Integrating Symbolic Knowledge and Machine Learning in Healthcare

Christel Sirocchi[1,*,†], Sara Montagna[1,*,†]

[1]Department of Pure and Applied Sciences, University of Urbino, Piazza della Repubblica 13, 61029, Urbino, Italy

Abstract

The intersection of Artificial Intelligence and healthcare has driven advancements, particularly through machine learning, which exploits large datasets to develop predictive models and identify risk factors. Despite its success in clinical medicine, only a few models are FDA-approved due to issues of trustworthiness and lack of explainability, hindering adoption in clinical settings. Addressing these issues, symbolic knowledge injection and symbolic knowledge extraction have emerged. The first approach integrates domain-specific expertise encoded as rules into machine learning models, while the second extracts interpretable rules from trained models.

In this study, this framework is validated using the Pima Indians diabetes dataset, a benchmark in diabetes research. By incorporating a diagnostic protocol for diabetes into machine learning models, the study demonstrates an improvement in the predictive capabilities of these models. By extracting rules from pure data-driven trained models and integrating them with medical knowledge, we reduce false negatives, while achieving a fully explainable diagnostic system. Finally, a combination of these two methods is explored, reporting higher diabetes detection rates and improved model explainability. Accordingly, this study demonstrates the potential of combining machine-learnt insights with medical guidelines to improve healthcare outcomes.

Keywords

Hybrid ML architecture, Symbolic knowledge extraction, Symbolic knowledge injection

1. Introduction

In medical settings, critical decisions often rely on clinical protocols that, while generally reliable and trustworthy, sometimes fail to correctly identify a subtle yet significant subset of patients. These patients fall within the ”grey zone”, characterised by uncertainty about the appropriate course of action, as they are not clearly defined as either normal or abnormal, healthy or diseased [1]. In these cases, decisions may be more subjective or open to interpretation, challenging the accuracy of conventional protocols. In response, the literature recognises the advanced capabilities of Machine Learning (ML) models, which can uncover latent patterns and knowledge from data that extend beyond the scope of traditional medical protocols [2].

Despite advancements, significant issues persist. The accuracy of certain ML algorithms is not consistently satisfactory, and discrepancies are often observed between predictions made

RuleML+RR'24: Companion Proceedings of the 8th International Joint Conference on Rules and Reasoning, September 16–22, 2024, Bucharest, Romania

*Corresponding author.

†These authors contributed equally.

c.sirocchi2@campus.uniurb.it (C. Sirocchi); sara.montagna@uniurb.it (S. Montagna)

0000-0002-5011-3068 (C. Sirocchi); 0000-0001-5390-4319 (S. Montagna)

by these models and those derived from clinical protocols. Moreover, in most cases, they are characterised by a level of opacity that makes it hard for humans to understand their behaviour. However, both interpreting and explaining model predictions is crucial in the medical domain, which is a safety- and ethic-critical application. Given these premises, there is a growing recognition of the need for hybrid models that integrate the robustness of medical protocols with the adaptive learning capabilities of ML. This integration aims to harness the strengths of both approaches while ensuring the decisions are both explainable and reliable.

Our goal in this paper is to engineer new Artificial Intelligence (AI) solutions that address these challenges. We aim to integrate medical knowledge and ML solutions, building upon existing literature that introduces the concepts of Symbolic Knowledge Injection (SKI) and Symbolic Knowledge Extraction (SKE) [3]. Our objectives are twofold: first, to demonstrate the advantages of SKI-SKE technologies in terms of various indicators within the medical domain, showcasing how performance improves; and second, to experiment with these technologies which are often only introduced in literature and only partially validated, especially within the medical context. This paper demonstrates how performance and explainability evolve, starting from simple knowledge bases (KB) and progressing to pure ML algorithms. Building on the two models with the highest recall (decision trees and neural networks), we applied SKI and SKE technologies and conducted novel experimentation with a SKI-SKE loop. In this loop, recently proposed in the literature and open to exploration, medical knowledge is injected into an ML model, rules are extracted from the trained model, and then re-injected into the model.

The potential of this integrated approach is demonstrated using the Pima Indians Diabetes dataset for diabetes prediction [4]. Results show that applying SKI techniques to inject clinical knowledge into ML models improves performance, specifically reducing the number of false negatives in diabetes diagnosis. Additionally, SKE techniques can derive interpretable models that are further enhanced when combined with clinical knowledge. Integrating both techniques into a loop yields novel and promising results, where knowledge extracted from neural networks and re-injected can further enhance model performance and explainability.

2. Motivations and Background

The intersection of artificial intelligence and healthcare has fostered significant advancements. ML, in particular, is the most discussed technology in this field [5, 2], as it allows for the exploitation of large datasets by discovering relationships and patterns hidden in data. Beyond developing accurate and robust clinical predictive models, ML is also extensively used to identify risk factors by detecting key features in predictions. ML has achieved remarkable performance in various domains of clinical medicine, outperforming human physicians in some cases and enabling the development of computer-aided diagnosis systems [6]. However, with thousands of studies applying ML to medical data, only a handful have significantly contributed to clinical care: indeed, only a few of these systems have been FDA-approved for healthcare use [7].

Resistance to embrace ML in clinical settings can be attributed to the prevailing reliance on evidence-based clinical guidelines as the foundation for clinical decision-making [8], while classical ML does not rely on medical knowledge but solely on data. Novel ML models, even when reporting superior performance compared to current protocols, might be unsuitable

for clinical use if they *(a)* fail to correctly predict cases effectively managed by the protocol in place due to potential liabilities, *(b)* make predictions based on confounding variables and erroneous relationships that contradict established clinical knowledge [9] or *(c)* make predictions that cannot be explained to the user, suffering from opacity and offering poorly interpretable solutions [10]. On the other side, medical protocols alone can sometimes fail to detect complex patterns, correlations, causal relationships and little variations in data due to their reliance on predefined rules and thresholds, making them less effective in borderline decision cases [11].

Since healthcare is a safety and ethic-critical application requiring humans to be in full control of the computational system supporting their decisions, the goal is to find methods that ensure the best trade-off between performance and explainability. To bridge this gap, the integration of medical knowledge with ML has emerged as a topic of ongoing debate in the literature.

2.1. Symbolic Knowledge Injection and Extraction

In the context of knowledge exploitation, with the purpose of both creating more reliable recommenders and understanding the decision process, two main methods have been defined in literature [3]. Symbolic knowledge and methods involve the use of interpretable languages, such as logic formalisms, that are understandable by both humans and computers. In contrast, subsymbolic knowledge involves the use of numerical data processing, such as functions over fixed-sized tensors in NNs, which often results in less interpretable solutions despite their high predictive performance. Additionally, the literature introduces the concepts of Symbolic knowledge injection and extraction:

Symbolic knowledge injection – SKI Particular attention is given to methods performing knowledge injections into ML models, which fall under the paradigm of informed ML [12, 13]. This approach, also referred to as symbolic knowledge injection, aims to enhance ML models by integrating data-driven learning with domain-specific expertise typically encoded as rules. It encompasses a class of algorithms that ensure sub-symbolic predictors draw their inferences *consistently* with a given set of symbolic knowledge. SKI procedures of this kind influence either the structure or the training process of sub-symbolic predictors, ensuring that these predictors incorporate symbolic knowledge when making predictions. Consequently, these procedures compel sub-symbolic predictors to learn from both data and symbolic knowledge. SKI can thus result in a higher control over what the ML model is learning, ensuring more reliable and trustworthy predictors whose behaviour is consistent with domain knowledge.

Symbolic knowledge extraction – SKE Symbolic knowledge extraction methods are also documented in the literature as a means to derive symbolic knowledge from trained ML models, which can then be used in decision support systems [14]. The goal of SKE is manifold. First, given a black-box predictor and a knowledge-extraction procedure, the extracted knowledge can be used as a basis to construct explanations for that predictor. The extracted knowledge may serve as an interpretable replacement, also referred to as *surrogate model* for the original predictor, provided that the two have a high-fidelity score. Moreover, this approach facilitates the discussion of *how* and *if* the extracted knowledge can be merged with existing domain knowledge to improve the classifications

based solely on domain knowledge. Finally, open research questions arise, discussing whether the surrogate model can truly enrich the domain knowledge or if it presents any contradictions and, in this case, how to reconcile the two. The same considerations apply if we aim to integrate surrogate models extracted from different predictors.

SKI and SKE are thus methods devised to integrate knowledge *into* and *from* predictors. Several approaches have been developed which, according to [3], may be categorised as follows. SKI methods are classified by input knowledge form, strategy, targeted predictor type, and purpose. They accept logic formulæ or expert knowledge, including First Order Logic and Knowledge Graphs (KGs). SKI strategies include predictor structuring, knowledge embedding, and guided learning. They primarily target NN-based predictors. Conversely, SKE methods are mainly classified by translucency, *i.e.*, if they rely on the inspection of the internal structure of black-box models and output knowledge (rule lists, graphs, decision trees, tables). The method can inspect (even partially) the internal parameters of the underlying black-box predictor, such as with neural networks. The symbolic knowledge produced can be in the form of propositional and fuzzy rules, decision trees or triplets of KGs. The potential of the *joint* exploitation of both SKI and SKE is also recognised in the literature, specifically in the loop presented in [3] as *train–extract–fix–inject*. In this loop, a trained model is inspected via SKE, the extracted knowledge is verified by a domain expert, and the corrected knowledge is injected back into the trained predictor via SKI to align with the corrected symbolic knowledge. This approach is proposed for debugging purposes but has not yet been thoroughly investigated and experimented with for improving classifier performance and explainability.

2.2. Knowledge Integration in Medicine

Literature reports different integration strategies, mainly devoted to injecting knowledge in the various stages of the ML pipeline [12, 15, 16]. A comprehensive review is out of the scope of this paper, but we report here the main methods:

Data Pre-processing Inconsistencies and errors in datasets are mitigated by removing anomalous samples based on clinical norms. To counter insufficient or missing clinical data, virtual samples adhering to medical knowledge can be generated [17].

Feature Engineering Novel features can be derived from existing ones using mathematical or logical models based on medical knowledge [18]. Feature selection can be strategically informed by prior knowledge [19].

Model Learning Rules can be incorporated into model loss function and architecture [20, 21].

Output Evaluation ML models can be combined with rule-based systems modelling clinical guidelines, either by integrating outputs, filtering predictions in series, or verifying consistency with domain knowledge [22].

However, these attempts are sparse and do not refer to the SKI-SKE framework, where also the extraction of knowledge plays a crucial role, thereby losing part of the expected benefits, especially in terms of model explainability. Only recently some work introduced a discussion on SKE, but still only in one direction and within the specific domain of diagnostic imaging [23].

3. Materials and Methods

Given the identified gaps in the literature, in this paper, we explore some of the SKI-SKE methods presented above, with the goal of defining a framework that effectively leverages the advantages of data analytics and the exploitation of well-grounded medical rules. Special attention is devoted to experimenting with the loop that exploits both SKI and SKE methods to assess the validity of this approach and evaluate improvements in model performance and explainability. To the best of our knowledge, no attempts in this direction are discussed in the literature.

3.1. Dataset and domain knowledge

The dataset analysed in this study is the Pima Indians Diabetes dataset, compiled by the National Institute of Diabetes and Digestive and Kidney Diseases. This dataset originates from a study of the Pima Indian population, known for its high incidence of diabetes. It includes 768 medical profiles of women aged 21 and older who underwent an Oral Glucose Tolerance Test (OGTT) to measure their glucose and insulin levels after two hours. The target variable is binary, indicating whether diabetes was diagnosed within five years, and is unbalanced, with diabetes diagnoses accounting for 35% of the cases. Details about the dataset features are listed in Table 1. Missing values in the attributes I_{120} (48.70%), ST (29.56%), BP (4.55%), BMI (1.43%), and G_{120} (0.65%) were imputed using the median value.

Table 1
Pima Indians Diabetes dataset

Feature name	Code	Description
Pregnancies		Number of times pregnant
Glucose	G_{120}	2-hour plasma glucose concentration in OOGT in mg/dL
Blood Pressure	BP	Diastolic blood pressure in $mmHg$
Skin Thickness	ST	Triceps skin-fold thickness in mm
Insulin	I_{120}	2-hour serum insulin in $\mu U/mL$
Body mass index	BMI	Body mass index as weight/(height)^2 in kg/m^2
Diabetes Pedigree Function	DPF	Likelihood function of diabetes based on family history [4]
Age		Age in years

Public health guidelines on type-2 diabetes risks indicate that individuals with a high BMI ($\geq$ 30) and elevated blood glucose levels ($\geq$ 126) are at a severe risk for diabetes. Conversely, those with a normal BMI ($\leq$ 25) and low blood glucose levels ($\leq$ 100) are less likely to develop the disease. These guidelines have been used to design rules [24] expressed as logic predicates (Table 2), which form the KB for this case study.

Table 2
Knowledge base for predicting risk of type-2 diabetes as formalised by Kunapuli et al. (2010) [24].

Rule 1	$(BMI \geq 30) \wedge (G_{120} \geq 126)$	$\Longrightarrow$	diabetes
Rule 2	$(BMI \leq 25) \wedge (G_{120} \leq 100)$	$\Longrightarrow$	healthy

3.2. Machine learning models and metrics

In this study, a wide range of ML classifiers are explored, including linear models such as Logistic Regression (LR) and linear Support Vector (SV) classifiers, tree-based approaches including single learners like Decision Trees (DT) and ensemble methods such as Gradient Boosting (GB) and Random Forest (RF), as well as Neural Networks (NN). The data was normalised to a mean of 0 and a standard deviation of 1 for facilitating the learning of scale-sensitive models, such as NN. Performance evaluation encompassed Accuracy (A), Precision (P), F1 score (F1), Balanced Accuracy (BA), and Matthew's Correlation Coefficient (MCC), as well as True Positive Rate (TPR) or recall, True Negative Rate (TNR), False Positive Rate (FPR), and False Negative Rate (FNR). Nested cross-validation with 10 outer folds for evaluation and 5 inner folds for hyperparameter tuning was employed with an extensive parameter search. Hyperparameter optimisation was conducted by maximising accuracy with class weights set inversely proportional to class frequency to address data imbalance. Alternative strategies, such as random oversampling of the positive class and undersampling of the negative class, were also tested but did not improve performance.

For NN, the optimal number of training epochs was determined by early stopping. This method involved splitting the training set into 90% training and 10% validation subsets and monitoring the validation loss during training, for a maximum of 100 epochs. Early stopping was configured with a patience of 5 epochs, meaning training would halt if the validation loss did not improve for 5 consecutive epochs, and the best weights observed during training were restored. Performance metrics were computed for each outer fold using the model parameters optimised in the inner folds, and the average of these metrics was calculated to provide a comprehensive understanding of the models performance.

In the remainder of this paper, we focus on NN and DT along with their respective learning methods. These two families of predictors are particularly relevant as they are closely related to many surveyed SKI and SKE methods. DTs are noteworthy for their user-friendliness, making them accessible and interpretable for users. In contrast, NNs are predominantly popular due to their superior predictive performance and flexibility, allowing them to adapt to a wide range of tasks and data types. Moreover, considering the clinical context where correctly identifying positive cases is critical and recall is the key metric to minimise the risk of missing critical diagnoses, NN and DT are identified as the best-performing models according to results presented in Table 3 and are considered for further exploration.

In particular, the reference NN architecture, derived through hyperparameter optimisation, was configured as follows: an input layer of size 8; two hidden layers of size 12 and 8 with Rectified Linear Unit (ReLU) activation function; an output layer comprising a single neuron with a sigmoid activation function. The model was compiled using the Adam optimiser and binary cross-entropy with class weights as the loss function, with performance evaluation based on weighted accuracy. Models were trained with a batch size of 32 for a number of epochs determined by early stopping with patience 5 and a maximum of 100 epochs, as described. The reference DT architecture was configured with a maximum depth of 10 and Gini impurity as the split criterion. DTs were trained with class weights to account for data imbalance.

3.3. Knowledge injection and extraction: the PSyKE and PsyKI Platforms

Knowledge injection and extraction in NNs leveraged two Python libraries [1]: PSyKI (Platform for Symbolic Knowledge Injection) [25] and PSyKE (Platform for Symbolic Knowledge Extraction) [26]. Knowledge injection is facilitated by methods available in PSyKI [25]. This Python library primarily uses logic formulae for knowledge representation, supported by the Prolog language through integration with 2P-Kt [2], a multi-paradigm logic programming framework. Key components of PSyKI include Injectors, Theories, and Fuzzifiers, which represent SKI algorithms, domain-specific symbolic knowledge, and methods for translating symbolic knowledge into sub-symbolic data structures, respectively. The available injectors include Knowledge-Based Artificial Neural Networks (KBANN) [27], one of the first injectors introduced in the literature, Knowledge Injection via Lambda Layer (KILL) [28] and Knowledge Injection via Network Structuring (KINS) [29], which structures knowledge by adding ad-hoc layers into a NN. In this work, knowledge injection in NNs, depicted in Figure 1 (a), was performed using **KINS** due to its several advantages: it does not constrain the NN to a specific architecture, does not require logic predicates to be grounded, and is robust to both data scarcity and imperfect or incomplete knowledge, often found in clinical scenarios. In the KINS method, a neural network (NN) is first initialised with a specified architecture. The architecture is then augmented with additional neural modules specifically designed to incorporate symbolic knowledge. Each module functions as a sub-network, sharing the input layer with the original NN and producing an output that represents the continuous interpretation of a logic formula. The weights and biases within these modules can be either trainable or fixed, while the rest of the network's weights and biases remain trainable. In this study, the knowledge module weights are not trained to ensure that all provided logic rules are given equal importance, regardless of data evidence.

Knowledge extraction methods are available in PSyKE, which offers several algorithms for both classification and regression problems, allowing knowledge to be extracted in the form of a Prolog theory. PSyKE is designed around the notion of an Extractor, which is composed of a trained predictor, used as an oracle and a set of feature descriptors. The supported extraction algorithms include those based on trees, iteratively dividing the feature space, like Classification and Regression Trees (CART) and Trepan, as well as those based on hypercubes, iteratively expanding in the input space, like ITER, GridEx, and GridREx [26]. In this study, knowledge extraction from NNs, illustrated in Figure 1 (b), was performed using **CART** due to its simplicity and interpretability. CART performs rule extraction by training a decision tree on the inputs and outputs of the NN and converting the tree structure into human-readable if-then rules. The fidelity of the obtained rule set was evaluated in terms of accuracy and F1-score with respect to the black-box model. The optimal number of leaves, and thus rules, in the CART rule-extraction process was determined by varying the leaf number from 5 to 20 and selecting the value that maximised the accuracy of the rule set on a validation set.

Knowledge injection and extraction in DTs was relatively straightforward as both DTs and domain knowledge can be formalised as rules. Knowledge injection by model restructuring was achieved by modifying the structure of the DT to incorporate the two domain-specific rules as its initial split criteria. Beyond these rules, the tree expanded as a typical DT. For knowledge

[1] https://github.com/psykei
[2] http://tuprolog.apice.unibo.it

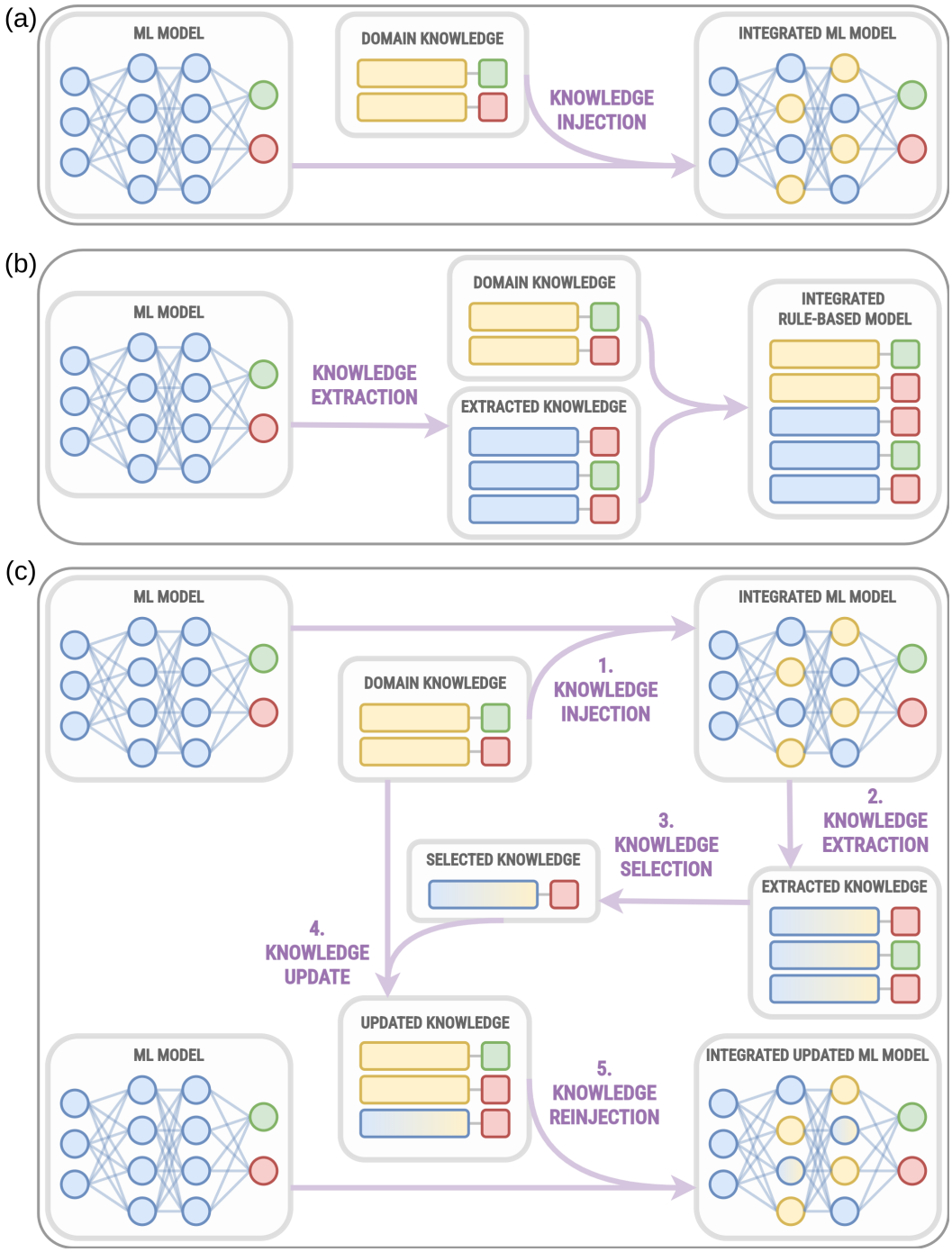

Figure 1: Diagrams illustrating the three integrated approaches leveraging SKI-SKE technologies that were implemented and evaluated in this study: (a) knowledge injection, (b) knowledge extraction, (c) injection-extraction-injection loop.

extraction, the DT was simply converted into a rule set by translating root-to-leaf paths into if-then rules and adding the two domain-specific rules with priority such that, if an instance satisfies the conditions of multiple rules, priority is given to the clinical rules.

The effectiveness of knowledge injection in enhancing predictive model performance was evaluated by training the reference NN and DT architectures, along with their injected counterparts by 10-fold cross-validation. Performance metrics were averaged across folds and compared to assess improvements resulting from knowledge injection. Similarly, the same reference NN and DT architectures were trained using 10-fold cross-validation, and for each fold, converted into interpretable rule sets. The predictive performance of these extracted rule sets was averaged across all folds and compared to that of the original clinical protocol. Additionally, integrated rule sets, which combined clinical rules with ML-derived rules, were evaluated to detect any increase in predictive performance as a result of this integration.

3.4. Knowledge injection-extraction feedback loop

The potential to apply a combination of SKI and SKE strategies in a feedback loop to further enhance the predictor's performance was explored. The process, outlined in Figure 1 (c), begins with the initial injection of available domain knowledge. The model is then trained, and rules are extracted from it. The quality of these rules is evaluated, and the best rules are added to the current domain rules, which are then re-injected into a new model.

In this study, the injection-extraction process was structured as follows. The dataset was divided into training, validation, and test sets in a 60:20:20 ratio. A NN injected with the two protocol rules was trained on the training set, with training parameters optimised based on performance on the validation set. Rules were then extracted from the trained injected NN, with the rule set size fine-tuned according to validation set performance. These extracted rules were evaluated using performance metrics as well as coverage, which measures the proportion of dataset samples accounted for by the rule set. Four rules predicting diabetic outcomes were identified and added to the clinical protocol, both individually and in combination, and reinjected into new NN models. Consequently, five new NN models were injected with the updated knowledge bases. Their performance was compared against the initial injected NN model and the traditional NN model to assess the impact of injecting ML-derived rules.

4. Results and discussion

4.1. ML performance

The initial performance comparison of various ML models trained on the Pima Indians diabetes dataset is summarised in Table 3. All models show moderate prediction accuracy, ranging from 0.73 to 0.78. Among these, RF stands out with the highest quality of positive predictions, evidenced by superior precision, and the highest scores for overall performance metrics, such as A, BA, F1, and MCC. SV excels in predicting the negative class (healthy individuals), with the lowest FPR and highest TNR. In contrast, NN demonstrates the best capability for predicting the positive class (diabetic individuals), achieving the highest TPR and lowest FNR. DT and RF follow closely and are notable for their diabetes prediction capabilities.

Table 3
Evaluation metrics for ML models trained on the Pima Indians diabetes dataset. The best value for each metric is highlighted in bold, corresponding to the highest value for all metrics, except for FPR and FNR for which it is the lowest.

Metric	A	BA	F1	MCC	P	TNR	TPR	FNR	FPR
Neural Network	0.738	0.742	0.670	0.472	0.612	0.730	**0.754**	**0.246**	0.270
Decision Tree	0.738	0.741	0.667	0.468	0.604	0.730	0.753	0.247	0.270
Random Forest	**0.772**	**0.768**	**0.697**	**0.522**	**0.652**	0.782	0.753	0.247	0.218
Gradient Boosting	0.756	0.754	0.681	0.499	0.637	0.762	0.746	0.254	0.238
Support Vector	0.762	0.751	0.678	0.497	0.651	**0.786**	0.717	0.283	**0.214**
Logistic Regression	0.751	0.742	0.666	0.477	0.636	0.774	0.709	0.291	0.226

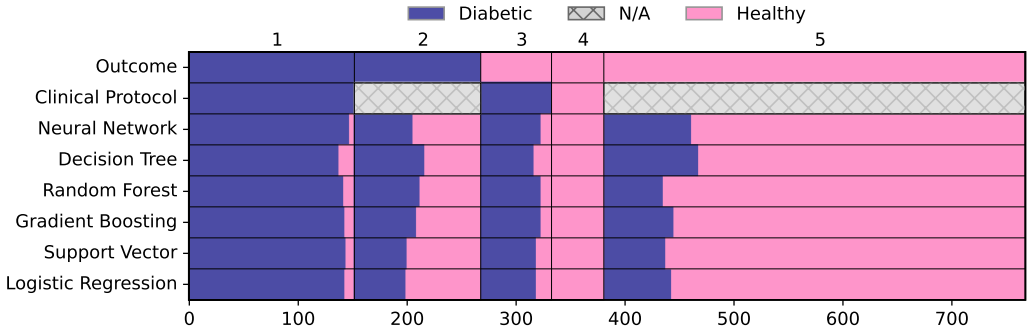

Figure 2: Diabetes dataset divided into five regions based on the predictions of the clinical protocol with respect to the actual outcomes. The proportion of diabetic and healthy predictions made by six ML models is shown for each region.

A detailed analysis of the predictions made by each model, compared to those made by the clinical protocol and the actual outcomes, is illustrated in Figure 2. The graph is divided into regions based on whether the clinical protocol correctly predicts positive and negative instances. For each region, the proportion of healthy and diabetic predictions made by ML models is displayed. It can be observed that the coverage of the clinical protocol is relatively low, at about 34.5%, leaving many cases, primarily healthy individuals, without a diagnosis. Such cases are generally deferred to follow-up, thus treated for the time being as healthy individuals. For this reason, in performance metrics computation, these cases are considered healthy. Additionally, it can be noted that the protocol produces false positives (region 3) but no false negatives, which is highly desirable in a clinical setting where a positive outcome typically leads to specialised tests for confirmation, whereas a negative outcome usually does not prompt further examination.

Examining the predictions of the ML models in detail reveals several insights. In **region 1**, which includes diabetic cases correctly predicted by Rule 1 of the protocol, all models make some mistakes, with NN reporting the fewest errors in this region and DT the most. In **region 2**, which includes diabetic cases where the protocol could not make predictions, the most crucial classification challenge arises, as these patients inhabit a clinical ”grey zone” and often do not

receive adequate care. All ML models struggle to classify this region. DT emerges as the best-performing model and the only one correctly identifying over 50% of the patients as diabetic. Poor performance indicates that the available features may not be sufficiently predictive for these cases. However, some patients are correctly identified by multiple models, indicating potential criteria for accurate classification. In **region 3**, which includes cases incorrectly classified as diabetic by Rule 1 of the protocol, most models also classify these instances as diabetic, suggesting that the available features are not sufficiently predictive also for these patients. This misclassification needs to be addressed as it increases over-triage for healthcare providers but takes lower priority, as our primary focus is on reducing false negatives rather than false positives. In **region 4**, which includes healthy individuals correctly predicted by Rule 2 of the protocol, all models also predict these patients as healthy. In **region 5**, which consists of healthy individuals for whom the protocol cannot give a prediction, all models correctly predict most patients. The fraction of false positives remains below 20% for all models, demonstrating the value of ML in predicting these patients. These findings underscore the opportunities (region 5) and challenges (regions 2 and 3) in leveraging ML for clinical prediction. Combining data-driven ML with rule-based knowledge may address these challenges, forming the basis for investigating knowledge injection and extraction to enhance predictive models.

4.2. Knowledge injection and extraction

Performance evaluation of DT and NN architectures injected with clinical rules by model restructuring is presented in Table 4. Injected models were evaluated against the standard ML architectures and the clinical protocol. Despite the vastly different learning paradigms, the effect of knowledge injection on the two models was similar. The injection led to an increase in the classification of positive outcomes, with a rise in both true positives and false positives. This is due to the fact that, as discussed in the previous section, the clinical protocol does not predict false negatives but does predict false positives through Rule 1. This increase in positive predictions yields an increase in TPR and a decrease in FNR for both injected models, a desirable outcome in clinical scenarios where the primary objective is identifying positive cases. However, this comes at the expense of P, particularly in the DT model, where the overall performance metrics—including A, BA, F1, and MCC—degraded. In contrast, the NN model showed an improvement in these metrics, indicating a more balanced trade-off between precision and recall. These results highlight the potential of augmenting ML models with available knowledge.

However, in clinical settings, black-box models like NNs, and even rule-based methods like decision trees DTs when the elevated number of rules impacts model interpretability, are often not adopted due to their lack of transparency and trustworthiness. Therefore, working with a small set of interpretable rules that closely approximate the behaviour of trained ML models could be more useful and applicable in clinical practice. In this regard, effective knowledge integration can be achieved by combining protocol rules with rules derived from ML models through knowledge extraction methods. The performance evaluation of rule sets extracted from trained ML models and composite rule sets combining extracted rules with protocol rules, is presented in Table 5. As with the injected models, integration results in an increase in positive predictions, as indicated by higher TPR and lower FNR. In this case, however, also P either remains stable or improves. All global performance metrics—A, BA, F1, and MCC—also show

improvement. These findings suggest that when using a surrogate interpretable model in place of a black-box model, integrating additional rules from clinical knowledge can enhance predictions, especially in areas where the protocol is effective.

Table 4
Knowledge injection. Evaluation metrics computed for the clinical protocol formalising the KB and ML models trained on the Pima Indians dataset. ML models comprise DT and NN, both trained solely on data, or injected with domain knowledge by model restructuring, denoted as NN-I and DT-I, respectively.

Metric	A	BA	F1	MCC	P	TNR	TPR	FNR	FPR
KB	0.764	0.719	0.626	0.466	0.707	0.869	0.567	0.433	0.131
NN	0.752	0.751	0.679	0.493	0.634	0.756	0.746	0.255	0.244
NN-I	0.759	0.765	0.694	0.513	0.628	0.747	**0.783**	**0.218**	0.253
DT	0.721	0.719	0.638	0.424	0.582	0.725	0.711	0.287	0.275
DT-I	0.676	0.686	0.607	0.360	0.527	0.650	**0.723**	**0.276**	0.350

Table 5
Knowledge extraction. Evaluation metrics computed for rule sets over the Pima Indians diabetes dataset, including the clinical protocol formalising the Knowledge Base (KB), the Decision Tree model trained on data (DT), the rule set extracted from the Neural Network using CART (NN-E), as well as composite rule sets DT+KB and NN-E+KB, integrating protocol rules with priority.

Metric	A	BA	F1	MCC	P	TNR	TPR	FNR	FPR
KB	0.764	0.719	0.626	0.466	0.707	0.869	0.567	0.433	0.131
NN-E	0.722	0.723	0.643	0.435	0.588	0.722	0.724	0.275	0.278
NN-E+KB	0.725	0.731	0.654	0.449	0.589	0.711	**0.750**	**0.249**	0.289
DT	0.721	0.719	0.638	0.424	0.582	0.725	0.711	0.287	0.275
DT+KB	0.726	0.736	0.661	0.454	0.583	0.704	**0.768**	**0.232**	0.296

4.3. Knowledge injection-extraction feedback loop

The explorations in the previous section highlight the potential of using injection and extraction techniques to incorporate symbolic knowledge into the learning process or derive symbolic knowledge from trained models. However, the combined application of these approaches is heavily understudied, and a preliminary investigation is presented here. The combination of these strategies was set up as an injection-extraction-injection loop, capitalising on the enhanced performance through knowledge injection and improved explainability from knowledge extraction (due to the intrinsic interpretability of rule-based systems). A model injected with clinical rules was trained on data, and a rule set maximising fidelity with the model was extracted. The extracted rules, along with performance metrics, are presented in Table 6. Four rules predict diabetic outcomes and are further analysed. Extracted **Rule 1** closely mirrors Rule 1 of the clinical protocol, predicting diabetic individuals with elevated glucose and BMI. The thresholds for these features are lower in the extracted rules, suggesting that individuals with glucose and BMI just below the clinical thresholds should also be considered at elevated risk.

Extracted **Rule 2** suggests that individuals with elevated glucose could be considered at higher risk above a certain age, even if they do not have elevated BMI, identifying age as an additional risk factor not considered in the protocol. Conversely, extracted **Rule 3** indicates that even if the glucose level is not elevated, risk could still be high if BMI is very elevated, prompting to consider these two features not only in combination but also individually. Finally, extracted **Rule 4** suggests that even if glucose and BMI are not elevated, risk might still be high above a certain age and with a family history of diabetes quantified by DPF, prompting to consider these two additional factors even when the two main diabetes risk factors are in the normal range. The extracted rules can be used to augment rule-based protocols or to improve ML training.

Table 6
Performance metrics computed on the test set for 6 rules extracted from a NN model injected with the rules of the diabetes clinical protocol and trained on the Pima Indians diabetes dataset.

	Rule	Outcome	Total	#TP	#TN	#FP	#FN	A	Coverage
1	$G120 > 121.5 \wedge BMI > 29.1$	diabetes	262	168	0	94	0	0.641	0.341
2	$G120 > 121.5 \wedge BMI =< 29.1 \wedge Age > 30.5$	diabetes	44	18	0	26	0	0.409	0.057
3	$G120 =< 121.5 \wedge BMI > 40.75$	diabetes	33	12	0	21	0	0.364	0.043
4	$G120 =< 121.5 \wedge BMI =< 40.75 \wedge DPF > 0.65 \wedge Age > 40$	diabetes	75	23	0	52	0	0.307	0.098
5	$G120 =< 121.5 \wedge BMI =< 40.75 \wedge DPF =< 0.65$	healthy	317	0	277	0	40	0.874	0.413
6	$G120 > 121.5 \wedge BMI =< 29.1 \wedge Age =< 30.5$	healthy	37	0	30	0	7	0.811	0.048

Table 7
Evaluation metrics computed for NNs injected with prior domain knowledge together with rules extracted from trained NNs.

Metric	A	BA	F1	MCC	P	TNR	TPR	FNR	FPR
KB	0.764	0.719	0.627	0.463	0.700	0.869	0.567	0.433	0.131
NN	**0.771**	0.768	0.698	0.519	**0.643**	**0.773**	0.765	0.235	**0.227**
NN-I	0.741	0.752	0.680	0.481	0.598	0.716	0.787	0.212	0.284
NN-I update #1	0.760	0.769	0.699	0.516	0.622	0.740	0.799	0.201	0.260
NN-I update #2	0.768	**0.772**	**0.702**	**0.523**	0.636	0.760	0.784	0.218	0.240
NN-I update #3	0.754	0.770	0.700	0.516	0.609	0.716	**0.825**	**0.175**	0.284
NN-I update #4	0.758	0.761	0.690	0.503	0.623	0.750	0.772	0.226	0.250
NN-I update #5	**0.771**	0.770	0.701	**0.523**	**0.643**	**0.773**	0.769	0.232	**0.227**

Adding each of the four extracted rules (Rules 1 through 4 in Table 6) to the protocol yielded four updated knowledge bases named respectively KB update #1, KB update #2, KB update #3, and KB update #4, while adding all four rules resulted in KB update #5, depicted in Figure 3. Injecting each updated knowledge base into NN resulted in five injected models, termed NN-I updated #1 through #5. Performance evaluation of these models, compared against the first injected model (NN-I) and the standard NN model, is presented in Table 7. NN injected with Rule 3 reported the best scores for TPR and FNR. It excelled in predicting cases in the challenging Region 2, where the clinical protocol fails, achieving 62% accuracy in this region, compared to 48% for the uninjected model and 53-55% for the other injected models. NN injected with Rule 1

reported the second-best scores for these metrics due to improved prediction in Region 2 and almost perfect prediction in Region 1. All injected models with updated rules reported TPR and FNR scores at least as good as those of the standard NN. However, only the injections of Rule 1 and Rule 3 improved these scores above those of NN-I. Notably, Rule 2 scored higher than Rule 3 in terms of accuracy and coverage but had a less beneficial effect on TPR, indicating that the available metrics to evaluate rules are not always predictive of the effect of adding that rule to the knowledge base. This suggests a need for novel metrics for evaluating new rules against existing ones. The model that performed the worst was NN-I Update #5, which incorporated all four rules, resulting in a complex architecture. These findings suggest that adding a few high-quality rules is more beneficial than incorporating many rules. For this reason, only one loop of knowledge injection-extraction was applied in this study. However, this approach can potentially be repeated multiple times, allowing the rule knowledge base to grow and increasingly complex knowledge to be injected. These explorations demonstrate the potential of augmenting ML models with ML-derived rules in addition to domain knowledge. They also highlight the challenges in identifying high-quality ML-derived rules for reinjection. Further investigation is required to understand the potential of this integration architecture.

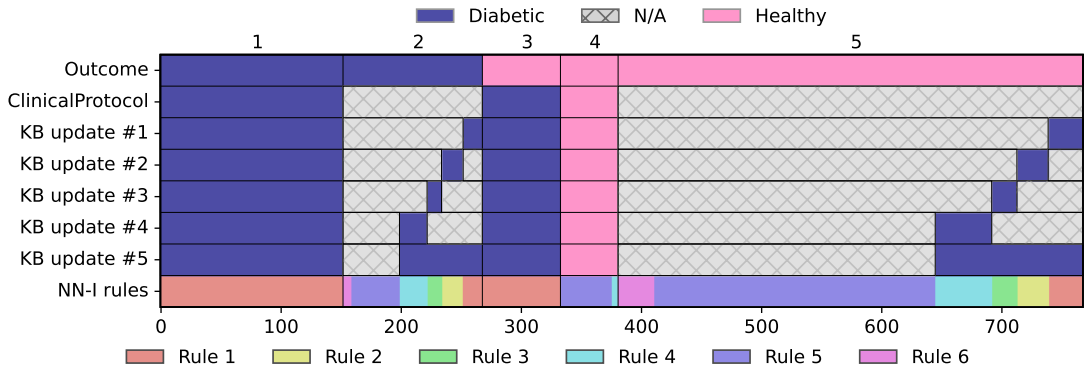

Figure 3: Clinical protocol and updated knowledge bases (KB update #1 through #5) integrating, either individually or collectively, four rules extracted from the injected neural network (NN-I rules).

5. Conclusions and future work

To leverage the potential of ML while addressing its limitations, we experimented with SKI, SKE, and their combination on a diabetes benchmark dataset. SKI effectively improved diabetes detection by enhancing recall, albeit with a reduction in precision. To increase explainability, SKE was applied, integrating the extracted rules with domain-specific knowledge, which resulted in higher recall while preserving precision. Additionally, implementing a loop that combines rule extraction and reinjection led to further performance improvements. Future research will focus on refining integration techniques and exploring additional knowledge extraction and injection methods. This includes extending knowledge representation from propositional logic to first-order logic, Datalog-like rules, and knowledge graphs.

Availability of data and code The dataset analysed is publicly available (https://www.kaggle.com/datasets/uciml/pima-indians-diabetes-database), and the code to replicate the experiments can be found in the GitHub repository (https://github.com/ChristelSirocchi/hybrid-ML).

References

[1] S. Montagna, C. Sirocchi, Hybrid personal medical assistant agents, in: 25th Workshop "From Objects to Agents", volume 3735 of *CEUR Workshop Proceedings*, CEUR-WS.org, 2024, pp. 58–72.

[2] P. Rajpurkar, E. Chen, O. Banerjee, E. J. Topol, AI in health and medicine, Nature Medicine 28 (2022) 31–38. doi:10.1038/s41591-021-01614-0.

[3] G. Ciatto, F. Sabbatini, A. Agiollo, M. Magnini, A. Omicini, Symbolic knowledge extraction and injection with sub-symbolic predictors: A systematic literature review, ACM Computing Surveys 56 (2024). URL: https://doi.org/10.1145/3645103. doi:10.1145/3645103.

[4] J. W. Smith, J. E. Everhart, W. Dickson, W. C. Knowler, R. S. Johannes, Using the adap learning algorithm to forecast the onset of diabetes mellitus, in: Proceedings of the annual symposium on computer application in medical care, American Medical Informatics Association, 1988, p. 261.

[5] E. Topol, High-performance medicine: the convergence of human and artificial intelligence, Nature Medicine 25 (2019) 44–56. doi:10.1038/s41591-018-0300-7.

[6] F. Piccialli, V. Di Somma, F. Giampaolo, S. Cuomo, G. Fortino, A survey on deep learning in medicine: Why, how and when?, Information Fusion 66 (2021) 111–137.

[7] S. Benjamens, P. Dhunnoo, B. Meskó, The state of artificial intelligence-based fda-approved medical devices and algorithms: an online database, NPJ digital medicine 3 (2020) 118.

[8] J. J. Clinton, K. McCormick, J. Besteman, Enhancing clinical practice: The role of practice guidelines., American Psychologist 49 (1994) 30.

[9] Z. Qian, W. Zame, L. Fleuren, P. Elbers, M. van der Schaar, Integrating expert odes into neural odes: pharmacology and disease progression, Advances in Neural Information Processing Systems 34 (2021) 11364–11383.

[10] C. C. Yang, Explainable artificial intelligence for predictive modeling in healthcare, Journal of healthcare informatics research 6 (2022) 228–239.

[11] Z. Obermeyer, T. H. Lee, Lost in thought — the limits of the human mind and the future of medicine, New England Journal of Medicine 377 (2017) 1209–1211.

[12] L. Von Rueden, S. Mayer, K. Beckh, B. Georgiev, S. Giesselbach, R. Heese, B. Kirsch, J. Pfrommer, A. Pick, R. Ramamurthy, et al., Informed machine learning–a taxonomy and survey of integrating prior knowledge into learning systems, IEEE Trans. on Knowledge and Data Engineering 35 (2021) 614–633.

[13] C. Sirocchi, A. Bogliolo, S. Montagna, Medical-informed machine learning: integrating prior knowledge into medical decision systems, BMC Medical Informatics and Decision Making 24 (Suppl 4) (2024) 186. doi:https://doi.org/10.1186/s12911-024-02582-4.

[14] M. Magnini, G. Ciatto, F. Cantürk, R. Aydoğan, A. Omicini, Symbolic knowledge extraction for explainable nutritional recommenders, Computer Methods and Programs in Biomedicine 235 (2023) 107536. doi:10.1016/J.CMPB.2023.107536.

[15] S. Kierner, J. Kucharski, Z. Kierner, Taxonomy of hybrid architectures involving rule-based reasoning and machine learning in clinical decision systems: A scoping review, Journal of Biomedical Informatics (2023) 104428.
[16] M. van Bekkum, M. de Boer, F. van Harmelen, A. Meyer-Vitali, A. t. Teije, Modular design patterns for hybrid learning and reasoning systems: a taxonomy, patterns and use cases, Applied Intelligence 51 (2021) 6528–6546.
[17] A. Bochare, A. Gangopadhyay, Y. Yesha, A. Joshi, Y. Yesha, M. Brady, M. A. Grasso, N. Rishe, Integrating domain knowledge in supervised machine learning to assess the risk of breast cancer, International journal of medical engineering and informatics 6 (2014) 87–99.
[18] Z. H. Janjua, D. Kerins, B. O'Flynn, S. Tedesco, Knowledge-driven feature engineering to detect multiple symptoms using ambulatory blood pressure monitoring data, Computer Methods and Programs in Biomedicine 217 (2022) 106638.
[19] R. Gazzotti, C. Faron, F. Gandon, V. Lacroix-Hugues, D. Darmon, Extending electronic medical records vector models with knowledge graphs to improve hospitalization prediction, Journal of Biomedical Semantics 13 (2022) 1–20.
[20] J. Huang, H. Yan, J. Li, H. M. Stewart, F. Setzer, Combining anatomical constraints and deep learning for 3-d cbct dental image multi-label segmentation, in: 2021 IEEE 37th International Conference on Data Engineering (ICDE), IEEE, 2021, pp. 2750–2755.
[21] S.-C. Tsai, T.-Y. Chang, Y.-N. Chen, Leveraging hierarchical category knowledge for data-imbalanced multi-label diagnostic text understanding, in: Proceedings of the Tenth International Workshop on Health Text Mining and Information Analysis (LOUHI 2019), 2019, pp. 39–43.
[22] L.-Y. Lee, C.-H. Yang, Y.-C. Lin, Y.-H. Hsieh, Y.-A. Chen, M. D.-T. Chang, Y.-Y. Lin, C.-T. Liao, A domain knowledge enhanced yield based deep learning classifier identifies perineural invasion in oral cavity squamous cell carcinoma, Frontiers in Oncology 12 (2022).
[23] K. H. Ngan, E. Mansouri-Benssassi, J. Phelan, J. Townsend, A. d. Garcez, From explanation to intervention: Interactive knowledge extraction from convolutional neural networks used in radiology, PLOS ONE 19 (2024) 1–29.
[24] G. Kunapuli, K. P. Bennett, A. Shabbeer, R. Maclin, J. Shavlik, Online knowledge-based support vector machines, in: Machine Learning and Knowledge Discovery in Databases: European Conference, 2010, Proceedings, Part II 21, Springer, 2010, pp. 145–161.
[25] M. Magnini, G. Ciatto, A. Omicini, On the design of psyki: a platform for symbolic knowledge injection into sub-symbolic predictors, in: International Workshop on Explainable, Transparent Autonomous Agents and Multi-Agent Systems, Springer, 2022, pp. 90–108.
[26] F. Sabbatini, G. Ciatto, R. Calegari, A. Omicini, et al., On the design of psyke: a platform for symbolic knowledge extraction, in: CEUR WORKSHOP PROCEEDINGS, volume 2963, Sun SITE Central Europe, RWTH Aachen University, 2021, pp. 29–48.
[27] G. G. Towell, J. W. Shavlik, M. O. Noordewier, Refinement of approximate domain theories by knowledge-based neural networks, in: Proceedings of the eighth National conference on Artificial intelligence-Volume 2, 1990, pp. 861–866.
[28] M. Magnini, G. Ciatto, A. Omicini, et al., A view to a kill: knowledge injection via lambda layer., in: WOA, 2022, pp. 61–76.
[29] M. Magnini, G. Ciatto, A. Omicini, Knowledge injection of datalog rules via neural network structuring with kins, Journal of Logic and Computation 33 (2023) 1832–1850.