0.14	3.41 $\pm$ 0.38	0.94 $\pm$ 0.04	1.43 $\pm$ 0.04
	MARLENA	0.33 $\pm$ 0.13	2.52 $\pm$ 0.26	0.75 $\pm$ 0.06	0.80 $\pm$ 0.01

The last set of experiments concerns explanations for each predicted label and can be seen in Table 4. This setup allows us to use single-label interpretation techniques like Anchors and CHIRPS. In all datasets, CHIRPS outperforms its competitors in terms of Coverage. The longest rules are provided by LF-l, while the shortest are provided by CHIRPS. In terms of precision, LF-l works flawlessly, with Anchors ranking second with a small advantage over CHIRPS. Regarding the time response, LF-l and CHIRPS produce similar results with

**Table 4.** Comparison between techniques explaining each predicted label separately

Dataset	Algorithms	C	L	P	T
D1	LF-l	0.03 $\pm$ 0.00	41.85 $\pm$ 4.8	1.00 $\pm$ 0.00	10.26 $\pm$ 0.90
	Anchors	0.06 $\pm$ 0.02	8.95 $\pm$ 2.3	0.99 $\pm$ 0.02	341.62 $\pm$ 105.93
	CHIRPS	0.73 $\pm$ 0.09	3.66 $\pm$ 1.1	0.89 $\pm$ 0.03	8.47 $\pm$ 0.35
D2	LF-l	0.01 $\pm$ 0.00	57.64 $\pm$ 5.71	1.00 $\pm$ 0.00	1.37 $\pm$ 0.11
	Anchors	0.04 $\pm$ 0.01	31.05 $\pm$ 3.47	0.96 $\pm$ 0.02	302.07 $\pm$ 44.43
	CHIRPS	0.38 $\pm$ 0.05	10.36 $\pm$ 1.37	0.76 $\pm$ 0.05	8.24 $\pm$ 0.50
D3	LF-l	0.05 $\pm$ 0.00	62.58 $\pm$ 5.24	1.00 $\pm$ 0.00	5.77 $\pm$ 0.42
	Anchors	0.31 $\pm$ 0.05	7.36 $\pm$ 1.24	0.97 $\pm$ 0.02	63.00 $\pm$ 17.91
	CHIRPS	0.71 $\pm$ 0.05	2.09 $\pm$ 0.45	0.94 $\pm$ 0.03	2.35 $\pm$ 0.19
D4	LF-l	0.03 $\pm$ 0.00	5.04 $\pm$ 0.48	1.00 $\pm$ 0.00	1.41 $\pm$ 0.31
	Anchors	0.14 $\pm$ 0.05	3.98 $\pm$ 0.52	0.92 $\pm$ 0.07	29.53 $\pm$ 3.42
	CHIRPS	0.33 $\pm$ 0.14	2.44 $\pm$ 0.66	0.93 $\pm$ 0.06	1.25 $\pm$ 0.09

small deviations. Anchors, on the other hand, requires a significant amount of computational resources, leading to longer time responses.

### 5.3 Qualitative experiments

Our qualitative experiments focus on the AI4I (D4) data set, as its small feature set makes it easier to present and analyze an example. The features available in this data set along with their ranges and the values of a sample instance can be found in Table 5.

**Table 5.** Dataset features, their ranges, and the values of a sample instance

Feature	Range	Instance values
Type	[1, 3]	3
Air temperature [K]	[295.6, 304.4]	300.7
Process temperature [K]	[306.1, 313.7]	310.2
Rotational speed [rpm]	[1212, 2874]	1364
Torque [Nm]	[4.2, 76.2]	65.3
Tool wear [min]	[0, 251]	208

The first qualitative comparison found in Table 6 includes the rules produced by the different strategies of LF for the examined instance. As the quantitative experiments in Table 2 suggested, we can see that LF-a, the strategy explaining all predicted labels, provides the lengthier and more specific individual rule. When LF-l is employed, we can see that for the predicted label ‘OSF’, the rule is 2 feature-ranges smaller, while the rule regarding the predictions ‘PWF’ and ‘TWF’ are 1 feature-range smaller and have wider ranges. The third strategy, LF-p, produces rules explaining frequent label subsets. In this example, it produces a rule explaining the labels ‘PWF’ and ‘OSF’, a frequent labelset present in the

**Table 6.** Example Rules by the proposed strategies

Techniques	Interpretation
LF-a	<b>If</b> $2.5 \leq Type \leq 3$ and $300.6 \leq Air\ temperature \leq 301.7$ and $310 \leq Process\ temperature \leq 313.7$ and $1351 \leq Rotational\ speed \leq 1380$ and $65.2 \leq Torque \leq 65.5$ and $207.5 \leq Tool\ wear \leq 209$ <b>then</b> TWF PWF OSF
LF-l	<b>If</b> $2.5 \leq Type \leq 3$ and $295.6 \leq Air\ temperature \leq 301.7$ and $1322.5 \leq Rotational\ speed \leq 1419.5$ and $65.2 \leq Torque \leq 76.2$ and $206.5 \leq Tool\ wear \leq 251.0$ <b>then</b> TWF <b>If</b> $2.5 \leq Type \leq 3$ and $295.6 \leq Air\ temperature \leq 301.7$ and $1351 \leq Rotational\ speed \leq 1380$ and $65.2 \leq Torque \leq 76.2$ and $188 \leq Tool\ wear \leq 251$ <b>then</b> PWF <b>If</b> $2.5 \leq Type \leq 3$ and $300.6 \leq Air\ temperature \leq 300.8$ and $65.2 \leq Torque \leq 65.5$ and $207.5 \leq Tool\ wear \leq 251$ <b>then</b> OSF
LF-p	<b>If</b> $2.5 \leq Type \leq 3$ and $295.6 \leq Air\ temperature \leq 301.7$ and $1351 \leq Rotational\ speed \leq 1380$ and $65.2 \leq Torque \leq 76.2$ and $185 \leq Tool\ wear \leq 251$ <b>then</b> PWF OSF

prediction, and the rule is 1 feature-range smaller. Therefore, the user can choose between the available strategies based on their needs. We should mention that all these rules are conclusive, therefore, any change on the features between the given ranges, or any change on the features not appearing in the rules will not impact the prediction.

**Table 7.** Example rules for the whole labelset

Technique	Interpretation
GS	<b>If</b> $Air\ temperature \leq 301.7$ and $Tool\ wear > 176.5$ and $Torque > 65.2$ <b>then</b> TWF PWF OSF
LS	<b>If</b> $Air\ temperature \leq 301.7$ and $Tool\ wear > 188$ and $Torque > 48.4$ and $Type > 2.5$ <b>then</b> TWF PWF OSF
MARLENA	<b>If</b> $Air\ temperature \leq 303$ and $Rotational\ speed \leq 1382.4$ and $Type > 2.97$ <b>then</b> TWF PWF OSF

Continuing our comparisons, Table 7 presents the explanations given by GS, LS, and MA for the whole predicted labelset. These techniques provide rules that are substantially shorter and have broader feature-ranges than LF-a. However, the trade-off for these properties is the loss of conclusiveness. To support our claim, we perform three separate modifications on the values of the examined instance (one feature at a time), to demonstrate that the rules provided by the competitors do not account for these changes, in contrast to LF-a. GS does not contain a feature-range for *Type*, suggesting that it does not impact the prediction. Nonetheless, changing the value of this feature to either 1 or 2 alters the prediction to ‘OSF’. The feature-range given by LS for *Torque* is deceptively

wide. Lowering the value of this feature from 65.3 to 50 changes the prediction like before. Additionally, the range displayed by MA regarding *Air temperature* is inaccurate, as increasing its value from 300.7 to 302 causes the prediction to change from ‘TWF’, ‘PWF’, and ‘OSF’ to ‘OSF’. Finally, both LS and MA incorrectly ignore a feature, *Air temperature* and *Torque*, respectively, since both affect the prediction as seen before.

**Table 8.** Example rules per label

Technique	Interpretation
Anchors	<b>If</b> <i>Air temperature</i> $\leq$ 301.6 and <i>Type</i> $>$ 2 and <i>Tool wear</i> $>$ 207.5 and <i>Torque</i> $>$ 61.2 and <i>Rotational speed</i> $\leq$ 1365 <b>then</b> TWF
	<b>If</b> <i>Torque</i> $>$ 61.2 and <i>Air temperature</i> $\leq$ 301.6 and <i>Type</i> $>$ 2 and $1326.5 <$ <i>Rotational speed</i> $\leq$ 1365 and $309.5 <$ <i>Process temperature</i> $\leq$ 311.2 <b>then</b> PWF
	<b>If</b> <i>Tool wear</i> $>$ 207.5 and <i>Torque</i> $>$ 61.2 <b>then</b> OSF
CHIRPS	<b>If</b> {} <b>then</b> TWF <b>If</b> <i>Air temperature</i> $\leq$ 302.5 and <i>Torque</i> $>$ 65 <b>then</b> PWF <b>If</b> <i>Tool wear</i> $>$ 176.5 and <i>Torque</i> $>$ 65 <b>then</b> OSF

Concluding our qualitative study, we present the rules provided by Anchors and CHIRPS in Table 8. We can see that Anchors provides rules of similar length to LF-1, compared to the significantly shorter ones of CHIRPS. Nevertheless, similarly to before, the shorter rules are inconclusive. Rules provided by CHIRPS have a lot of inaccuracies, with the most obvious being the empty rule for the ‘TWF’ prediction. We also spot few inaccuracies in the lengthier rules of Anchors. For example, the explanations for labels ‘TWF’ and ‘PWF’ suggest that values above 61.2 for *Torque* lead to predictions containing these 2 labels. However, increasing the value to 65 results in both not being predicted by the model. Contrarily, the rules provided by LF-1 contain the correct feature ranges for *Torque*.

## 6 Conclusion

This paper proposed three different strategies that extend LF, so it can also be used to provide explanations for multi-label classification problems. Each of these strategies, explain the predicted labelset from a different point of view, resulting in rules of different length for each one of them, as well as different time responses. All three, however, retain the conclusiveness property of the original technique, providing concise explanations.

This was validated by our experimental procedure, where all strategies achieve a Precision of 1 throughout all the different setups and data sets, something that no other competitor manages to reach. Having said that, our strategies tend to produce lengthier rules that cover a smaller portion of instances than

the competitors, meaning they are more specific. This attribute is not necessarily a shortcoming, considering that expert users prefer longer, more informative explanations. In addition, the low rule length of the other techniques can be misleading, as they tend to provide even empty explanations as showcased in our qualitative experiments. However, the user is also given the option to reduce the quorum, resulting in smaller rules losing the conclusiveness property.

Some of the future steps of this research include the extension of the technique, so it can also be applied to multi target regression problems, in conjunction to an extensive experimental procedure including new competitors, more data sets and additional metrics. Furthermore, a user study to assess the quality of the rules the three different strategies produce, can be performed. Doing so, will allow us to obtain insight from different types of users about the strategies and their applicability in various domains. Finally, applying a similar strategy to produce explanations for other ensemble models can also be explored in another work.

## Acknowledgments

The research work was supported by the Hellenic Foundation for Research and Innovation (H.F.R.I.) under the “First Call for H.F.R.I. Research Projects to support Faculty members and Researchers and the procurement of high-cost research equipment grant” (Project Number: 514)

## References

1. Amina Adadi and Mohammed Berrada. Peeking inside the black-box: A survey on explainable artificial intelligence (xai). *IEEE Access*, 6:52138–52160, 2018.
2. Hendrik Blockeel, Sašo Džeroski, and Jasna Grbović. Simultaneous prediction of multiple chemical parameters of river water quality with tilde. In *European Conference on Principles of Data Mining and Knowledge Discovery*, pages 32–40. Springer, 1999.
3. Matthias Bogaert, Justine Lootens, Dirk Van den Poel, and Michel Ballings. Evaluating multi-label classifiers and recommender systems in the financial service sector. *European Journal of Operational Research*, 279(2):620–634, 2019.
4. Leo Breiman. Random forests. *Machine Learning*, 45(1):5–32, Oct 2001.
5. Mark Craven and Jude Shavlik. Extracting tree-structured representations of trained networks. In D. Touretzky, M.C. Mozer, and M. Hasselmo, editors, *Advances in Neural Information Processing Systems*, volume 8. MIT Press, 1995.
6. Houtao Deng. Interpreting tree ensembles with intrees. *Int. J. Data Sci. Anal.*, 7(4):277–287, 2019.
7. Dheeru Dua and Casey Graff. UCI machine learning repository, 2017.
8. Alex A. Freitas. Comprehensible classification models: A position paper. *SIGKDD Explor. Newsl.*, 15(1):1–10, March 2014.
9. Jerome H Friedman and Bogdan E Popescu. Predictive learning via rule ensembles. *The Annals of Applied Statistics*, 2(3):916–954, 2008.
10. Tao Gong, Bin Liu, Qi Chu, and Nenghai Yu. Using multi-label classification to improve object detection. *Neurocomputing*, 370:174–185, 2019.

11. Riccardo Guidotti, Anna Monreale, Fosca Giannotti, Dino Pedreschi, Salvatore Ruggieri, and Franco Turini. Factual and counterfactual explanations for black box decision making. *IEEE Intelligent Systems*, 34(6):14–23, 2019.
12. Riccardo Guidotti, Anna Monreale, Salvatore Ruggieri, Franco Turini, Fosca Giannotti, and Dino Pedreschi. A survey of methods for explaining black box models. *ACM Comput. Surv.*, 51(5), aug 2018.
13. Satoshi Hara and Kohei Hayashi. Making tree ensembles interpretable: A bayesian model selection approach. In Amos Storkey and Fernando Perez-Cruz, editors, *Proceedings of the Twenty-First International Conference on Artificial Intelligence and Statistics*, volume 84 of *Proceedings of Machine Learning Research*, pages 77–85. PMLR, 09–11 Apr 2018.
14. Julian Hatwell, Mohamed Medhat Gaber, and R. Muhammad Atif Azad. CHIRPS: Explaining random forest classification. *Artificial Intelligence Review*, 53(8):5747–5788, June 2020.
15. Samaneh Kouchaki, Yang Yang, Alexander Lachapelle, Timothy M. Walker, A. Sarah Walker, CRyPTIC Consortium, Timothy E. A. Peto, Derrick W. Crook, and David A. Clifton. Multi-label random forest model for tuberculosis drug resistance classification and mutation ranking. *Frontiers in Microbiology*, 11, 2020.
16. Stephan Matzka. Explainable artificial intelligence for predictive maintenance applications. In *2020 Third International Conference on Artificial Intelligence for Industries (AI4I)*, pages 69–74. IEEE, 2020.
17. Ioannis Mollas, Nick Bassiliades, and Grigorios Tsoumakas. Conclusive local interpretation rules for random forests. *CoRR*, abs/2104.06040, 2021. to appear in DAMI, Springer.
18. Alexander Moore, Vanessa Murdock, Yaxiong Cai, and Kristine Jones. Transparent tree ensembles. In *The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval, SIGIR '18*, page 1241–1244, New York, NY, USA, 2018. Association for Computing Machinery.
19. Mário Popolin Neto and Fernando V. Paulovich. Explainable matrix - visualization for global and local interpretability of random forest classification ensembles. *IEEE Trans. Vis. Comput. Graph.*, 27(2):1427–1437, 2021.
20. Cecilia Panigutti, Riccardo Guidotti, Anna Monreale, and Dino Pedreschi. Explaining multi-label black-box classifiers for health applications. In *International Workshop on Health Intelligence*, pages 97–110. Springer, 2019.
21. Yannis Papanikolaou, Grigorios Tsoumakas, Manos Laliotis, Nikos Markantonatos, and Ioannis P. Vlahavas. Large-scale online semantic indexing of biomedical articles via an ensemble of multi-label classification models. *J. Biomed. Semant.*, 8(1):43:1–43:13, 2017.
22. Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. Why should i trust you?: Explaining the predictions of any classifier. In *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining*, pages 1135–1144. ACM, 2016.
23. Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. Anchors: High-Precision Model-Agnostic Explanations. In *Thirty-Second AAAI Conference on Artificial Intelligence*, 2018.
24. Adriano Rivolli, Larissa C Parker, and Andre CPLF de Carvalho. Food truck recommendation using multi-label classification. In *EPIA Conference on Artificial Intelligence*, pages 585–596. Springer, 2017.
25. Lior Rokach, Alon Schclar, and Ehud Itach. Ensemble methods for multi-label classification. *Expert Systems with Applications*, 41(16):7507–7523, 2014.

26. Seema Sharma and Deepti Mehrotra. Comparative analysis of multi-label classification algorithms. In *2018 First International Conference on Secure Cyber Computing and Communication (ICSCCC)*, pages 35–38, 2018.
27. Karim Tabia. Towards explainable multi-label classification. In *2019 IEEE 31st International Conference on Tools with Artificial Intelligence (ICTAI)*, pages 1088–1095, 2019.
28. Grigorios Tsoumakas and Ioannis Katakis. Multi-label classification: An overview. *Int. J. Data Warehous. Min.*, 3(3):1–13, 2007.
29. Qingyao Wu, Mingkui Tan, Hengjie Song, Jian Chen, and Michael K. Ng. Mlforest: A multi-label tree ensemble method for multi-label classification. *IEEE Transactions on Knowledge and Data Engineering*, 28(10):2665–2680, 2016.
30. Xin Wu, Yuchen Gao, and Dian Jiao. Multi-label classification based on random forest algorithm for non-intrusive load monitoring system. *Processes*, 7(6), 2019.
31. Xun Zhao, Yanhong Wu, Dik Lun Lee, and Weiwei Cui. iforest: Interpreting random forests via visual analytics. *IEEE Transactions on Visualization and Computer Graphics*, 25(1):407–416, 2019.