Deep Learning Final Report - Spring 2025

Project Title:

Symbolic and Neuro-Symbolic AI for Chronic Kidney Disease Classification

By:

Priyabrata Dash
PhD Student, Computer Engineering
Department of Electrical and Computer Engineering
The University of Texas at San Antonio

Supervised by:

Dr. John Parsi
Adjunct Assistant Professor (DDS/MBA/PhD)
Electrical and Computer Engineering Department The University of Texas at San Antonio

Acknowledgment

I would like to express my sincere gratitude to **Dr. John Parsi**, Adjunct Assistant Professor and instructor for this course, for teaching such a well-structured and insightful course on Deep Learning during Spring 2025. His clear explanations and real-world examples greatly enhanced my understanding of this complex subject.

I also wish to thank my research advisor, **Dr. Dharanidhara Dang**, Assistant Professor in the Department of Electrical and Computer Engineering at UTSA, for his invaluable guidance and encouragement in integrating neuro-symbolic approaches into my work.

Additionally, I am thankful to my classmates for engaging discussions and constructive feedback throughout the semester. A special thanks to my friend **Spandana** for her thoughtful suggestions and input during the preparation of this report.

Contents

1 Introduction

1.1 What is Symbolic AI?

Symbolic Artificial Intelligence (Symbolic AI), also referred to as *Good Old-Fashioned AI (GOFAI)*, is a classical paradigm of artificial intelligence that represents knowledge using explicit, human-interpretable symbols and rules. This approach was dominant from the 1950s to the 1980s and was based on formal logic, production rules, and expert systems [17]. Symbolic AI systems excel in interpretability and transparency, enabling humans to understand and trace reasoning steps.

However, these systems struggle with perception, ambiguity, noise, and generalization from raw data—areas where neural networks have shown superior performance. Symbolic AI requires hand-crafted rules, which limits scalability in complex real-world applications.

1.2 What is Neuro-Symbolic AI?

Neuro-Symbolic AI (NSAI) is a modern hybrid paradigm that integrates symbolic reasoning with deep learning. It seeks to combine the structure and clarity of logic-based systems with the pattern recognition capabilities and scalability of neural networks. NSAI systems are capable of:

- Structured reasoning over symbolic relationships
- Learning from noisy or high-dimensional data
- Providing symbolic interpretations of learned knowledge

Broadly, NSAI methods fall into two categories:

1. **Logic + Neural Networks:** Symbolic logic is injected as constraints or regularizers into neural networks during training (e.g., Logic Tensor Networks).
2. **Neural Networks → Logic:** Logic is extracted post hoc from trained neural networks through rule extraction (e.g., DeepRED, Kolmogorov–Arnold Networks).

This integration has gained increasing traction in fields where interpretability and trustworthiness are essential, such as healthcare, law, and scientific discovery [4].

1.3 Why Explainability Matters in Medicine

In domains such as healthcare, explainability is not a luxury—it is a necessity. While deep neural networks may offer high predictive accuracy, their black-box nature limits their adoption in clinical settings. Medical professionals and regulatory bodies require models whose decisions can be audited, justified, and aligned with known medical knowledge.

For example, a model predicting chronic kidney disease (CKD) should not only output a binary diagnosis but also provide insight into *why* the decision was made. Explainable AI (XAI) addresses this need through:

- **Transparent decision-making:** Clinicians can validate the model's logic against medical knowledge.
- **Regulatory compliance:** Traceable logic flows are necessary for approval in critical applications.
- **Knowledge integration:** Symbolic systems allow for incorporating domain rules (e.g., "creatinine ¿ 1.5 implies CKD").

1.4 Overview of This Work

In this report, we present a comprehensive study of symbolic and neuro-symbolic AI techniques applied to the task of chronic kidney disease (CKD) classification. We begin by evaluating traditional interpretable models such as decision trees to establish a baseline. Subsequently, we train a simple feedforward neural network and explore its performance and limitations in explainability.

To address the need for interpretable models, we delve into two complementary neuro-symbolic approaches. First, using the DeepRED algorithm, we extract symbolic rules layer by layer from the trained neural network, reconstructing a logic flow from input features to final classification. Second, we employ Kolmogorov-Arnold Networks (KANs), which are inherently symbolic by design, and demonstrate their ability to learn closed-form symbolic expressions directly from data. The extracted symbolic formula is analyzed to understand feature dependencies and nonlinear interactions.

Finally, we present ongoing experiments using Logic Tensor Networks (LTNs), where domain knowledge is expressed as first-order logic constraints embedded directly into the learning process. LTNs enable us to encode medically meaningful rules, such as "serum creatinine ¿ 1.5 implies CKD," and guide the model toward rule-consistent predictions.

This multi-stage exploration provides both empirical results and symbolic insight, paving the way for interpretable and trustworthy AI systems in clinical decision-making.

2 Dataset Overview: Chronic Kidney Disease (CKD)

2.1 Dataset Description

The **Chronic Kidney Disease (CKD) dataset** is a well-known medical dataset commonly used in machine learning research for binary classification tasks. It contains

patient-level records assessed for kidney health, including a variety of biological, physiological, and clinical indicators [8]. The primary target variable indicates whether a patient has `chronic kidney disease (ckd)` or `notckd`.

- **Size**: 400 patient samples
- **Features**: 24 attributes including age, blood pressure, blood urea, serum creatinine, red blood cell count, hemoglobin, and other clinical variables
- **Label**: Binary class – `ckd` (1) or `notckd` (0)

2.2 Key Attributes

Feature	Type	Description
`age`	Numeric	Age of the patient
`bp`	Numeric	Blood pressure (mm Hg)
`sg`	Categorical/Numeric	Specific gravity of urine
`al`	Numeric	Albumin level
`sc`	Numeric	Serum creatinine
`hemo`	Numeric	Hemoglobin level
`rbc`, `pc`, `pcc`	Categorical	Red blood cells, pus cells, pus cell clumps
`class`	Categorical	Diagnosis label: `ckd` or `notckd`

Table 1: Representative attributes in the CKD dataset

The dataset includes both numerical and categorical data types and contains missing values across several features. As such, robust preprocessing techniques including imputation and encoding are required before model training.

2.3 Motivation for Using This Dataset

This dataset is particularly appropriate for the investigation of *symbolic* and *explainable* AI techniques due to several reasons:

- It is small, interpretable, and aligned with medical expertise (e.g., nephrologists can understand the features directly).
- It involves high-stakes clinical decision-making where model transparency is essential [10].
- It allows for head-to-head comparison between black-box neural networks and interpretable neuro-symbolic models.

In the following sections, we describe the data cleaning and transformation steps used to prepare the dataset for downstream experiments.

Dataset Source:
The original dataset can be found on the UCI Machine Learning Repository and has been accessed through Kaggle [8]. Our preprocessing and modeling scripts are available on UCI website:
`https://archive.ics.uci.edu/dataset/336/chronic+kidney+disease`

3 Data Cleaning and Preprocessing

High-quality data preprocessing is foundational to the success of any machine learning or symbolic reasoning system. The Chronic Kidney Disease (CKD) dataset, like many real-world medical datasets, contained a mix of numerical, categorical, and missing values. To prepare it for training interpretable and symbolic models, the following systematic steps were performed.

3.1 Handling Missing and Ambiguous Values

Medical datasets often contain entries where certain measurements are either not recorded or expressed ambiguously (e.g., with placeholders like `?`). In the CKD dataset, such placeholders were first recognized and treated as missing values (`NaN`), allowing consistent handling using standard imputation techniques.

To retain as much information as possible without discarding valuable patient records, missing values were filled (imputed) using a domain-appropriate strategy. For categorical variables such as `rbc` (red blood cells) and `htn` (hypertension), the *most frequent value* was imputed. This assumes that the most common medical status (e.g., absence of abnormality) is a reasonable guess in the absence of data. For numerical features like blood pressure and serum creatinine, *mean imputation* was applied to preserve distribution characteristics.

This step is critical not only for improving model performance but also for maintaining fairness and reducing bias in learned decision rules—especially important when generating symbolic and interpretable models.

3.2 Encoding Categorical Features

Since most symbolic and numerical learning models cannot operate directly on non-numeric data, categorical variables were converted into numerical equivalents. For binary features (e.g., `yes`/`no`, `present`/`notpresent`, `good`/`poor`), a standard binary encoding was adopted:

- Positive or abnormal indicators (e.g., `yes`, `present`, `poor`) were encoded as 1
- Negative or normal indicators (e.g., `no`, `notpresent`, `good`) were encoded as 0

This transformation preserved the clinical semantics while making the data compatible with symbolic rule extraction pipelines used later in our neuro-symbolic experiments.

3.3 Label Transformation

The target label in the CKD dataset was originally categorical: `ckd` and `notckd`. These were transformed into a binary numeric format to facilitate classification:

- `ckd` $\rightarrow$ 1 (indicating presence of disease)
- `notckd` $\rightarrow$ 0 (indicating absence of disease)

This conversion enabled compatibility with binary classifiers and symbolic systems that assume Boolean outputs.

3.4 Feature Normalization

To ensure stable convergence in gradient-based learning and uniform feature importance in symbolic modules (e.g., decision trees and Kolmogorov-Arnold Networks), the numerical features were standardized. Each feature was transformed to have a mean of zero and a standard deviation of one.

Standardization is particularly important in hybrid neuro-symbolic systems where wide numerical variation could distort learned logic boundaries or predicate evaluations.

3.5 Dataset Splitting

To evaluate model generalization, the dataset was split into training and validation sets using *stratified sampling*. This ensured that both subsets reflected the same class distribution, an essential step for medical datasets that often suffer from class imbalance.

In the following section, we evaluate the performance of decision trees as our first symbolic learning baseline and assess their interpretability and predictive effectiveness on the CKD dataset.

4 Decision Trees and Rule-Based Learning

Decision trees are among the most fundamental and interpretable models in machine learning [19]. At their core, decision trees operate by recursively splitting the dataset into smaller subsets based on the values of input features, creating a hierarchy of conditional rules. Each internal node represents a decision based on a single feature, and each leaf node corresponds to a predicted outcome. This hierarchical and symbolic structure makes

decision trees highly transparent and easy to understand — a critical factor in medical applications where practitioners demand clarity in decision-making [10].

In the context of Chronic Kidney Disease (CKD) detection, we trained a decision tree classifier on the cleaned and preprocessed dataset. The model was designed to classify patients into two categories: CKD (1) and not-CKD (0). After training, the model was evaluated on a held-out validation set to assess its performance.

Decision Tree Structure: We have generated a complete decision tree visualization (see Figure 1) to examine the feature splits and logic pathway from root to leaf nodes.

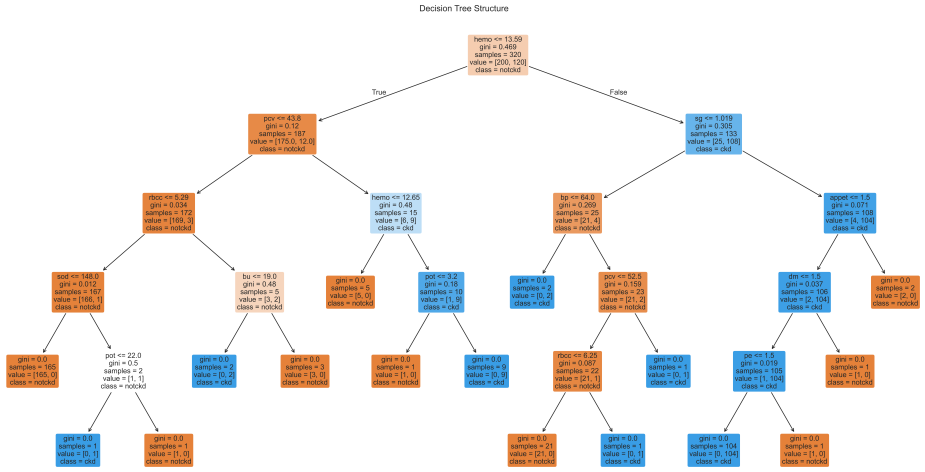

Figure 1: Visualization of the trained decision tree for CKD classification.

4.1 Evaluation Results

The table below summarizes the evaluation metrics obtained from the validation set:

Metric	Value
Accuracy	0.9625
Precision	0.9655
Recall	0.9333
F1-Score	0.9492

Table 2: Performance metrics of the decision tree model on the validation set.

The high **accuracy** and **precision** indicate that the model correctly identifies both CKD and non-CKD patients with minimal misclassifications. The **recall** of 93.33% shows that the model successfully captures the majority of CKD cases, a particularly important trait in medical diagnosis where false negatives (missed diagnoses) can be dangerous. The **F1-score**, a harmonic mean of precision and recall, further validates the overall robustness of the classifier.

4.2 Confusion Matrix

The confusion matrix below visually represents the distribution of predictions:

	Predicted: Not CKD (0)	**Predicted: CKD (1)**
True: Not CKD (0)	49	1
True: CKD (1)	2	28

Table 3: Confusion matrix of decision tree predictions.

From this matrix, we observe that out of 80 samples, only 3 were misclassified. Specifically, the model made 1 false positive and 2 false negatives, which is a strong result for a rule-based learner.

4.3 Interpretable Rules

Another key benefit of decision trees is their rule extraction capability. For example, the tree might learn interpretable rules like:

- If `serum_creatinine > 1.5` and `hemoglobin < 11`, then **CKD**
- If `albumin == 0` and `rbc == normal`, then **Not CKD**

Such rules are not only easy to understand but can also be cross-validated with domain expertise, making them ideal for human-in-the-loop decision systems in clinical settings [10].

4.4 Limitations of Decision Trees

Despite their many advantages, decision trees are not without drawbacks:

1. **Overfitting:** Decision trees are prone to overfitting the training data, especially when the tree grows too deep. This leads to reduced generalization on unseen data.
2. **Instability:** Small changes in the data can lead to completely different tree structures, which affects reproducibility.
3. **Limited Expressiveness:** Simple threshold-based splits may fail to capture complex, nonlinear relationships in the data.

While decision trees serve as an excellent starting point for interpretable modeling, their limitations motivate the exploration of more expressive and generalizable neuro-symbolic methods, such as DeepRED and Kolmogorov–Arnold Networks (KANs), which we explore in subsequent sections.

5 Neural Networks for CKD Classification

While decision trees offer interpretability and ease of use, they may struggle to fully capture complex, nonlinear relationships present in high-dimensional medical datasets. In contrast, deep learning models — particularly neural networks — excel in learning intricate patterns from data, making them suitable for tasks where precision and generalization are critical. However, their black-box nature often raises concerns in sensitive domains like healthcare, where explainability is essential. To establish a performance benchmark, we trained a small neural network on the same CKD dataset used for the decision tree model.

The architecture used was a standard feedforward neural network composed of multiple fully connected layers with non-linear activation functions. The model was trained using binary cross-entropy loss, optimized with the Adam optimizer, and evaluated on a stratified validation set to ensure balanced representation of both CKD and non-CKD classes.

5.1 Evaluation Results

Metric	**Value**
Accuracy	0.9250
Precision	0.8529
Recall	0.9667
F1-Score	0.9062

Table 4: Performance metrics of the neural network model on the validation set.

The **recall** of 96.67% is especially noteworthy, indicating the model's strong ability to detect CKD cases — a valuable trait in clinical diagnostics where failing to identify a positive case could have serious consequences. The **F1-score**, which balances precision and recall, also supports the model's overall reliability.

5.2 Confusion Matrix

	Predicted: Not CKD (0)	**Predicted: CKD (1)**
True: Not CKD (0)	45	5
True: CKD (1)	1	29

Table 5: Confusion matrix of neural network predictions.

Out of 80 validation samples, the neural network misclassified 6. This includes 5 false positives and only 1 false negative — a favorable tradeoff in medical applications where catching true positives is typically more critical than avoiding occasional false alarms.

5.3 Training Performance

Figure 2 illustrates the combined loss and accuracy trends over training epochs.

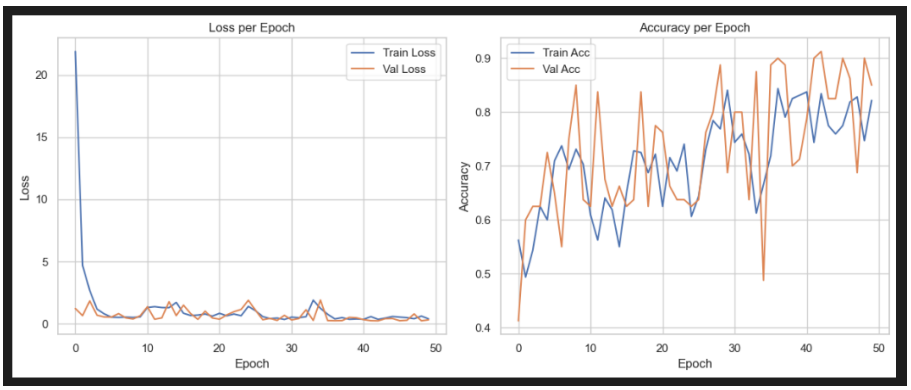

Figure 2: Training and validation loss and accuracy over 50 epochs. Final epoch: Train Loss = 0.3976, Acc = 0.8219; Val Loss = 0.3196, Acc = 0.8500.

5.4 Tradeoff Between Accuracy and Interpretability

Despite the strong predictive performance, neural networks suffer from a significant limitation: their lack of transparency. Unlike decision trees, neural networks do not provide clear, human-readable rules to explain their predictions. This opacity makes it difficult for medical professionals to validate the decision process or justify it during patient care.

Thus, while neural networks are powerful tools for classification tasks, their black-box nature creates barriers to clinical trust and deployment. This observation motivates the exploration of **Neuro-Symbolic AI** — an emerging class of models that aim to combine the learning capacity of neural networks with the interpretability of symbolic reasoning. The following sections delve into how such hybrid approaches can offer the best of both worlds: high performance with explainable logic.

6 Need for Explainable Models in Healthcare

In the domain of healthcare, the stakes of decision-making are extraordinarily high. Diagnostic models are not just abstract algorithms—they influence real lives, clinical treatments, and patient trust. In this setting, black-box models such as deep neural networks, despite their predictive power, face serious limitations due to their opacity and lack of transparency [23, 22].

Physicians and healthcare providers must often justify their decisions to patients, regulatory bodies, and ethical review boards. Models that cannot be interpreted are unlikely to be trusted, accepted, or adopted in clinical workflows. For example, if a model flags a patient as being at risk for chronic kidney disease (CKD), it is crucial to understand

which clinical features—such as elevated serum creatinine or abnormal blood urea—were responsible for this classification [11]. Without such transparency, clinicians may justifiably be skeptical of integrating AI tools into medical practice.

Black-box models, including deep learning architectures, typically offer limited insight into how predictions are made. This becomes particularly problematic when a model makes an incorrect diagnosis or recommends an unusual treatment pathway. Clinicians cannot simply accept these results without a rational explanation that aligns with domain knowledge.

To address this issue, there is a growing emphasis on **explainability** in AI. In the context of healthcare, an explainable model is one that provides *symbolic, rule-based, or otherwise human-interpretable logic* that links inputs (such as lab results or symptoms) to outputs (such as diagnoses or risk scores) [18]. This level of interpretability is not only critical for trust, but also for *clinical validation, ethical accountability, and informed consent.*

Furthermore, the recent regulatory landscape, including the European Union's proposed *Artificial Intelligence Act*, increasingly mandates that AI systems deployed in healthcare must be interpretable and auditable [3]. Models must not only perform well but must also be explainable either by design or through post-hoc analysis.

This need for interpretability has spurred the development of **Neuro-Symbolic AI**—a class of hybrid models that combine the powerful pattern recognition abilities of neural networks with the transparency and formality of symbolic reasoning. These models promise to deliver high performance while remaining transparent, auditable, and aligned with clinical reasoning.

In the next section, we explore the different categories of Neuro-Symbolic AI and how they provide a bridge between black-box learning and logical interpretability.

7 Neuro-Symbolic AI Categories

The limitations of purely symbolic systems — which often struggle with ambiguity, noise, and data incompleteness — and the black-box nature of deep neural networks — which lack interpretability — have motivated the development of **Neuro-Symbolic Artificial Intelligence (Neuro-Symbolic AI)**. This hybrid paradigm aims to unify logical reasoning with sub-symbolic learning, combining the strengths of both approaches [4, 2].

Neuro-symbolic AI systems typically fall into two primary categories: **Logic + Neural Networks**, and **Neural Networks → Logic**. These directions represent complementary strategies toward integrating learning and reasoning.

7.1 Logic + Neural Networks: Symbolic Constraints as Guidance

In this approach, symbolic knowledge is embedded into the training of neural networks as soft constraints or inductive biases. Instead of learning solely from data labels, models are guided by propositional rules, ontologies, or domain-specific logic [13]. This method enables the network to generalize better from limited data, respect known scientific relationships, and avoid logically inconsistent predictions.

One practical implementation of this paradigm is through *Logic Tensor Networks (LTNs)*, where first-order logic statements are expressed in a differentiable form and used to define loss functions that enforce logical consistency [6]. This mechanism allows models to learn while adhering to high-level symbolic principles — for instance, rules like "if creatinine is high, then CKD is likely" can be encoded as differentiable constraints during training.

7.2 Neural Networks → Logic: Extracting Symbolic Interpretations

The reverse strategy focuses on extracting symbolic representations from already trained neural models. Here, the objective is to analyze the hidden activations and decision boundaries of neural networks in order to reconstruct logical rules that approximate their behavior [25]. This approach enables transparency and explainability — especially crucial in high-stakes domains such as healthcare or autonomous systems.

Techniques like *DeepRED* decompose multi-layer networks into decision trees at each layer, translating sub-symbolic activations into symbolic conditions. This mapping allows us to understand how input features contribute to intermediate neuron activations and ultimately influence the final classification, making neural decisions verifiable and interpretable.

In the next section, we explore one such rule extraction method — DeepRED — and apply it to a neural network trained on the CKD dataset to extract transparent symbolic rules.

8 Experiment 1: DeepRED (NN → Logic)

One of the core challenges in deploying deep neural networks (DNNs) in sensitive domains like healthcare is their opaque decision-making process. To address this, we conducted an experiment using a neuro-symbolic technique known as **DeepRED**, which stands for *Rule Extraction from Deep Neural Networks*. DeepRED provides a framework to decompose a trained neural model and extract *layer-wise human-readable rules* that approximate its

internal representations and decisions [25].

In this experiment, we trained a feedforward neural network for binary classification of Chronic Kidney Disease (CKD) based on clinical features. Once trained, we applied a DeepRED-style symbolic decomposition, which involves tracing activation thresholds at each hidden layer and fitting **decision trees** to neurons to approximate their behavior. This makes it possible to derive symbolic rules — logical if-then statements — that reveal how input features influence activations and ultimately the output [5].

8.1 Input → First Hidden Layer (h1)

At the first decomposition level, we extracted rules that map directly from input features to the first hidden layer (h1). One such rule is:

$$[\texttt{wbcc} \leq 8620.00,\ \texttt{wbcc} \leq 6450.00,\ \texttt{wbcc} \leq 5150.00] \rightarrow \texttt{h1 neuron} \approx 0$$

This reveals that if the white blood cell count (wbcc) is progressively lower than three decreasing thresholds, a specific neuron in h1 becomes inactive. This symbolic trace suggests that low wbcc levels suppress early neural activations — a finding that aligns with clinical associations between wbcc and kidney function [9].

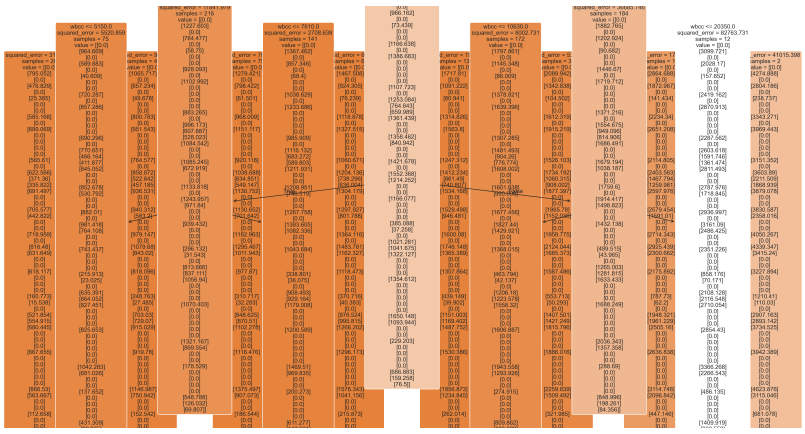

Figure 3: Decision Tree approximating Input → h1 neuron logic

8.2 h1 → h2 (First Hidden → Second Hidden Layer)

Next, we performed symbolic regression from the h1 layer to the second hidden layer (h2). A representative rule learned from this layer is:

$$[\texttt{h1\_61} \leq 698.13,\ \texttt{h1\_3} \leq 725.82,\ \texttt{h1\_29} \leq 949.69] \rightarrow \texttt{h2 neuron} \approx 0$$

This rule shows that low activations in multiple neurons from h1 together lead to the suppression of a neuron in h2. Despite being deep in the architecture, symbolic rules like these let us understand how abstract representations build up over time [16].

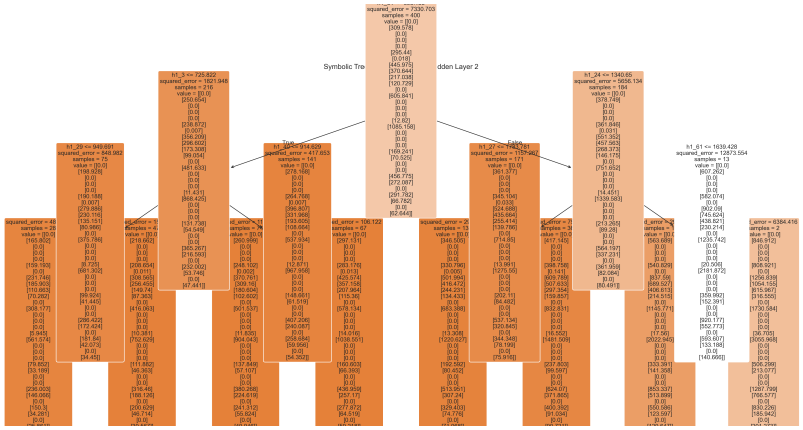

Figure 4: Decision Tree approximating h1 → h2 neuron logic

8.3 h2 → Output (CKD Prediction)

Finally, we examined the symbolic pathways from the last hidden layer (h2) to the output layer that performs binary classification. One top rule derived was:

$$[\texttt{h2\_16} \leq 11.92,\ \texttt{h2\_26} \leq 176.38,\ \texttt{h2\_16} \leq 3.84] \rightarrow \texttt{CKD}$$

This means that if specific neurons in h2 remain below certain thresholds, the model confidently predicts CKD. Such symbolic rules demonstrate that the network learns decision boundaries that can be distilled into logical expressions [20].

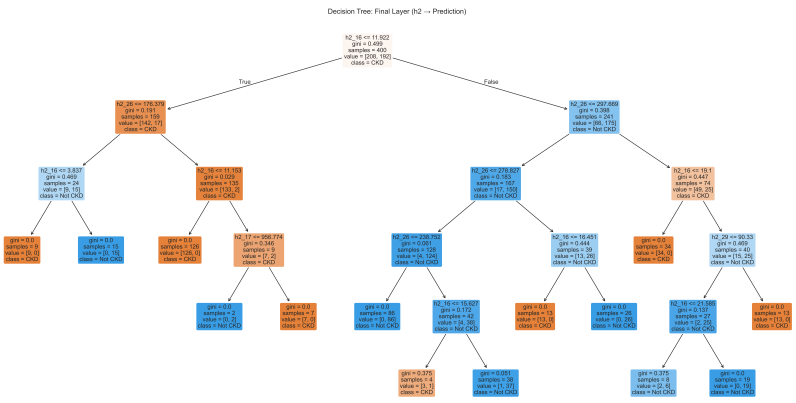

Figure 5: Decision Tree approximating h2 → Output logic

8.4 Summary Trace

To summarize the entire decision pipeline, the symbolic chain of logic learned by DeepRED is illustrated as:

Input Feature	→ h1 Activation	→ h2 Activation	→ Class
wbcc ≤ 5150	h1 neuron ≈ 0	h2_16 suppressed	CKD

8.5 Benefits of Symbolic Tracing

By applying symbolic tracing via DeepRED, we transform a black-box DNN into a transparent decision system. This process provides several advantages:

- Enables **auditability** and **trust** in clinical environments [10].
- Allows medical experts to inspect, critique, or validate model behavior [24].
- Reveals hidden pathways and potential biases [14].
- Bridges deep learning with logical reasoning for healthcare AI.

In the next section, we explore an alternative symbolic method based on *Kolmogorov-Arnold Networks*, which are symbolic by design and avoid the need for post-hoc rule extraction.

9 Experiment 2: Kolmogorov–Arnold Networks (KANs)

As an alternative to logic extraction techniques like DeepRED, we explored **Kolmogorov–Arnold Networks (KANs)**—a new class of neural architectures that are *symbolic by design*, enabling direct extraction of mathematical expressions from trained models. KANs are built upon deep theoretical underpinnings from classical approximation theory and offer a transparent alternative to traditional neural networks.

9.1 What Are KANs?

KANs are inspired by the **Kolmogorov–Arnold representation theorem**, which states that any multivariate continuous function $f(x_1, x_2, \ldots, x_n)$ can be decomposed as a finite sum of compositions of univariate continuous functions:

$$f(x_1, x_2, \ldots, x_n) = \sum_{q=1}^{2n+1} \phi_q \left(\sum_{p=1}^{n} \psi_{q,p}(x_p) \right)$$

This theorem, initially proved by Andrey Kolmogorov and later refined by Vladimir Arnold [12, 1], guarantees that any function can be represented using a shallow network composed of nonlinear univariate functions and linear weights.

KANs operationalize this idea by *replacing traditional fixed activation functions with learnable univariate functions on each edge*, implemented as interpolating splines. This design allows for flexible, expressive, and *analytically tractable* modeling [7].

9.2 Why KANs Are Symbolic by Design

Unlike classical neural networks, which require post-hoc interpretation or rule extraction, KANs allow one to *directly read the functional form* of the learned relationships. Each unit in a KAN corresponds to a symbolic expression built from a predefined library of basis functions (e.g., $\sin(x)$, x^2, e^x). After training, symbolic regression is applied automatically to convert numerical representations into **closed-form analytic expressions**, providing clear human interpretability.

This design bridges the gap between black-box deep learning and symbolic reasoning, making KANs well-suited for high-stakes domains like healthcare where explainability is crucial [10].

9.3 Training KAN on CKD Dataset

We trained a small KAN on the Chronic Kidney Disease (CKD) dataset from the UCI repository [8], using 24 normalized clinical features as input. The network was defined with a width of `[24, 10, 1]`, grid size 5, and polynomial order $k = 3$. We used the LBFGS optimizer with regularization to promote sparsity and smoothness. After 100 training steps, we pruned unused edges and re-optimized the model.

9.4 Extracting Symbolic Formula

After training, we applied automatic symbolic regression to extract the following decision function:

$$\begin{aligned} f(x) = &-0.0017x_1 - 0.0013x_{10} + 0.00015x_{11} - 0.0012x_{12} - 0.00056x_{13} + 0.0669x_{15} \\ &+ 0.0067x_{16} - 1.53 \times 10^{-5}x_{17} + 0.0080x_{18} - 0.0041x_2 \\ &+ 0.0384(-0.5581x_6 - 1)^2 + 0.0037(-0.7270x_7 - 1)^2 \\ &+ 0.1433\sin(0.3717x_4 + 8.4441) + 0.0575\sin(0.4727x_5 + 5.1152) \\ &- 0.1488 \end{aligned}$$

This symbolic expression shows that the model integrates linear, quadratic, and sinusoidal features in a compact interpretable form.

9.5 Interpretation: What Features Matter?

- **Strong linear weights:** Variables x_{15} and x_{16} (potentially serum creatinine or albumin levels) have high positive influence on the output.

- **Nonlinear components:** Quadratic terms for x_6, x_7 and sinusoidal terms for x_4, x_5 indicate non-monotonic decision regions—possibly capturing biological thresholds or periodic signals.
- **Sparse coefficients:** Many coefficients are close to zero, demonstrating the model's ability to discard irrelevant features.

In summary, KANs enable a hybrid model that is both data-driven and interpretable, producing symbolic rules that clinicians can potentially validate or use to generate hypotheses. In the next section, we explore another symbolic framework — Logic Tensor Networks — which incorporates logic constraints directly into the training process.

10 Ongoing Work: Logic Tensor Networks (LTNs)

As part of our broader investigation into interpretable AI for healthcare, we explored **Logic Tensor Networks (LTNs)** — a neuro-symbolic framework that integrates deep learning with *first-order fuzzy logic*, enabling AI models to reason over both data and symbolic rules simultaneously. LTNs are particularly well-suited to domains like medicine, where expert knowledge in the form of symbolic constraints can enhance both *accuracy* and *trustworthiness* of predictions [21, 6].

10.1 What Are LTNs?

LTNs combine continuous logic with neural networks by embedding **first-order logic (FOL)** into real-valued tensors. The core idea is to interpret truth values not as binary {0,1}, but as values in $[0, 1]$, which allows logical expressions to be differentiable and seamlessly integrated into backpropagation-based learning. These expressions are encoded as *constraints* on a neural network's outputs, enabling a model to optimize for both data fit and logical consistency [15].

For example, a rule like:

$$\forall x \; \texttt{creatinine}(x) > 1.5 \rightarrow \texttt{CKD}(x)$$

can be embedded directly into the training process using fuzzy logic implication and satisfied to a high degree via gradient descent.

LTNs were first introduced by Serafini and d'Avila Garcez [21] as a framework for grounding logic in vector spaces, where constants, predicates, and functions are all represented by tensors and neural modules. Recent frameworks such as `LTNtorch` [15] have made these ideas practical by implementing them in PyTorch.

10.2 Incorporating Fuzzy Logic into Deep Learning

LTNs allow one to define symbolic priors alongside regular loss functions. For instance, in the context of CKD classification, we can specify:

- **Supervised logic:**

$$\forall x \, \texttt{HasCKD}(x) \Leftrightarrow \texttt{label}(x)$$

  This enforces that the model's predicate for "has CKD" matches the ground-truth label.

- **Domain logic (prior knowledge):**

$$\forall x \, \texttt{creatinine}(x) > 1.5 \rightarrow \texttt{HasCKD}(x)$$

  This injects a medically meaningful rule to influence model behavior in borderline or low-data cases.

These logical clauses are compiled into differentiable expressions via fuzzy operators (e.g., Łukasiewicz, Gödel, Product logic), and optimized together with standard data loss using an aggregated satisfiability objective [6].

10.3 High-Level Training Sketch on CKD

Our prototype used `LTNtorch` [15], a PyTorch-based implementation of LTNs. We defined:

- A predicate `HasCKD(x)` modeled as a neural network.
- Variables `x` (patients) and `y` (labels).
- Two core constraints:
  1. Supervised equivalence between prediction and label.
  2. Symbolic implication based on the rule: "If serum creatinine is high, CKD should be predicted".

Training was done using the Adam optimizer. The logic loss was computed as the mean violation across axioms. We observed that the model began respecting symbolic rules, even when data alone was ambiguous — showcasing the power of logic-guided learning.

10.4 Relevance for CKD

In medical datasets like CKD, data is often sparse, noisy, and incomplete. Purely data-driven models may generalize poorly or learn spurious correlations. LTNs offer a principled way to inject domain rules, such as:

- `eGFR < 60` implies CKD,
- `Serum creatinine > 1.5` is abnormal,
- `No CKD` if all renal markers are within normal range.

These constraints enhance model robustness and interpretability, making the system more clinically acceptable and reliable [10].

In the next section, we summarize our findings and present concluding remarks on the value of neuro-symbolic AI for interpretable and trustworthy healthcare decision systems.

11. Summary and Conclusion In this report, we explored a comprehensive neuro-symbolic approach to classifying Chronic Kidney Disease (CKD) using interpretable and trustworthy machine learning models. Starting from traditional symbolic methods and advancing toward hybrid neural-symbolic systems, our experiments and analysis revealed critical insights into model performance, explainability, and practical medical relevance.

We began by preprocessing the CKD dataset and applying a standard decision tree classifier. While decision trees offered clear, rule-based logic understandable to clinicians, they lacked the flexibility and depth to capture complex patterns in the data. This motivated the use of neural networks, which provided significantly improved classification performance but came at the cost of interpretability — a known barrier in high-stakes fields like healthcare.

To bridge this gap, we explored Neuro-Symbolic AI, a paradigm that integrates symbolic reasoning with neural learning. We categorized neuro-symbolic models into two types:

Logic + NN: Neural models constrained or guided by symbolic rules.

NN $\rightarrow$ Logic: Logic and rules extracted post hoc from trained neural networks.

In Experiment 1, we applied DeepRED, a layer-wise rule extraction method that decomposed a neural network into symbolic rules. We showed how CKD predictions could be traced from input features through hidden layers to the output via interpretable decision paths. This provided a level of transparency necessary for clinical decision support.

In Experiment 2, we implemented Kolmogorov–Arnold Networks (KANs), a novel neural architecture capable of learning symbolic expressions directly from data. The extracted symbolic formula revealed which biomarkers mattered most, highlighted nonlinear dependencies, and presented a mathematical expression that clinicians could interpret and validate. This positions KANs as a promising tool for interpretable deep learning.

Lastly, we discussed our ongoing work with Logic Tensor Networks (LTNs), which bring fuzzy logic into deep learning. By incorporating domain-specific symbolic constraints (e.g., "creatinine ¿ 1.5 $\rightarrow$ CKD"), LTNs allow medical expertise to guide learning, improving both generalization and trust.

In conclusion, this work demonstrates that combining deep learning with symbolic reasoning is not only feasible but essential for building accountable, trustworthy, and interpretable AI in medicine. As machine learning continues to evolve, neuro-symbolic frameworks offer a balanced approach — merging data-driven insight with human-understandable logic.

Future work may include integrating patient-specific longitudinal data, expanding symbolic knowledge bases, and deploying models into real-world clinical workflows, where explainability is not just an advantage but a requirement.

References

[1] V.I. Arnold. On functions of three variables. *Doklady Akademii Nauk SSSR*, 114:679–681, 1957.

[2] Tarek R. Besold, Artur d'Avila Garcez, Sebastian Bader, Howard Bowman, Pedro Domingos, Pascal Hitzler, Kai-Uwe Kühnberger, Luis C. Lamb, Daniel Lowd, Priscila Machado Vieira Lima, et al. Neural-symbolic learning and reasoning: A survey and interpretation. In *Neural-Symbolic Learning and Reasoning*, pages 1–59. Springer, 2017.

[3] European Commission. Proposal for a regulation on a european approach for artificial intelligence, 2021. `https://eur-lex.europa.eu/legal-content/EN/TXT/?uri=CELEX:52021PC0206`.

[4] A. d'Avila Garcez, T. R. Besold, L. De Raedt, P. Földiak, P. Hitzler, T. Icard, and D. Silver. Neuro-symbolic ai: The state of the art. *Trends in Cognitive Sciences*, 27(4):319–334, 2023.

[5] Kedar Dhamdhere, Mukund Sundararajan, and Qiqi Yan. How important is a neuron? *arXiv preprint arXiv:1905.12292*, 2019.

[6] Ivan Donadello, Luciano Serafini, and Artur d'Avila Garcez. Logic tensor networks for semantic image interpretation. *Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence (IJCAI)*, pages 1596–1602, 2017.

[7] Xuanyu Dong, Stefanie Jegelka, Tommi Jaakkola, and David Sontag. Symbolic neural networks with kolmogorov–arnold representations. arXiv preprint arXiv:2401.10036, 2024. `https://arxiv.org/abs/2401.10036`.

[8] Dheeru Dua and Casey Graff. Uci machine learning repository: Chronic kidney disease dataset. `https://archive.ics.uci.edu/ml/datasets/Chronic_Kidney_Disease`, 2019.

[9] W.G. Guder, S. Narayanan, H. Wisser, and B. Zawta. *Samples: From the Patient to the Laboratory*. Wiley-VCH, 1996.

[10] A. Holzinger, C. Biemann, C. Pattichis, and D. Kell. What do we need to build explainable ai systems for the medical domain?, 2017. arXiv preprint arXiv:1712.09923.

[11] A. Holzinger, G. Langs, H. Denk, K. Zatloukal, and H. Müller. Causability and explainability of artificial intelligence in medicine. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, 2019.

[12] Andrey N Kolmogorov. On the representation of continuous functions of several variables by superposition of continuous functions of one variable and addition. *Doklady Akademii Nauk SSSR*, 114:953–956, 1957.

[13] Robin Manhaeve, Sebastijan Dumancic, Angelika Kimmig, Thomas Demeester, and Luc De Raedt. Deepproblog: Neural probabilistic logic programming. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 31, 2018.

[14] Gary Marcus. The next decade in ai: Four steps towards robust artificial intelligence. *arXiv preprint arXiv:2002.06177*, 2020.

[15] Giuseppe Marra, Robin Manhaeve, Eugenia Ternovska, and Luc De Raedt. Ltntorch: Differentiable first-order logic in pytorch. In *Neuro-Symbolic AI Workshop at NeurIPS*, 2023.

[16] Grégoire Montavon, Wojciech Samek, and Klaus-Robert Müller. Methods for interpreting and understanding deep neural networks. *Digital Signal Processing*, 2018.

[17] Allen Newell and Herbert A. Simon. Computer science as empirical inquiry: Symbols and search. *Communications of the ACM*, 19(3):113–126, 1976.

[18] W. N. Price and I. G. Cohen. Privacy in the age of medical big data. *Nature Medicine*, 2019.

[19] J. Ross Quinlan. Induction of decision trees. *Machine Learning*, 1(1):81–106, 1986.

[20] Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. "why should i trust you?": Explaining the predictions of any classifier. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 1135–1144, 2016.

[21] Luciano Serafini and Artur S. d'Avila Garcez. Learning and reasoning with logic tensor networks. *arXiv preprint arXiv:1606.04422*, 2016.

[22] E. H. Shortliffe and M. J. Sepúlveda. Clinical decision support in the era of artificial intelligence. *JAMA*, 2018.

[23] E. Tjoa and C. Guan. A survey on explainable artificial intelligence (xai): Toward medical xai. *IEEE Transactions on Neural Networks and Learning Systems*, 2020.

[24] Sasha Tonekaboni, Shalmali Joshi, Melissa D McCradden, and Anna Goldenberg. What clinicians want: Contextualizing explainable machine learning for clinical end use. *Machine Learning for Healthcare Conference*, 2019.

[25] Jan R. Zilke, Eneldo Loza Mencía, and Frederik Janssen. Deepred–rule extraction from deep neural networks. In *International Conference on Discovery Science*, pages 457–473. Springer, 2016.