Argumentation-based Explainable Machine Learning (ArgEML): a Real-life Use Case on Gynecological Cancer

Nicoletta Prentzas[a1], Athena Gavrielidou[a], Marios Neophytou[a] and Antonis Kakas[a]

[a] *University of Cyprus, 1 Panepistimiou Avenue, Nicosia, 2109, Cyprus*

Abstract

This paper studies the application of a general methodology of synthesis of Learning with Explainable Argumentation (ArgEML) to a particular real-life learning problem with the aim to validate the approach and to provide feedback for its further development. The problem concerns that of learning to prognose from a real-life image of a gynecological tumor whether this is benign or malignant. This dataset has already been analyzed and studied using various methods. Our goal is to synthesize and integrate these lower-level statistical and sub-symbolic methods with a symbolic and explainable layer of argumentation. The purpose is not so much to improve on the accuracy of these previous efforts but rather to validate the argumentation approach to ML and to possibly learn from this example how to further automate the search for learning argumentation theories from real-life data. The application of the ArgEML approach was carried out in a semi-automated manner using the Gorgias argumentation framework and the Gorgias Cloud system. We show how using the natural explanations for the predictions (definite or plausible) of the learned argumentation theory we can separate the problem space into groups showing in each such group the basic argumentative tension between arguments for and against the alternatives.

Keywords

Argumentation, Explainable Machine Learning, Explainable AI

1. Introduction

Argumentation is a naturally suitable target language for Machine Learning (ML) model's representation. It offers flexible coverage and prediction notions that are appropriate in the context of learning, where the data from which we are learning may be incomplete and appear to be inconsistent, or simply is inadequate to reveal the full process or theory generating the data. This suitability of argumentation as an umbrella framework in which learning can occur has been exposed recently in [1],[2] where the emphasis is shifted away from achieving optimal predictive accuracy to that of satisfactory or confident accuracy together with the recognition of difficult dilemma cases or sub-domains of the problem where a definite prediction cannot be safely taken. Rather in these cases, the learned theory provides explanations that support the possible alternatives thus helping a subsequent process that is to utilize the learned theory to take a more informed decision. Explanations not only give enhanced meaning to the learned theory but they can also be used during the learning process to guide this, e.g. by focusing on the more relevant features for cases that are ambitious under the current state of the learned theory.

In this paper we present an Argumentation-based Explainable Machine Learning (ArgEML) framework and its application to real-life imaging data on Gynecological Cancer. The ArgEML approach relies on a strong coupling of Learning with Reasoning within a framework of structured argumentation. In this work, we will be using the Gorgias argumentation framework [3] but most of the conceptual elements of the approach can be applied using other structured argumentation frameworks. We show how the ArgEML approach can help us understand the learning problem space by partitioning

[1] Corresponding author, E-mail: nicolep@ucy.ac.cy.

ArgML'22: 1st InternationalWorkshop on Argumentation & Machine Learning, September 13, 2022, Cardiff,Wales, UK

this into sub spaces each of which is classified by its own argumentation framework and argumentative explanations for the prediction.

Our work follows the same motivation as that of several other studies in the literature that explore how to integrate machine learning and argumentative reasoning. A review of these studies up to 2020 can be found in [1], while [4–8] and references therein reflect more recent efforts in this area. All these aim to exploit the flexibility of argumentation and its natural connection to explanation in order to enhance the expressibility and interpretability of a learned function.

The rest of the paper is organized as follows. In section 2 we provide background information about (1) the real-life imaging dataset we will be learning from and (2) the Gorgias argumentation framework we will be using. In Sections 3 and 4 we present the general elements of the ArgEML approach and its application to the real-life dataset. Then in Section 5 we present an analysis of the problem space based on the explanations for prediction that can be drawn from the learned argumentation theory and how this can help in understanding the problem space into its possible subclasses. Finally, Section 6 concludes and discusses future work.

2. Background Information

We briefly describe the dataset for endometrial cancer detection taken from [9]. Then in Section 2.2 we review the basic concepts and terminology of the Gorgias argumentation framework that are relevant for the learning process that we will be using in this paper.

2.1. From Imaging Data to Prognosis

In previous work a hysteroscopy Computer Aided Diagnostic system (CADs) was developed for the early detection of endometrial cancer [9–11]. Regions of Interest (ROIs) were extracted from hysteroscopic images of patients with (1) postmenopausal uterine bleedings and/or suspected endometrial lesions, and, patients with (2) normal endometrium. The ROIs were equally distributed among normal and abnormal cases. The CADs supported the ROIs texture feature extraction in different color systems. A total of 26 texture features were extracted from each color component, using three texture features algorithms: (i) Statistical Features (SF), (ii) Spatial Gray Level Dependence Matrices (SGLDM), and (iii) Gray Level Difference Statistics (GLDS). Our work builds on a combination of SF+SGLDM+GLDS features from the endometrial cancer detection dataset[2], as these are shown in Table 1.The dataset consists of 445 records, 209 (47%) correspond to normal cases (benign) and 236 (53%) to abnormal cases (malignant). Tumor is classified as 0-Malignant or 1-Benign.

Table 1
Dataset Features

Algorithm	Texture Feature	Feature Name	Feature Code
SGLDM (Spatial gray-level dependence matrices)	Homogeneity Entropy	sgldm_homog sgldm_entr	Feature_0 Feautre_1
SF (Statistical features)	Energy Entropy	fos_ener fos_ent	Feature_2 Feuture_3
GLDS (Gray-level difference statistics)	Homogeneity Contrast Energy Entropy Mean	gldm_hom gldm_con gldm_eng gldm_ent gldm_mean	Feature_4 Feature_5 Feature_6 Feature_7 Feature_8

[2] The dataset is available upon request from the authors.

2.2. Gorgias Argumentation Framework

Gorgias[3] is a structured argumentation framework where **arguments** are constructed using a basic (content independent) scheme of **argument rules**. Two types of arguments rules are constructed within a Gorgias argumentation theory: **object-level arguments** and **priority arguments** expressing a preference, or relative strength, between other arguments. The dialectic argumentation process of Gorgias to determine the acceptability (admissibility) of an argument supporting a desirable claim or conclusion typically occurs between **composite arguments** where priority arguments are included alongside object-level arguments in order to strengthen (against counter-arguments) the arguments currently committed to.

In general, argument rules are named associations between a set of premises and a claim or position that these premises are supporting via the argument rule. They have the general form of: "*Argument_Name: Premises* ►*Claim*", where Premises is a set of literal (i.e. positive or negative atomic statement) conditions and Claim is a single literal. They can be chained together to form a support of a desired claim. In their concrete form within the Gorgias system, argument rules are expressed using the syntax of Extended Logic Programming, where an argument rule has the following parametric syntactic form[4]:

$$rule(Argument\_Name, Claim, DefeasiblePremises){:}-Non\_Defeasible\_Premises. \qquad (1)$$

Argument_Name can be any Prolog term with which we parametrically name arguments expressed by this rule. *Claim* is a positive or negative atomic formula (negation in the Gorgias system is written by wrapping the positive atom with ``neg(.)"). *Defeasible_Premises* and *Non_Defeasible_Premises* are conjunctions of positive or negative atomic formulae: the former are executed under Gorgias while the latter directly under Prolog. In the context of learning, the non-defeasible conditions of argument rules are built from the concrete information that we have on the features of our dataset cases. The defeasible conditions allow the opportunity to use conditions for which we do not have complete information or even to invent new conditional predicates (we will not be concerned with the later in this paper).

Example: The Gorgias code below shows two *object-level argument* rules (i.e. r1(), r2()) for and against buying an object with *priority argument* rules (i.e. pr1(), pr2()) between the object-level rules depending on whether we are low on funds.

$$rule(r1(X), buy(X), []) :- need(X).$$
$$rule(r2(X), neg(buy(X)), []){:}- urgency(X, no).$$
$$rule(pr1(X), prefer(r1(X), r2(X)), []).$$
$$rule(pr2(X), prefer(r2(X), r1(X)), []){:}- level\_of\_funds(low).$$

The combination of object-level arguments together with the contextual priority arguments result into a theory that captures the policy of *"Normally, we buy something that we need even if this is not urgently needed. But when we are low on funds we may not buy something for which there is no urgency."*.

In a learning context we would have an underlying process that generates, according to this policy data points by observing if an object is bought or not in different scenarios described by the three features of "need(.), urgency(.,.) and level_on_funds(.)". The task is then to **learn or reconstruct** the above Gorgias theory (or an equivalent form of this).

The coverage and prediction notions for the argumentation-based approach to learning will be build using the standard argumentation reasoning within a structured argumentation like the one of Gorgias. This depends on the central notion of an **acceptable coalition of arguments**, which in the case of the Gorgias framework relates to a (minimal) composite argument that is **admissible**. As in the standard definition of admissibility [12] a composite argument is admissible iff it is conflict free and it attacks back all other composite arguments that attack it.

[3] The Gorgias Argumentation framework was introduced in [13] and extended in [14]. The system of GORGIAS was developed in 2003 and has since been used by several research groups for a variety of real-life applications [3]. Today it is publicly available through Gorgias Cloud as a https://aiasvm1.amcl.tuc.gr:8087/.

[4] In this paper, we will be using the cumbersome internal code syntax of the Gorgias system to present examples. This will help the interested reader to reproduce the learned results and/or apply the learning process to their own learning problems using the open Gorgias Cloud system.

We can then define plausible and definite conclusions or predictions according to whether there exists an admissible composite argument that supports the conclusion of interest, in which case we say the **conclusion is *plausible or possible***. If in addition there exists no admissible composite argument that supports any other conclusion that is in conflict with the conclusion of interest then we say that this is a ***definite conclusion***. Note that it is possible for a conclusion and some other conflicting conclusion to both be plausible conclusions from the same argumentation theory, in which case we say the theory is ***(locally) ambiguous*** and the conclusion forms a ***dilemma*** within the theory.

The above definition of admissibility of composite arguments hinges on the definition of attacks between composite arguments. Informally, a composite argument, D1, attacks another one, D2, iff they are in conflict and the arguments in D1 are rendered by the priority arguments that it contains at least as strong as the arguments contained in D2. The exact technical details of this central notion can be found in the associated references [13, 14]. What is important to note is that attacks can occur at two levels: (1) the object level based on a conflict between statements in the application language or at (2) a (hierarchy of) priority level(s) where the conflict between the two composite arguments refers to a preference between two arguments at a lower level. Accordingly, to build an admissible composite argument we consider attacks at the object level and then include priority arguments to strengthen its object rules against the attacking ones.

To illustrate this, consider in the above example an object, obj1, for which need(obj1), urgency(obj1,no) and level_of_funds(low) all hold true and let us ask the Gorgias query of buy(obj1). This is supported by the simple argument arg1= [r1(obj1)] but this is not admissible as it is attacked by arg2=[r2(obj1),pr2(obj1)] which arg1 does not attack back. To do so we can extend arg1 to form the composite argument arg1'=[r1(obj1), pr1(obj1)]. Both arg1' and arg2 are then admissible indicating that the case of obj1 is a dilemma of the theory having reasons for both to buy it or not to buy it. The ambiguity, *"But when we are low on funds we may not buy something for which there is no urgency."* in the policy, that is represented by this theory, is reflected by the existence of such dilemma cases where the theory cannot make a definite prediction. Indeed, in a learning context the data produced by this policy will contain the ambiguity and it is thus natural for a theory learned from this data to reflect this ambiguity as a reasoned dilemma rather than insist on making a definite prediction for these cases.

3. ArgEML Framework and Methodology

The argumentation-based framework for Explainable Machine Learning (ArgEML) is based on a novel approach to ML that integrates sub-symbolic methods with logical methods of argumentation to provide explainable solutions to learning problems. The goal is to learn argumentation theories from data, using statistical learning techniques, to uncover significant features in developing argumentation theories and represent knowledge as contextual hierarchies within a preference-based structured argumentation framework. In the following subsections we present a conceptual description of the ArgEML approach and a high-level description of its learning process.

3.1. ArgEML approach (conceptual description)

Our ArgEML approach is based on acknowledging the predictive accuracy difficulties in real-life learning problems and the importance of explanations, as a means of understanding the reasoning behind a prediction and providing the domain expert with a tool to take more informed decisions. The approach views the notion of prediction from a different perspective than that of a traditional ML model, by relaxing the requirement of accuracy and introducing the notions of *definite prediction* and *ambiguity*. In this perspective, if we cannot uniquely predict, but can focus the prediction and give justifications for the alternatives, we have a valuable output of learning.

Utilizing argumentation as a framework for explainable decision making we aim at learning contextual hierarchies starting from general and simple statements to more specific ones and structuring these using priorities between them. The learning process is not driven only by strict accuracy but for solutions that would be *sufficiently good* in terms of accuracy compensating with the high-level of explainability of the learned theory. This concept of *sufficiently good* but explainable solution motivates a set of metrics that will govern the learning process. These are defined and explained in Section 3.1.1.

The ArgEML method consists of a high-level iterative learning process that follows a set of semi-automated steps as presented in Figure 1. The first step *initiates* the learning process by (1) deciding the language of the problem and (2) defining the basic contexts of the problem domain in terms of object-level arguments. The *iterative process* starts from an interim evaluation of the initial theory and repeats steps (3) mitigate errors and/or (4) reduce dilemmas until the evaluation results in no further improvement of the learned theory or exit criteria are met. The ArgEML methodology steps are further explained in Section 3.2.

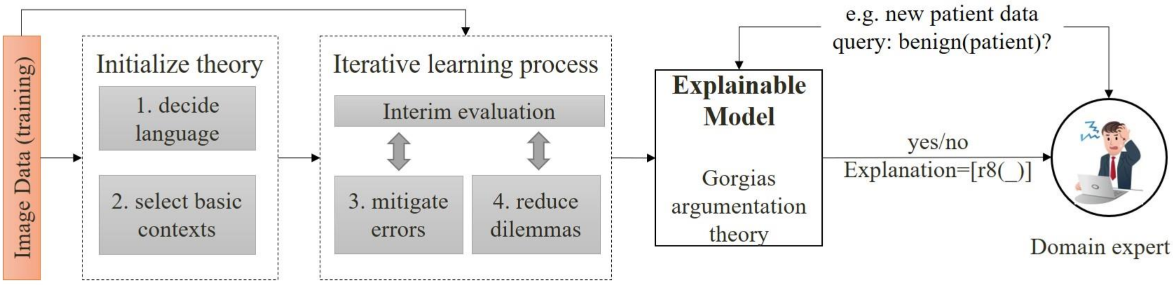

Figure 1: ArgEML conceptual description.

3.1.1. Learning metrics

Table 2
Learning Metrics - Equations

Metric	Equation	
Coverage	$Coverage(D, arg\_i) \leftarrow Objs\_i/N$	(2)
Total Coverage	$TotalCoverage(D, arg\_theory) \leftarrow \bigcup_{i=1}^{m} Coverage(D, arg\_i)$	(3)
Definite Accuracy	$Accuracy(D, arg\_theory) \leftarrow Objs\_acc/N$	(4)
Definite Errors	$Errors(D, arg\_theory) \leftarrow Objs\_err/N$	(5)
Ambiguity	$Ambiguity(D, arg\_theory) \leftarrow Objs\_amb/N$	(6)

Learning metrics are defined in terms of the number of observations or data points ***N*** in the dataset ***D*** that we are learning from, using the equations in Table 2.

- **Coverage**. The *coverage* of an argument *arg_i: Premises_i* ►*Claim* is equal to the number of observations *Objs_i* in a dataset ***D*** that *Premises_i* is true (equation (2) in Table 2).
  The *total coverage* metric for an argumentation theory *arg_theory* with *m* arguments is defined as in equation (3) in Table 2.
- **Definite Prediction**. This metric is related to the predictive accuracy that we normally have in a ML model, but in the ArgEML approach this only applies to the observations for which the theory provides a definite prediction (see Table 2).
  - **Accuracy** or **Definite Accuracy**: is defined as the percentage of the number of observations *Objs_acc* in a dataset *D* that an argumentation theory *arg_theory* provides a definite prediction and the prediction matches the actual target value (equation (4) in Table 2).
  - **Errors** or **Definite Errors**: is defined as the percentage of the number of observations *Objs_err* in a dataset *D* that an argumentation theory *arg_theory* provides a definite prediction but the prediction does not match the target value (equation (5) in Table 2).
- **Ambiguity**. Ambiguity measures the percentage of observations *Objs_amb* in a dataset *D* that an argumentation theory *arg_theory* provides *plausible predictions* (equation (6) in Table 2).
- **Compactness**. This metric relates to the explanation complexity and aims to capture a form of simplicity. It can be defined in a number of ways, in relation to the argumentation theory,

suggesting a compact (small) number of arguments, or, in relation to an individual argument, indicating low complexity of its premises (small number of conditions).

Compact Coverage is one of the major metrics of the ArgEML approach, it combines the metric of *total coverage* and the notion of *compactness*, suggesting a compact argumentation theory with high total coverage.

Given this set of metrics, a solution (theory) can be evaluated using a combination of properties, not simply based on optimal prediction. Hence, a solution can be "sufficiently good" if it provides *compact coverage*, and *acceptable levels* of definite accuracy (or definite errors) and ambiguity with (useful) justifications (explanations), depending on how hard the problem is.

3.2. Integrated learning process - Methodology

Starting from a state of absolute ambiguity, the objective is to learn an argumentation theory that covers all or most observations in a given dataset, eliminates ambiguity, and improves the accuracy of definite predictions, by mitigating the errors. A high-level overview of the methodology is illustrated in Table 3. The first step (step 1) aims at selecting the language (features) to develop the theory. The second step (step 2) concerns the selection of a compact set of arguments to describe the basic contexts of the problem domain. Then, the learning process repeats step 3 and step 4, generates different versions of the argumentation theory, until an exit criterion is met or learning has no further improvement. Exit criteria can be defined using e.g. thresholds for the metrics of *definite errors* (Err_Thold) (or definite accuracy) and *ambiguity* (Amb_Thold).

Table 3

ArgEML Methodology Overview

Learning Step		Goal
Step 1:	Decide the language of the learning problem.	Feature selection
Step 2:	Select the basic contexts of the problem domain.	Compact coverage
Repeat (Steps 3 & 4) until Goal is reached or learning has no further improvement:		
Step 3:	Mitigate the error of individual arguments.	Errors ≤ Err_Thold
Step 4:	Reduce dilemmas between pairs of arguments in conflict.	Ambiguity ≤ Amb_Thold
Evaluation:	Select "sufficiently good" argumentation theory.	Explainable Model

We now briefly describe these steps in *operational* terms.

Initialize theory:

- **Step 1: Decide the language of the problem.**

This step is similar to the data processing step in a machine learning pipeline. It mostly involves independent statistical analysis of the feature set to separate out a set of significant features. Examples include filter methods that select features based on their correlation to the output (target variable). More information on these methods can be found in [15].

- **Step 2: Select the basic contexts of the problem domain.**

In Step 2 we initialize the argumentation theory by building a compact set of object-level arguments (general scenarios) that achieve a high total coverage of the data (Compact Coverage). We can use a combination of learning operators, working directly on the significant features set, or use a surrogate sub-symbolic machine learning algorithm amenable to rule-extraction. For example, we can train a Random Forest or XGBoost model and use a rule-extraction method (e.g. Interpreting Tree Ensembles with inTrees [16]) to construct object-level arguments that form the basic contexts of the argumentation theory.

Iterative Learning Process: The process starts with an interim evaluation of the initial theory and repeats steps 3 and 4 based on the exit criteria.

- **Step 3: Mitigate the error of individual arguments.**

Individual object-level arguments will support erroneously the target conclusion for a number of cases. To mitigate this error, we construct a **defeat argument** against this, which together with a (possibly conditional) **priority argument** will remove a significant number of these erroneous predictions. Step 3 is executed as long as condition Errors > Err_Thold holds. At the end of each execution we generate a new version of the theory and we repeat the iterative learning process (steps 3 & 4).

- **Step 4: Reduce dilemmas between pairs of arguments in conflict.**

In Step 4 we identify the pairs of object-level arguments (and local defeat arguments, if any-that are in conflict to construct **conditional priority arguments** to resolve the conflict in **either way**. Step 4 is executed as long as condition Ambiguity > Amb_Thold holds. At the end of each execution we generate a new version of the theory and we repeat the iterative learning process (steps 3 & 4).

- **Evaluation step: select a "sufficiently good" argumentation theory.**

This step carries out a **global** evaluation, in terms of some overall **information gain,** of the results of the previous local steps in the current theory. Using this we can compare different versions of the argumentation theory and select a sufficiently good improvement of the current theory or terminate. For example, information gain can be calculated using some adopted notion of entropy (as in Decision Trees) based on the values of the new metrics of *compact coverage*, *definite errors* and *ambiguity*. We can use definite errors or definite accuracy interchangeably. While these metrics-based evaluation approaches, also known as objective approaches, are the ones mainly used today, human-centered evaluation is of equal importance with studies suggesting a more active role of the end user in the process [17][18].

4. ArgEML applied to Cancer Prognosis

In this section, we illustrate the (semi-automated) application of the ArgEML methodology on the dataset described in Section 2 for the classification of hysteroscopy images and the endometrial cancer detection. At the beginning of the process Err_Thold is set to 20% and Amb_Thold to 30%.

- **Step 1: Decide the language of the problem.**

We used a set of features from [9] as show in Table 1. The dataset of 445 observations was divided into training and test sets with 400 (90%) and 45 (10%) observations respectively. While techniques like cross-validation are usually employed at this step we simplified this process to focus on the validation of the ArgEML approach.

- **Step 2: Select the basic contexts of the problem domain.**

We followed the rule-extraction method, trained a Random Forest model using the training set and extracted a number of decision rules from the model. Then we selected a compact list of these rules, to cover most of the observations in the training set, to create the basic object-level arguments of the theory. This gave us an initial version of the theory with a small number of low-complexity arguments, as show in Table 5[5], and a total coverage of 99.75%. At this point we noticed that each data point is covered (roughly) twice by this initial theory and hence its predictive accuracy as a whole is low.

Table 4

Object-level arguments.

Argument	Premises	Claim	C	A	E
r4(X)	$gldm\_mean > 1.65\ AND\ sgldm\_entr > 5.25$	benign(X)	48%	79%	21%
r6(X)	$gldm\_con > 4.89\ AND\ fos\_ener \leq 0.06$	benign(X)	50%	78%	22%
r8(X)	$gldm\_con \leq 5.03\ AND\ gldm_{ent} > 1.30$	malignant(X)	50%	72%	28%
r10(X)	$gldm\_mean \leq 1.65\ AND\ sgldm\_hmog > 0.45$	malignant(X)	50%	72%	28%

C: Coverage. A: Accuracy. E: Error.

- **Step 3: Mitigate the error of individual arguments.**

The object-level arguments selected in Step 2 were further analyzed using the properties of Coverage, Accuracy and Error as shown in Table 4. For each argument in the list (r4, r6, r8, r10) we

[5] The numerical conditions in these argument rules can be discretized, e.g. into low, medium and high, to help with the readability of the explanations generated from these. This matter is beyond the scope of this paper.

isolate the observations in the training set that the argument covers and try to learn a new set of conditions (premises) to construct a **defeat argument**. For example, for the argument r8, we examined the 201 (50%) observations from the training set, using a feature frequency distribution operator, looking for new conditions to support the contratictive conclusion of "benign(X)". We learned the defeat argument r8b defined as follows:

$$rule(r8b(X), benign(X), [\quad]){:}-gldm\_hom > 0.50\ AND\ fos\_ener \leq 0.05.$$

In the context of mitigating errors, defeat arguments are created together with the corresponding priority arguments to ensure local correction of the error. Therefore, for the arguments r8, r8b we added the priority argument pr3:

$$rule\big(pr3(X), prefer\big(r8b(X), r8(X)\big), [\quad]\big).$$

Furthermore, to avoid side effects of defeat arguments on other object-level arguments we can add further priority rules that make these weaker than other conflicting arguments. For argument r8b we have therefore added:

$$rule\big(pr7(X), prefer\big(r10(X), r8b(X)\big), [\quad]\big).$$

The revised properties of Accuracy and Error for the initial object-level arguments is shown in Table 5. Step 3 improved the quality of the object-level arguments by reducing their Errors and satisfying the threshold of 20%.

Table 5

Object-level argument's properties revised, after execution of Step 3.

Argument	Claim	C	A	E
r4(X)	benign(X)	48%	83%	17%
r6(X)	benign(X)	50%	82%	18%
r8(X)	malignant(X)	50%	82%	18%
r10(X)	malignant(X)	50%	80%	20%

- **Step 4: Reduce dilemmas between pairs of arguments in conflict.**

During this step we examined all pairs of contradictory object-level arguments created in Step 2. This examination resulted in the following list of {*(arguments pair=number of dilemmas)*}: *{pair(r4(X), r8(X))=5, pair(r4(X), r10(X))=0, pair(r6(X), r8(X))=7, pair(r6(X), r10(X))=8}.* If a pair of arguments was in conflict then we tried to eliminate the dilemma using priority arguments, making object-level arguments stronger under a particular set of conditions. For each pair of contradictory object-level arguments we isolate the observations in the training set that both arguments covered, and try to find new conditions, using a frequency distribution operator, to construct priority arguments in favor of each contradictory conclusion. For example, for the pair of arguments r6(X), r10(X), we see that the majority of these dilemma cases belong in the class of benign. Therefore, we added a general priority argument, to express this preference.

$$rule\big(pr12(X), prefer\big(r6(X), r10(X)\big), [\quad]\big).$$

Secondly, we searched for a condition or a set of conditions under which argument r10 is stronger than r6, and constructed the preference argument pr13:

$$rule\big(pr13(X), prefer\big(r10(X), r6(X)\big), [\quad]\big){:}-sgldm\_homog > 0.454\ AND\ sgldm\_homog < 0.46$$

together with the higher-order preference of this specific preference over pr12:

$$rule\big(c6(X), prefer\big(pr13(X), pr12(X)\big), [\quad]\big).$$

At the end of Step 4 all dilemmas between the basic object-level arguments (r4,r6,r8,10) were resolved while other dilemmas, between pairs of defeat arguments and object-level arguments, may still remain. The resulting argumentation theory is provided as a Gorgias file in the Appendix. This theory was considered "sufficiently good" on the training set. It was then evaluated on the test set with similar results, as shown in Table 6.

Table 6
Argumentation theory assessment on the Training and Test sets.

Metric	Training set Assessment	Test set Assessment
Compact coverage	Acceptable (TC: 99.75%)	Acceptable (TC: 100%)
Definite accuracy*	72%	71%
Definite errors	18%	18%
Ambiguity	10%	11%

*TC: Total Coverage.

5. Explainable Analysis of the Problem Space

Using argumentation as the coverage notion for ML naturally affords the provision of explanations alongside the prediction of the learned output structure. Predicting the label of a case is carried out via the existence of an acceptable set of arguments that supports the prediction. The acceptability of this set of arguments can then be unraveled to produce an explanation that contains information both at the level of the basic **attributive support** of the prediction claim and at the level of the relative strength of the claim **in contrast** to other possible alternative claims. For the case of the Gorgias framework, this process of extracting natural explanations is facilitated by the form of the composite admissible arguments that are constructed as **Internal Explanations** by the Gorgias system and returned along with its answer to a query. Let us illustrate this kind of **application level explanations,** generated automatically in Cloud Gorgias, by supposing that we have the Gorgias internal explanation *[pr4 (101),r4(101),r6(101)]* for predicting that case 101 is benign. From this, we can generate the explanation illustrated in Table 7.

Table 7
Application level explanation.

The statement "benign (101)" ***is supported by*** *{gldm_mean(101)>1.65 & sgldm_entr(101)>5.25}*[a]. This reason is ***strengthened against*** the reason of *{gldm_mean>1.66 & gldm_hom <= 0.45}*[b] supporting "malignant (101)" ***by*** *{gldm_con (101)>4.89 & fos_ener(101) <0.06}*[c].

[a] Premises of r4. [b] Premises of r4b. [c] Premises of r6.

We can see that this contains an **attributive** part, giving the basic reasons on which, the prediction is supported (or else answering “why this prediction”) as well as a **contrastive** part, which gives additional reasons that strengthen the basic reason against reasons supporting the opposite prediction (or else answering “why-not a different prediction”). Such explanations provide a high-level of interpretability of the learned theory that facilitates its evaluation through experts who would be able to judge the prognosis results based not merely on the final result but on their accompanied explanations and, in fact, provide useful feedback at the level of the explanation. We can then improve the learned model through a new learning phase from such new data cases which are further annotated by the argumentative explanation that supports their labels (c.f. the learning method of [19]).

Furthermore, and perhaps more importantly, the Gorgias internal explanations can help us analyze the problem space and understand how this can be structured into different sub-parts. We can use these internal explanations of composite arguments to **partition the problem space into (equivalence) groups**, where each group is characterized by a unique type or pattern of explanation. In our prognosis application, we have found that the training data space is partitioned into a set of groups as shown in Table 8. In Groups 1-4, the prediction of the learned argumentation theory is definite whereas in groups 5 and 6 the learned theory is in a dilemma, i.e. it returns admissible arguments supporting either of the two possible outcomes of the prediction. We can use this partitioning to grade our confidence in the prediction of the theory depending on the group that a new case may fall. For example, we might be more confident for a prediction that falls in group 3 over other predictions that fall in groups 1 or 4.

As mentioned above, each group is defined by the unique pattern of the Gorgias internal explanation returned for all members of the group. From this we can extract two relevant pieces of information that describe the group: (1) the sub-space of features that concerns this group and (2) the arguments in the learned theory that are active in this sub-space as well as the active attacks between them. Combining these two pieces of information, we can understand how the learned theory captures the decision problem for each group by constructing the argumentation framework pertaining to each group.

Table 8
Subgroups of Data identified by Gorgias Argumentative Explanations.

Group ID	Gorgias Explanation(s)	Number of Cases	Accuracy/ Dilemma
1	E1=[pr7(_),r10(_),r8(_)]	16	71%
2	E2=[r8(_)]	142	78%
3	E3=[pr4(_),r4(_),r6(_)]	170	83%
4	E4=[r4(_)]	14	69%

5	E51=[pr15(_),pr8(_),r10b(_),r8b(_)]
	E52=[pr19(_),pr7(_),r10(_),r8(_)]
		28	Dilemma

6	E61=[pr16(_),pr4(_),r4(_),r6(_)]
	E62=[pr14(_),pr3(_),r4b(_),r6b(_)]
		14	Dilemma

Others	----	16	----

Let us present this for group 3 whose internal Gorgias explanation, is the composite argument **E3**= [pr4(.), r4(.), r6(.)]. From this, we can recognize that the active arguments involved are: **A4**= [r4(.)], **A6**= [r6(.), pr4(.): r6(.) > r4b(.)] and **B4b**= [r6b(.), pr3(.): r4b(.) > r4(.)], together with the following attacks between these as shown in Figure 2 (left).

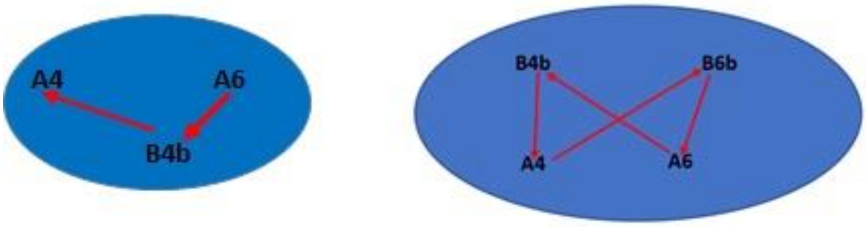

Figure 2. Argumentation Frameworks for Group 3 (left) and Group 6 (right)

Given this argumentation framework we see that the only admissible subsets are {A6} and {A4, A6} (the latter being E3), and hence in this group we have a definite prediction of benign. Note that although the prediction within this sub-part of the problem can be supported simply by the argument A6, this actually forms another sub-part of the problem, a small sub-group in "Others" of Table 7. Here in group 3, we see that the role of A6 is different, namely it comes to the defense of A4 against its defeater attack of B4b. The two arguments of A4 and A6 supporting the same conclusion of benign aggregate together to give a more informative explanation (see above).

Similarly, the argumentation framework corresponding to group 6 is shown in Figure 2 (right). This has two admissible subsets of composite arguments, D1={A4, A6}, and D2={B4b, B6b} supporting opposite predictions, indicating that this sub-part of the problem is identified by the theory as a "difficult case''. The learned theory though is not agnostic. It provides a **contrastive explanation** for each possible prediction.

6. Conclusions and Future Work

We have presented an integrated approach of Machine Learning with Argumentation and shown how this has been applied to a real-life problem of learning from images of endometrial cancer. The same method has been applied on other medical imaging data, e.g. on brain images for Alzheimer [19], and more recently on images relating to multiple sclerosis. We have shown how the explainability of such an argumentation-based approach to ML can help us understand and structure the learning problem space into meaningful sub-spaces.

The proposed ArgEML learning process can be executed in different modes, from semi-automated and hybrid with the help of external statistical and other ML modules (as followed in this paper) to a fully automated process starting from the data and carrying out iteratively the learning operator steps.

In particular, the learning operators of mitigation of errors and resolution of dilemmas can be automated with various parameters, depending on the features of the learning problem at hand. The long-term goal of our work is to automate this process of learning starting from the data to the final argumentation theory. While argumentation provides a natural link to explanations, a major challenge in this task of automating fully the learning process, is to consider how these explanations can meet the various qualities of explanations, as well as the involvement of the domain expert in the evaluation process, particularly in the context of Human-centric AI. The quality of explanations needs to drive the learning process as much as the prediction accuracy.

7. Acknowledgements

Part of this work was undertaken under the University of Cyprus internal project, Integrated Explainable AI (IXAI) for Medical Decision Support, ARGEML 8037P-22046. This study is also partly funded by the project ‘Atherorisk’ “Identification of unstable carotid plaques associated with symptoms using ultrasonic image analysis and plaque motion analysis”, code: Excellence/0421/0292, funded by the Research and In-novation Foundation, the Republic of Cyprus.

8. References

[1] Kakas A, Michael L. Abduction and Argumentation for Explainable Machine Learning: A Position Survey. *arXiv*; abs/2010.1, http://arxiv.org/abs/2010.12896 (2020).

[2] Prentzas N, Nicolaides A, Kyriacou E, et al. Integrating machine learning with symbolic reasoning to build an explainable ai model for stroke prediction. In: *Proceedings - 2019 IEEE 19th International Conference on Bioinformatics and Bioengineering, BIBE 2019*. Institute of Electrical and Electronics Engineers Inc., 2019, pp. 817–821.

[3] Kakas AC, Moraitis P, Spanoudakis NI. GORGIAS: Applying argumentation. *Argument Comput* 2019; 10: 55–81.

[4] Albini E, Lertvittayakumjorn P, Rago A, et al. *DAX: Deep Argumentative eXplanation for Neural Networks*. 2020.

[5] Rosenfeld A. Better metrics for evaluating explainable artificial intelligence. In: *Proceedings of the International Joint Conference on Autonomous Agents and Multiagent Systems, AAMAS*. 2021, pp. 45–50.

[6] Wardeh M, Coenen F, Capon TB. PISA: A framework for multiagent classification using argumentation. *Data Knowl Eng* 2012; 75: 34–57.

[7] Bench-Capon T. Using Issues to Explain Legal Decisions, http://arxiv.org/abs/2106.14688 (2021).

[8] Prakken H, Ratsma R. A top-level model of case-based argumentation for explanation: Formalisation and experiments. *Argument Comput* 2022; 13: 159–194.

[9] Neofytou MS, Tanos V, Constantinou I, et al. Computer-aided diagnosis in hysteroscopic imaging. *IEEE J Biomed Heal Informatics* 2015; 19: 1129–1136.

[10] Neofytou MS, Tanos V, Pattichis MS, et al. A standardised protocol for texture feature analysis of endoscopic images in gynaecological cancer. *Biomed Eng Online*; 6. Epub ahead of print 2007. DOI: 10.1186/1475-925X-6-44.

[11] Neofytou MS, Pattichis MS, Pattichis CS, et al. Texture-based classification of hysteroscopy images of the endometrium. In: *Annual International Conference of the IEEE Engineering in Medicine and Biology - Proceedings*. 2006, pp. 3005–3008.

[12] Dung PM. On the acceptability of arguments and its fundamental role in nonmonotonic reasoning, logic programming and n-person games. *Artif Intell* 1995; 77: 321–357.

[13] Kakas A, Mancarella P, Dung PM. The Acceptability Semantics for Logic Programs. In: *Logic Programming*. The MIT Press. Epub ahead of print 2019. DOI: 10.7551/mitpress/4316.003.0051.

[14] Kakas A, Moraïtis P. Argumentation Based Decision Making for Autonomous Agents. In: *Proceedings of the International Conference on Autonomous Agents*. 2003, pp. 883–890.

[15] Chandrashekar G, Sahin F. A survey on feature selection methods. *Comput Electr Eng* 2014;

40: 16–28.

[16] Deng H. Interpreting tree ensembles with inTrees. *Int J Data Sci Anal* 2019; 7: 277–287.

[17] Zhou J, Gandomi AH, Chen F, et al. Evaluating the quality of machine learning explanations: A survey on methods and metrics. *Electronics (Switzerland)* 2021; 10: 1–19.

[18] Bruckert S, Finzel B, Schmid U. The Next Generation of Medical Decision Support: A Roadmap Toward Transparent Expert Companions. *Front Artif Intell*; 3. Epub ahead of print 2020. DOI: 10.3389/frai.2020.507973.

[19] Achilleos KG, Leandrou S, Prentzas N, et al. Extracting Explainable Assessments of Alzheimer's disease via Machine Learning on brain MRI imaging data. In: *Proceedings - IEEE 20th International Conference on Bioinformatics and Bioengineering, BIBE 2020*. 2020, pp. 1036–1041.

Appendix

```
:- dynamic feature0/2, feature1/2, feature2/2, feature3/2, feature4/2, feature5/2, feature6/2, feature7/2,
        feature8/2.[6]

complement(malignant(Tumor), benign(Tumor)).
complement(benign(Tumor), malignant(Tumor)).

rule(r4(Tumor), benign(Tumor),[]):-feature8(Tumor,Value),Value>1.65,
feature1(Tumor,Value2),Value2>5.25.
rule(r4b(Tumor), malignant(Tumor),[]):-feature8(Tumor,Value),Value>1.66,
feature4(Tumor,Value2),Value2=<0.45.
rule(pr3(Tumor), prefer(r4b(Tumor), r4(Tumor)),[]).

rule(r8(Tumor), malignant(Tumor),[]):-feature5(Tumor,Value),Value=<5.03,
feature7(Tumor,Value2),Value2>1.30.
rule(r8b(Tumor), benign(Tumor),[]):-feature4(Tumor,Value),Value>0.50,
feature2(Tumor,Value2),Value2=<0.05.
rule(pr5(Tumor), prefer(r8b(Tumor), r8(Tumor)),[]).

rule(pr1(Tumor), prefer(r4(Tumor), r8(Tumor)),[]).
rule(pr2(Tumor),prefer(r8(Tumor), r4(Tumor)),[]):-feature0(Tumor,Value),Value>0.445,
feature4(Tumor,Value2),Value2>0.445.
rule(c1(Tumor),prefer(pr2(Tumor),pr1(Tumor)),[]).

rule(r6(Tumor), benign(Tumor),[]):-feature5(Tumor,Value),Value>4.89,
feature2(Tumor,Value2),Value2=<0.06.
rule(r6b(Tumor), malignant(Tumor),[]):-feature7(Tumor,Value),Value>1.67,
feature1(Tumor,Value2),Value2=<5.93, feature6(Tumor,Value3),Value3=<0.19.
rule(pr14(Tumor), prefer(r6b(Tumor), r6(Tumor)),[]).

rule(pr4(Tumor), prefer(r6(Tumor), r4b(Tumor)),[]).
rule(pr16(Tumor), prefer(r4(Tumor), r6b(Tumor)),[]).

rule(r10(Tumor), malignant(Tumor),[]):-feature8(Tumor,Value),Value=<1.65,
feature0(Tumor,Value2),Value2>0.45.
rule(r10b(Tumor), benign(Tumor),[]):-feature4(Tumor,Value),Value>0.50,
feature0(Tumor,Value2),Value2>0.50, feature3(Tumor,Value3),Value3>3.31.
rule(pr15(Tumor), prefer(r10b(Tumor), r10(Tumor)),[]).

rule(pr12(Tumor), prefer(r6(Tumor), r10(Tumor)),[]).
rule(pr13(Tumor),prefer(r10(Tumor), r6(Tumor)),[]):-feature0(Tumor,Value),Value>0.454,
feature0(Tumor,Value2),Value2<0.46.
rule(c6(Tumor),prefer(pr13(Tumor),pr12(Tumor)),[]).

rule(pr19(Tumor), prefer(r8(Tumor), r10b(Tumor)),[]).
rule(pr7(Tumor), prefer(r10(Tumor), r8b(Tumor)),[]).

rule(pr40(Tumor), prefer(r6(Tumor), r8(Tumor)),[]).
rule(pr41(Tumor),prefer(r8(Tumor), r6(Tumor)),[]):-feature0(Tumor,Value),Value>0.45.
rule(c40(Tumor),prefer(pr41(Tumor),pr40(Tumor)),[]).
```

[6] Predicates feature0,feature1,..,feature8 correspond to feature-names is shown in Table 1.