Interpretable Hybrid Machine Learning Models Using FOLD-R++ and Answer Set Programming

Sanne Wielinga 852641381
Model-Based Artificial Intelligence (IM1202), Open University of the Netherlands

December 5, 2024

Introduction

In healthcare and various other domains, machine learning (ML) has become an increasingly indispensable tool for predictive analytics and decision support (Tonekaboni, Joshi, McCradden, and Goldenberg (2019)). Advanced ML models, particularly black-box models such as neural networks and ensemble methods, have shown impressive predictive capabilities. However, a challenge still remains: the lack of interpretability and explainability. This limitation slows down the adoption of ML models in applications where understanding the reasoning behind a prediction is as important as the prediction itself (Arrieta et al. (2020)). In fields like medicine, decisions informed by ML models can have profound implications on patient outcomes. The ability to explain and justify predictions is therefore important for trust, accountability, and informed decision-making (Doshi-Velez and Kim (2017)).

Answer Set Programming (ASP), a form of declarative programming, has the potential to increase the interpretability of ML models. ASP allows knowledge to be represented through logical rules that are transparent and human-understandable (Lifschitz (2019)). The FOLD-R++ algorithm uses ASP to learn default rules with exceptions from data (Wang and Gupta (2022)). These rules can complement black-box models by offering explanations for their predictions.

However, integrating ASP-derived rules with black-box ML models without changing their internal mechanisms remains challenging. This integration is important for preserving the predictive performance of black-box models while improving their interpretability. Most existing approaches focus on integrating symbolic reasoning with neural networks, often requiring modifications to the learning algorithms or network architectures (Garcez et al. (2019); Manhaeve, Dumancic, Kimmig, Demeester, and De Raedt (2018); Yang, Ishay, and Lee (2023)). In contrast, there is a lack of methods that integrate symbolic reasoning with other types of black-box ML models, such as support vector machines or ensemble methods.

This study proposes a hybrid approach that fills this gap by combining the strengths of both methods. Specifically, logical rules derived from FOLD-R++ are combined with a variety of black-box ML models to create a system that delivers accurate predictions accompanied by explanations, without modifying the underlying ML models.

The proposed approach involves several steps. First, initial black-box ML classifiers are trained on multiple medical datasets to evaluate their performance and identify areas of weakness. Next, FOLD-R++ is applied to the same datasets to learn interpretable logical rules, focusing on capturing patterns that the ML models may have missed or misclassified. These learned rules are subsequently integrated with the ML models to form a hybrid system. The ASP component aims to correct errors made by the ML models, especially in instances where they are uncertain or less accurate.

The effectiveness of the hybrid approach is evaluated using performance metrics, and results before and after integration are compared. Statistical tests are conducted to determine the significance of any improvements. Interpretability is emphasized by providing detailed explanations based on logical rules, analyzing error patterns, and raking the importance of ASP rules based on their contribution to correcting ML errors.

The following research questions guide this study:

RQ1: Does integrating interpretable ASP rules derived from FOLD-R++ improve the predictive performance of black-box ML models across various medical datasets?

RQ2: How does the hybrid model of ML and ASP improve the interpretability of predictions?

The remainder of this paper is organized as follows: the Background section reviews related work on interpretability and hybrid models. The methodology is outlined in the Methods section, including data preparation, model training, and hybrid model implementation. Experimental findings are presented in the Results section. The findings and implications are discussed in the Discussion section. Finally, the Conclusion section concludes the study and suggests directions for future research.

By demonstrating how logical reasoning can improve both the performance and interpretability of various black-box ML models, particularly in the healthcare domain, this project contributes to the development of more reliable and transparent AI systems.

Background

The development of machine learning models that are both accurate and interpretable remains a foundational challenge in artificial intelligence (Guidotti et al. (2018)). While advanced ML techniques, such as deep neural networks and ensemble methods, achieve superior performance across diverse applications, they often function as "black-box" models, lacking transparency and making it difficult to understand and trust their decisions. This issue is particularly problematic in areas such as healthcare, where interpretability is important for ethical considerations, regulatory compliance, and fostering trust among clinicians and patients (Tonekaboni et al. (2019)).

Interpretability and Explainability

Interpretability refers to the extent to which a human can comprehend the cause of a decision made by a model, while explainability involves the extent to which the internal mechanics of a model can be explained in human-understandable terms (Gilpin et al. (2018); Xu et al. (2019)). For example, an interpretable model such as a decision tree provides a direct mapping from inputs to outputs. In contrast, explainability techniques apply to otherwise opaque models to explain their behavior post-hoc.

In healthcare, interpretability is vital. Clinicians must understand how a model arrived at a specific diagnosis to trust and effectively use it in decision-making (Tonekaboni et al. (2019)). Lack of transparency can lead to mistrust or rejection of ML systems (Arrieta et al. (2020)). Moreover, regulatory bodies increasingly require explanations for automated decisions, especially when they impact patient care (Palaniappan, Lin, and Vogel (2024)).

To address these issues, researchers have proposed various strategies. These include developing inherently interpretable models such as decision trees and rule-based systems (Rudin (2019)), and using post-hoc techniques like LIME (Ribeiro, Singh, and Guestrin (2016)) and SHAP (Lundberg (2017)) to explain predictions. However, there is often a trade-off between interpretability and performance. Simple models may not capture complex patterns in data, while complex models may be too opaque. Additionally, highly detailed explanations can overwhelm users (Xu et al. (2019)).

Symbolic Logic and Answer Set Programming

Symbolic logic provides a framework for building interpretable models by encoding knowledge in structured, human-readable formats. Answer Set Programming (ASP) is a form of declarative programming that combines logic programming with non-monotonic reasoning to effectively represent complex relationships and constraints (Lifschitz (2019)). ASP allows for the expression of knowledge through logical rules and facts, closely resembling human reasoning.

ASP is particularly suitable for tasks where interpretability is important. Applications of ASP span diverse areas, including planning, scheduling, and bio-informatics (Erdem, Gelfond, and Leone (2016)). In healthcare, ASP has been applied to model and reason about clinical guidelines (Spiotta, Terenziani, and Dupré (2017)), and solve scheduling problems (Erdem et al. (2016)).

FOLD-R++ Algorithm

The FOLD-R++ algorithm, introduced by Wang and Gupta (2022), extends the First-Order Logical Decision tree (FOLD) algorithm to learn default rules with exceptions from relational data, representing them in a form that is both human-readable and suitable for reasoning with ASP. FOLD-R++ generates default rules capturing general patterns in the data, along with exceptions accounting for special cases or anomalies. The algorithm operates by recursively partitioning the data to construct a decision tree, similar to algorithms like ID3 or C4.5. However, FOLD-R++ transforms the decision tree into a set of default logical rules with exceptions, represented in ASP.

The algorithm starts with the entire dataset and considers all possible literals (attribute-value pairs or relational literals) that can be used to split the data. At each node, the literal that best separates the positive examples from the negative ones is chosen according to a heuristic. This literal becomes part of the condition in the rule. When a rule does not perfectly classify the data, the algorithm identifies exceptions to the rule. These exceptions are themselves induced as sub-rules using the same recursive process. Mechanisms for pruning unnecessary rules or exceptions are included to prevent overfitting. The final set of rules and exceptions is translated into an ASP program of the form:

```
label(X, Class) :- conditions(X), not exceptions(X).
```

This means an instance `X` belongs to a class if it satisfies certain conditions and none of the exceptions apply.

Although newer algorithms in the same family, like FOLD-SE and FOLD-RM, offer advanced features, they were not selected due to accessibility constraints and alignment with the research focus. Specifically, FOLD-SE lacks a publicly available repository which limits integration into automated workflows. FOLD-R++ was chosen for its proven effectiveness and compatibility with the experimental setup.

Hybrid Models

Hybrid models that combine machine learning with symbolic reasoning aim to use the strengths of both approaches. Statistical ML models excel at capturing complex patterns and representations from large amounts of data but often lack interpretability. Symbolic reasoning provides transparency and the ability to incorporate domain knowledge but may struggle with noisy or high-dimensional data.

Most existing hybrid approaches focus on integrating symbolic reasoning with neural networks, creating neuro-symbolic systems. For instance, Garcez et al. (2019) discuss neuro-symbolic AI to combine learning and reasoning capabilities. Manhaeve et al. (2018) introduce DeepProbLog, integrating probabilistic logic programming with deep learning. In the context of ASP, Yang et al. (2023) propose NeurASP, a framework that combines neural networks with ASP.

These approaches often require modifications to the learning algorithms or network architectures, which can be complex and computationally intensive. They primarily focus on neural networks and do not address integration with other types of black-box ML models. There is a gap in research concerning the integration of symbolic reasoning with traditional classifiers and ensemble methods without altering their internal mechanisms.

This study addresses this gap by integrating interpretable ASP rules from FOLD-R++ with a variety of black-box ML classifiers, including support vector machines, random forests, k-nearest neighbors, and neural networks. This hybrid approach does not require changing the ML models, thus preserving their performance. The ASP component adds a layer of interpretability by offering human-understandable explanations for the predictions.

Methods

This section outlines the methodology used to develop and evaluate the hybrid model that integrates black-box ML models with interpretable rules generated by FOLD-R++ using ASP. The approach includes data preparation, model training, hybrid model implementation, experimental setup, and evaluation metrics.

Data Preparation

Five medical datasets from the UCI Machine Learning Repository (Kelly, Longjohn, and Nottingham (n.d.)) were selected to evaluate the performance of the hybrid model:

- **Heart Disease:** Contains 303 instances with 14 commonly used attributes. It is used to predict the presence of heart disease in patients based on medical measurements.
- **Autism Screening Adult:** Includes 704 instances with 21 attributes, used to predict whether an individual is likely to have autism spectrum disorder based on screening test scores and demographic information.
- **Breast Cancer Wisconsin:** Consists of 569 instances with 30 numerical features computed from digitized images. The goal is to classify tumors as malignant or benign.
- **Ecoli:** Contains 336 instances with 8 attributes, used to classify protein localization sites within a cell.
- **Chronic Kidney Disease:** Contains 400 instances with 24 attributes, used to predict the presence of chronic kidney disease in patients based on clinical and laboratory findings.

Each dataset underwent preprocessing to handle missing values, encode categorical variables, and scale numerical features where necessary. Stratified sampling was used to split the datasets into training and testing sets to maintain class distribution. A different random seed was used for each experiment.

FOLD-R++

FOLD-R++ is an inductive logic programming algorithm that generates default rules with exceptions from data using ASP (Wang and Gupta (2022)). In this study, original FOLD-R++ code was used and slightly refactored to fit the experimental setup. This included adding new functions to transform data into ASP-compatible formats and interface with Clingo. Additionally, a wrapper was created to handle the training of FOLD-R++, conversion of induced rules to ASP syntax, and prediction using Clingo. The training data was transformed into a format compatible with FOLD-R++, where each instance was represented as a set of logical facts with attributes and their values forming predicates. For example, an attribute-value pair such as `age = 45` was converted into a predicate `age(X,45)`.

FOLD-R++ induced rules of the form:

```
label(X, Class) :- Conditions(X), not Exceptions(X).
```

Here, `Conditions(X)` represent the conditions under which an instance `X` belongs to a particular class, and `Exceptions(X)` are exceptions to these rules. The algorithm recursively constructs these rules and exceptions. The induced rules were converted into ASP syntax compatible with the Clingo solver (Gebser, Kaminski, Kaufmann, and Schaub (2019)). Numeric values were scaled by a factor 10 to handle the limitations of Clingo with floating-point numbers. The scaling was applied consistently to keep relationships between variables valid. Clingo was then used to apply the logical rules to the test data, reasoning over the facts representing each instance and the induced rules to infer the class labels.

ML Models

Four black-box ML models were used to provide baseline predictive performance:

- **Random Forest (RF):** An ensemble method that builds multiple decision trees using bootstrap aggregation (bagging) and random feature selection. It reduces variance and improves generalization.
- **Support Vector Machine (SVM):** A model that finds the optimal hyperplane that best separates classes in a high-dimensional space. The radial basis function (RBF) was used to capture non-linear relationships.
- **K-Nearest Neighbors (KNN):** A non-parametric method that classifies instances based on the majority class among the k-nearest neighbors in the feature space.
- **Neural Network (MLPClassifier):** A multi-layer perceptron that learns complex non-linear relationships through backpropagation. A configuration with one hidden layer containing 100 neurons and the ReLU activation function was used.

Default hyperparameters were used for all models unless adjustments were necessary for convergence or performance. Random seeds were set for model initialization and data splitting to ensure reproducibility across experiments. For models that provide probability estimates (e.g., RF, SVM with `probability=True`, MLP), the `predict_proba` method was used to obtain confidence scores for each class. For KNN, probabilities were derived from the proportion of neighbors belonging to each class.

Hybrid Model Implementation

The hybrid model integrates predictions from the ML models with the interpretable rules generated by FOLD-R++ using ASP. First, predictions and confidence scores from the ML model on the test set were obtained. Clingo was then used to apply the induced rules to the test instances and generate predictions.

A fixed confidence threshold of 0.6 was used to determine when to rely on the prediction of the ML model versus the ASP rules. If the confidence score of the ML model for a prediction was above 0.6, the ML prediction was used; otherwise, the prediction of the ASP rule was used. This approach ensures that the ASP rules intervene specifically when the ML model is uncertain. During development, both dynamic and fixed confidence thresholds were experimented with, but the dynamic threshold led to overfitting on the training data.

For each prediction made using the ASP rules, explanations were generated based on the specific rules that fired. These explanations improve interpretability by providing a transparent reasoning path for the prediction. The ASP component aims to correct potential errors made by the ML models and offers insights that are not available from the black-box models alone.

Experimental Setup

Experiments were conducted to evaluate the performance of the ML models, the FOLD-R++ model, and the hybrid model. Ten experiments were conducted for each dataset and model combination. In each experiment, the dataset was split into training and testing sets using stratified sampling with an 80-20 split, using different random seeds to vary the splits and model initialization.

Performance metrics were averaged over the ten experiments, and standard deviations were calculated. The following metrics were assessed: accuracy, precision, recall, and F1 score.

Paired t-tests were conducted to determine whether the improvements observed with the hybrid model were statistically significant compared to the standalone ML models. A p-value threshold of 0.05 was used to determine statistical significance. If the ML model and hybrid model had the same accuracy in all experiments, the differences were zero, and the paired t-test could not be performed.

Due to variability in the induced ASP programs across different experiments - likely caused by randomness in data splitting - only the ASP programs and important rules from model-dataset combinations that showed statistically significant improvements were saved. The importance of each ASP rule was determined based on how frequently it contributed to correcting errors made by the ML models. Explanations were generated for instances where the hybrid model corrected the prediction of the ML model, using the proof trees from the FOLD-R++ algorithm.

The implementation was done using Python, with libraries such as scikit-learn for ML models, SciPy for statistical tests, and clingo for ASP solving. Pandas and NumPy were used for data manipulation and numerical computations. The FOLD-R++ library was used to induce logical rules from data. The project was modularized into separate components for data loading, model training, hybrid integration, and evaluation to improve readability.

The source code is available on GitHub[1].

Results

This section presents the experimental results of evaluating the hybrid model that integrates black-box ML classifiers with interpretable rules derived from FOLD-R++ using ASP. The performance of the ML models and the hybrid model is compared across multiple medical datasets. Additionally, the role of ASP in improving interpretability and correcting errors made by the ML models is demonstrated through the application of logical rules and case studies.

Table 1 summarizes the average accuracy and F1 score of the ML models and the hybrid models across all datasets. Standard deviations are included to indicate variability in the results.

The results indicate that, across several datasets, the hybrid models demonstrated an improvement in accuracy and F1 score over the standalone ML models. Notably, the SVM classifier showed significant improvements when combined with the ASP rules in the hybrid model. In some cases, other models such as Random Forest and MLPClassifier also showed significant improvements.

Statistical Significance

Paired t-tests were conducted to determine whether the improvements observed with the hybrid models were statistically significant compared to the ML models. If the ML model and hybrid model have the same accuracy in all experiments, the differences were zero, and the paired t-test could not be performed (denoted as "N/A").

Table 2 presents the results of the paired t-tests.

[1] `https://github.com/sannewielinga/mbai`

Dataset	**Model**	**ML Acc (%)**	**Hybrid Acc (%)**	**ML F1 (%)**	**Hybrid F1 (%)**
Heart					
	KNN	64.26 ± 5.38	64.26 ± 5.38	68.97 ± 4.49	68.97 ± 4.49
	MLP	80.93 ± 3.50	80.19 ± 4.94	81.22 ± 4.51	80.65 ± 5.70
	RF	79.44 ± 4.32	79.44 ± 5.69	81.24 ± 4.30	81.39 ± 5.69
	SVM	**63.52 ± 5.24**	**71.30 ± 8.25**	**70.96 ± 4.82**	**75.14 ± 7.21**
Autism					
	KNN	87.38 ± 1.56	87.38 ± 1.56	91.52 ± 1.11	91.52 ± 1.11
	MLP	97.52 ± 1.35	97.66 ± 1.21	98.30 ± 0.91	98.29 ± 0.82
	RF	**97.38 ± 1.57**	**96.10 ± 1.35**	**98.20 ± 1.11**	**97.33 ± 0.94**
	SVM	**72.62 ± 1.74**	**94.04 ± 1.84**	**84.13 ± 1.17**	**96.01 ± 1.26**
BreastW					
	KNN	94.36 ± 1.83	94.50 ± 2.10	91.62 ± 3.18	91.62 ± 3.18
	MLP	**93.57 ± 2.33**	**94.43 ± 2.20**	**91.04 ± 3.56**	**91.86 ± 3.37**
	RF	**96.36 ± 2.60**	**95.50 ± 2.40**	**94.78 ± 3.69**	**93.63 ± 3.59**
	SVM	95.21 ± 1.85	95.36 ± 1.94	93.11 ± 2.88	93.28 ± 3.15
Ecoli					
	KNN	65.74 ± 4.05	65.74 ± 4.05	57.46 ± 5.24	57.46 ± 5.24
	MLP	**92.50 ± 3.57**	**94.71 ± 2.52**	**91.14 ± 4.51**	**93.93 ± 2.64**
	RF	96.32 ± 2.62	96.47 ± 2.21	95.77 ± 2.97	95.96 ± 2.45
	SVM	**57.06 ± 6.96**	**87.50 ± 15.14**	**23.37 ± 18.45**	**82.99 ± 21.55**
Kidney					
	KNN	89.13 ± 2.13	62.00 ± 5.81	90.36 ± 2.07	76.01 ± 4.29
	MLP	95.25 ± 2.69	95.50 ± 2.30	96.05 ± 2.20	96.28 ± 1.87
	RF	100.00 ± 0.00	99.88 ± 0.40	100.00 ± 0.00	99.89 ± 0.34
	SVM	**91.38 ± 2.60**	**93.13 ± 1.58**	**92.60 ± 2.36**	**94.27 ± 1.46**

Table 1

Average Accuracy and F1 Score with Standard Deviations for ML and Hybrid Models

The results indicate that the improvements in accuracy and F1 score are statistically significant ($p < 0.05$) for datasets where the hybrid model outperformed the ML models. Specifically, significant improvements were observed for:

- **Heart Disease Dataset:** SVM classifier ($p = 0.006$)
- **Autism Screening Dataset:** SVM classifier ($p < 1 \times 10^{-10}$)
- **Breast Cancer Wisconsin Dataset:** MLPClassifier ($p = 0.0013$)
- **Ecoli Dataset:** SVM classifier ($p < 0.0001$) and MLPClassifier ($p = 0.022$)
- **Chronic Kidney Disease Dataset:** SVM classifier ($p = 0.005$)

In datasets where the ML models already performed well, such as the Random Forest model on the Chronic Kidney Disease dataset, the hybrid model maintained similar performance. For KNN and MLP models, the performance was generally consistent with the ML models, with no significant improvements or declines.

Role of ASP

The FOLD-R++ algorithm generated interpretable rules that were important in correcting the errors made by the ML models. The importance of each ASP rule was assessed based

Dataset	**Model**	**t-statistic**	**p-value**	**Significant**
Heart	KNN	N/A	N/A	N/A
Heart	MLP	-0.53	0.606	No
Heart	RF	0.00	1.000	No
Heart	SVM	3.55	0.006	**Yes**
Autism	KNN	N/A	N/A	N/A
Autism	MLP	1.00	0.343	No
Autism	RF	-3.04	0.014	**Yes**
Autism	SVM	32.12	1.35×10^{-10}	**Yes**
BreastW	KNN	N/A	N/A	N/A
BreastW	MLP	3.09	0.013	**Yes**
BreastW	RF	-2.54	0.032	**Yes**
BreastW	SVM	0.80	0.443	No
Ecoli	KNN	N/A	N/A	N/A
Ecoli	MLP	2.75	0.022	**Yes**
Ecoli	RF	0.36	0.726	No
Ecoli	SVM	7.30	4.58×10^{-5}	**Yes**
Kidney	KNN	N/A	N/A	N/A
Kidney	MLP	0.69	0.509	No
Kidney	RF	-1.00	0.343	No
Kidney	SVM	3.74	0.005	**Yes**

Table 2

Results of paired t-tests comparing ML models and hybrid models. N/A indicates the t-test could not be performed due to zero differences in all experiments. Significant results (p < 0.05) are highlighted in bold.

on its frequency of use in correct predictions. The following subsections detail the role of ASP rules in improving the performance for specific datasets and models. The most impactful rules based on the rule ranking from statistically significant programs are presented.

Heart Disease Dataset. For the SVM classifier, the hybrid model improved accuracy from 63.5% to 71.5%. Key ASP rules included:

1. **Rule for labeling "absent"**

   ```
   label(X, absent) :- thal(X, 3),
   maximum_heart_rate_achieved(
   X, V_max_hr_1),
   V_max_hr_1 > 71.0,
   not ab2(X, True), not ab3(X, True),
   not ab4(X, True), not ab5(X, True),
   not ab6(X, True), not ab7(X, True),
   not ab8(X, True).
   ```

   This rule states that if the attribute `thal` equals 3, the maximum heart rate achieved is greater than 71, and none of the exceptions apply, then the diagnosis is "absent" (no heart disease).

2. **Exception rule `ab2`**

   ```
       ab2(X, True) :- chest_pain(X, 4),
                       major_vessels(X, V_major_vessels_1),
   ```

```
V_major_vessels_1 != 0,
not ab1(X, True).
```

This exception accounts for cases where patients exhibit chest pain type 4 and have a certain number of major vessels affect, which indicates a higher risk of heart disease.

For example, `patient34` was initially predicted as "absent" (no heart disease) by the ML model. The hybrid model corrected the prediction to "present", as the value of texttttthal was 3, the maximum heart rate achieved (150) was higher than 71, and the exceptions did not hold.

Autism Screening Dataset. The SVM classifier showed significant improvement when combined with ASP rules, with accuracy increasing from 72.6% to 94.1%. Key ASP rules included:

1. **Rule for labeling "NO"**

   ```
   label(X, NO) :- a5(X, V_a5_0),
   V_a5_0 != 1,
   not ab1(X, True),
   not ab2(X, True).
   ```

   If the individual did not answer "1" to question 5 (`a5`), and exceptions `ab1` and `ab2` do not apply, then they are labeled as "NO" (not autistic). This rule was used 31 times in correcting misclassifications.

2. **Exception rule `ab1`**

   ```
             ab1(X, True) :- a9(X,1),
                            a3(X,1),
                            a1(X,1),
                            a6(X,1).
   ```

   The exception `ab1` is triggered when the individual answers "1" to questions 9, 3, 1, and 6, indicating strong autistic traits that override the general rule.

For example, for specific instance `patient1`, the values satisfied the rule: `a5(X)` was not equal to 1, and exceptions `ab1` and `ab2` did not hold. The ML model predicted the individual as "autistic", but the hybrid model corrected it to "non-autistic".

Breast Cancer Wisconsin Dataset. The neural network model's performance improved with the integration of ASP rules, increasing accuracy from 93.6% to 94.4%. Key ASP rules:

1. **Rule for labeling "Malignant"**

   ```
   label(X, malignant) :- cell_size_uniformity(X,
   V_cell_size_uniformity_0), V_cell_size_uniformity_0 != 1,
   not ab1(X, True), not ab2(X, True),
   not ab3(X, True).
   ```

   A non-uniform cell size suggests malignancy unless exceptions apply. This rule aligns with medical knowledge that irregular cell sizes are indicative of cancerous cells.

   For example, `patient33` was initially misclassified as "benign" by the ML model but was correctly classified as "malignant" by the hybrid model. Here, `cell_size_uniformity` was equal to 5, which satisfied the condition of not being 1. Additionally, exceptions `ab1` and `ab2` did not hold.

2. **Exception rule `ab1`**

```
ab1(X, True) :- bare_nuclei(X, 1),
                cell_size_uniformity(X,
                V_cell_size_uniformity_1),
                V_cell_size_uniformity_1 != 10,
                marginal_adhesion(X,
                V_marginal_adhesion_2),
                V_marginal_adhesion 2 != 10,
                clump_thickness(X,
                V_clump_thickness_3),
                V_clump_thickness_3 != 7,
                cell_size_uniformity(X,
                V_cell_size_uniformity_4),
                V_cell_size_uniformity_4 != 6,
                bland_chromatin(X,
                V_bland_chromatin_5),
                V_bland_chromatin_5 != 5.
```

The exception handles cases where despite non-uniform cell size, other features do not indicate malignancy.

Ecoli Dataset. The hybrid model improved SVM accuracy from 57.1% to 86.8%. Key ASP rules included:

1. **Rule for labeling "cp"**

```
label(X, cp) :- sn(X, V_sn_0), V_sn_0 != FECR,
alm1(X, V_alm1_1),
V_alm1_1 <= 0.38,
not ab1(X, True).
```

This rules states that if the sequence name (`sn`) is not "FECR", and the attribute `alm1` is less than or equal to 0.38, and exception `ab1` does not apply, then the sample is labeled as "cp".

“‘ For example, `patient48` was misclassified as "negative" by the ML model. The hybrid model corrected this to "positive", with the explanation being that the sequence name (`sn`) was not ’FECR’, `alm1` was 0.18 and therefore less than or equal to 0.38, and the exception was invalidated, satisfying the conditions.

2. **Exception rule `ab1`**

```
ab1(X, True) :- sn(X, V_sn_0), V_sn_0 != PTKB,
                gvh(X, V_gvh_1), V_gvh_1 > 0.55,
                mcg(X, V_mcg_2), V_mcg_2 > 0.41.
```

This exception accounts for cases where even if `alm1` is low, certain combinations of other features (`gvh`, `mcg`) indicate that the sample should not be labeled "cp".

“‘

Chronic Kidney Disease (CKD) Dataset. Here, the hybrid model improved SVM accuracy from 91.38% to 93.13%. Key ASP rules:

1. **Rule for labeling "ckd"**

```
label(X, ckd) :- sc(X, V_sc_0), V_sc_0 > 1.2.
```

For `patient70`, the ML model incorrectly predicted "no CKD". The hybrid model correctly identified "CKD" with the explanation that the value of serum creatinine (`sc`) was 1.3, exceeding the threshold of 1.2.

Discussion

The integration of ASP with ML models in this project aimed to improve predictive performance and interpretability. The results demonstrated that the hybrid model significantly improved predictions in instances where standalone ML models showed lower accuracy, particularly for the SVM classifier across multiple medical datasets. The hybrid model consistently outperformed the SVM classifier on datasets such as Heart Disease, Autism Screening, Ecoli, and Chronic Kidney Disease. For example, on the Autism Screening dataset, the accuracy improved from 72.62% to 94.04% after integrating ASP rules, with a statistically significant p-value. Similarly, on the Ecoli dataset, accuracy increased from 57.06% to 87.50%. Improvements were most visible in datasets where the classifier initially performed poorly. The hybrid model was able to correct misclassifications by using the logical rules derived from the FOLD-R++ algorithm. This suggests that the hybrid approach is effective in addressing the limitations of certain ML models when dealing with complex or noisy data.

Beyond improving predictive performance, the hybrid model significantly enhanced interpretability. The logical rules generated by FOLD-R++ are inherently human-readable and can be directly analyzed by domain experts. This transparency allows for better understanding and trust in the model's decisions, which is important in medical applications. For example, in the Breast Cancer Winsconsin dataset, the rule:

```
label(X, NO) :- a5(X, V_a5_0), V_a5_0 != 1, not ab1(X, True),
not ab2(X, True).
```

aligns with medical knowledge that irregular cell sizes are indicative of malignancy.

This project demonstrated that the hybrid model effectively uses the strengths of both approaches, providing accurate predictions along with understandable explanations. The ability to correct errors made by ML models using interpretable rules is valuable in critical domains like healthcare, where trust and transparency are important. By addressing the interpretability gap in black-box models and improving performance in challenging cases, the hybrid approach contributes to the development of more reliable AI systems.

While the hybrid model shows promising results, several limitations should be considered. As datasets become larger and more complex, the number of induced ASP rules can increase substantially. This may potentially affect interpretability. Managing and interpreting a large set of rules may become challenging for users. The variations in ASP programs across different experiments suggest sensitivity to data splitting and randomness. Different training sets may lead to different rules being induced, which affects the consistency of the hybrid model. There may also be multiple hypotheses that fit the data equally well. Furthermore, the hybrid model did not consistently improve performance across all ML models and datasets. For some models, such as KNN and RF on certain datasets, the benefits were minimal or nonexistent. This indicates that the effectiveness of the hybrid approach may depend on the characteristics of both the dataset and the base ML model.

Conclusion

This study addressed two research questions.

First,

The integration of interpretable ASP rules derived from FOLD-R++ significantly improved the predictive performance of some black-box ML models across several medical datasets. The hybrid model improved the accuracy and F1 scores of the ML models, particularly in instances where the ML models initially performed poorly. For example, in the Heart Disease dataset, the accuracy of the SVM classifier improved from 63.5% to 71.5%. In the Autism Screening dataset, the accuracy of the SVM increased from 72.6% to 94.1%. Statistical significance tests confirmed that these improvements were not due to random chance. The logical rules generated by FOLD-R++ provided additional knowledge, correcting errors made by the ML models. This demonstrated that the hybrid approach can improve predictive performance.

Second,

The hybrid model improved the interpretability of predictions without significantly compromising accuracy. The ASP rules derived from FOLD-R++ are inherently interpretable and offer clear insights into the reasoning behind each prediction. For instance, in the Breast Cancer Wisconsin dataset, the hybrid model increased the accuracy of the neural network classifier and provided explanations based on medically relevant features, such as cell size uniformity and clump thickness.

To address the limitations and build upon the findings of this project, future research could explore developing methods to optimize the rule induction process in FOLD-R++. This could reduce the number of rules and improve scalability. Techniques such as rule pruning, clustering, or prioritizing the most impactful rules may improve manageability. Furthermore, implementing the hybrid model in real-world medical settings and evaluating its impact on decision-making processes, clinician acceptance, and patient outcomes would provide valuable insights into its practical utility. Finally, combining the hybrid model with other explainable AI methods, such as feature importance analysis or visualization tools, may further improve interpretability.

References

Arrieta, A. B., Díaz-Rodríguez, N., Del Ser, J., Bennetot, A., Tabik, S., Barbado, A., … others (2020). Explainable artificial intelligence (xai): Concepts, taxonomies, opportunities and challenges toward responsible ai. *Information fusion*, *58*, 82–115.

Doshi-Velez, F., & Kim, B. (2017). Towards a rigorous science of interpretable machine learning. *arXiv preprint arXiv:1702.08608*.

Erdem, E., Gelfond, M., & Leone, N. (2016). Applications of answer set programming. *AI Magazine*, *37*(3), 53–68.

Garcez, A. d., Gori, M., Lamb, L. C., Serafini, L., Spranger, M., & Tran, S. N. (2019). Neural-symbolic computing: An effective methodology for principled integration of machine learning and reasoning. *arXiv preprint arXiv:1905.06088*.

Gebser, M., Kaminski, R., Kaufmann, B., & Schaub, T. (2019). Multi-shot asp solving with clingo. *Theory and Practice of Logic Programming*, *19*(1), 27–82.

Gilpin, L. H., Bau, D., Yuan, B. Z., Bajwa, A., Specter, M., & Kagal, L. (2018). Explaining explanations: An overview of interpretability of machine learning. In *2018 ieee 5th international conference on data science and advanced analytics (dsaa)* (pp. 80–89).

Guidotti, R., Monreale, A., Ruggieri, S., Turini, F., Giannotti, F., & Pedreschi, D. (2018). A survey of methods for explaining black box models. *ACM computing surveys (CSUR)*, *51*(5), 1–42.

Kelly, M., Longjohn, R., & Nottingham, K. (n.d.). *The uci machine learning repository.* https://archive.ics.uci.edu.

Lifschitz, V. (2019). *Answer set programming* (Vol. 3). Springer Heidelberg.

Lundberg, S. (2017). A unified approach to interpreting model predictions. *arXiv preprint arXiv:1705.07874*.

Manhaeve, R., Dumancic, S., Kimmig, A., Demeester, T., & De Raedt, L. (2018). Deepproblog: Neural probabilistic logic programming. *Advances in neural information processing systems*, *31*.

Palaniappan, K., Lin, E. Y. T., & Vogel, S. (2024). Global regulatory frameworks for the use of artificial intelligence (ai) in the healthcare services sector. In *Healthcare* (Vol. 12, p. 562).

Ribeiro, M. T., Singh, S., & Guestrin, C. (2016). " 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* (pp. 1135–1144).

Rudin, C. (2019). Stop explaining black box machine learning models for high stakes decisions and use interpretable models instead. *Nature machine intelligence*, *1*(5), 206–215.

Spiotta, M., Terenziani, P., & Dupré, D. T. (2017). Temporal conformance analysis and explanation of clinical guidelines execution: An answer set programming approach. *IEEE Transactions on Knowledge and Data Engineering*, *29*(11), 2567–2580.

Tonekaboni, S., Joshi, S., McCradden, M. D., & Goldenberg, A. (2019). What clinicians want: contextualizing explainable machine learning for clinical end use. In *Machine learning for healthcare conference* (pp. 359–380).

Wang, H., & Gupta, G. (2022). Fold-r++: a scalable toolset for automated inductive learning of default theories from mixed data. In *International symposium on functional and logic programming* (pp. 224–242).

Xu, F., Uszkoreit, H., Du, Y., Fan, W., Zhao, D., & Zhu, J. (2019). Explainable ai: A brief survey on history, research areas, approaches and challenges. In *Natural language processing and chinese computing: 8th ccf international conference, nlpcc 2019, dunhuang, china, october 9–14, 2019, proceedings, part ii 8* (pp. 563–574).

Yang, Z., Ishay, A., & Lee, J. (2023). Neurasp: Embracing neural networks into answer set programming. *arXiv preprint arXiv:2307.07700*.

Appendix

Description Heart disease dataset

Table 3

Overview heart disease dataset

Data	Explanation
age	age in years
sex	male or female
cp	chest pain (0 or 1)
trestbps	resting blood pressure
chol	serum cholesterol in mg/dl
fbs	fasting blood sugar > 120 mg/dl
restecg	resting electrocardiographic results (values 0,1,2)
thalach	maximum heart rate achieved
exang	exercise induced angina
oldpeak	oldpeak = ST depression induced by exercise relative to rest
slope	the slope of the peak exercise ST segment
ca	number of major vessels (0-3) colored by fluoroscopy
thal	thal: 0 = normal; 1 = fixed defect; 2 = reversible defect
target	0=normal; 1= heart disease