Integrating Graph Neural Networks and Fuzzy Logic to Enhance Deep Learning Interpretability

Giovanna Castellano, Raffaele Scaringi, Gennaro Vessio and Gianluca Zaza*

Department of Computer Science, University of Bari Aldo Moro, Italy

Abstract

We propose a novel methodology that combines Graph Neural Networks (GNNs) with Fuzzy Logic to enhance the interpretability of deep learning models. GNNs handle structured data, while Fuzzy Logic provides a framework that excels in handling uncertainty and imprecision. To solve the challenge of interpretability in GNNs, we present a novel approach that marries GNNs' expressive power with Fuzzy Logic's readability. Preliminary experiments show promising results, indicating the potential of this approach to create AI systems that are transparent and trustworthy.

Keywords

eXplainable Artificial Intelligence, Graph Neural Networks, Fuzzy Logic

1. Introduction

In recent years, the proliferation of Artificial Intelligence (AI) across many applications has underscored the critical need for intelligent but also interpretable and trustworthy systems. This necessity has given rise to eXplainable Artificial Intelligence (XAI), which seeks to address the "black box" nature of many AI models, particularly deep learning models [1]. XAI aims to make the decision-making processes of AI systems transparent, understandable, and accountable to users, a demand increasingly stressed in ethical guidelines and regulatory frameworks worldwide [2].

The quest for XAI has led to exploring various methodologies, each with strengths and limitations. With its roots in logic and explicit knowledge representation, symbolic AI offers a pathway to interpretability but often at the expense of flexibility and scalability [3]. Conversely, the advent of deep learning has unlocked unprecedented capabilities in handling complex, high-dimensional data, albeit often exacerbating the opacity of the decision-making process.

Amidst these dichotomies, the integration of Graph Neural Networks (GNNs) [4] and Fuzzy Inference Systems (FIS) [5] emerges as a promising frontier. GNNs, with their ability to capture complex relational structures within data, offer a robust framework for modeling complex interactions in diverse domains, from social networks to molecular biology. However, their interpretability remains challenging, limiting their utility in applications demanding transparency.

EXPLIMED: First Workshop on Explainable Artificial Intelligence for the Medical Domain at ECAI 2024, October 19-20, 2024, Santiago de Compostela, Spain

*Corresponding author.

giovanna.castellano@uniba.it (G. Castellano); raffaele.scaringi@uniba.it (R. Scaringi); gennaro.vessio@uniba.it (G. Vessio); gianluca.zaza@uniba.it (G. Zaza)

0000-0002-6489-8628 (G. Castellano); 0000-0001-7512-7661 (R. Scaringi); 0000-0002-0883-2691 (G. Vessio); 0000-0003-3272-9739 (G. Zaza)

Fuzzy Inference Systems, on the other hand, grounded in Fuzzy Logic, excel in handling uncertainty and imprecision—traits inherent in real-world data. By articulating knowledge in the form of fuzzy rules, FIS provides a mechanism for reasoning that is both interpretable and adaptable, offering insights into the decision-making process that are intuitively understandable.

Therefore, the synthesis of GNNs and FIS represents an innovative approach to surmounting the barriers of interpretability and flexibility in AI. This paper explores this novel *neuro-symbolic* approach, demonstrating its potential to advance the state of the art in XAI. By bridging the gap between the rich learning capabilities of GNNs and the interpretability afforded by Fuzzy Logic, we propose a hybrid method that incorporates Fuzzy Logic into a GNN so that the graph-based knowledge acquired from data can be easily expressed in the form of interpretable fuzzy rules. Our basic idea is to apply a "fuzzification" process to transform the input features into fuzzy variables with associated linguistic terms. This granular form of input features allows for the adaptation of tabular data to graph representations so that a GNN can be applied to learn a classification model from data. To further enhance explainability, we apply GNNExplainer, which effectively improves the transparency of the graph-based model by analyzing the nodes that were most significant in the classification task [6]. The proposed method is a preliminary step toward developing effective, efficient, transparent, and trustworthy AI systems, thereby aligning AI technologies more closely with human values and ethical standards.

The rest of this paper is structured as follows. Section 2 reviews the state of the art. Section 3 describes the proposed method. Section 4 presents preliminary quantitative and qualitative experiments. Section 5 summarizes our findings, draws conclusions, and outlines future works.

2. Related Work

In recent years, Graph Neural Networks have emerged as powerful tools for modeling complex relationships and structures in data represented as graphs [7]. GNNs have experienced a significant surge in popularity, being used in various contexts ranging from biomedical data to social networks [8]. However, while GNNs excel at capturing complex patterns, their interpretability can be challenging. To address this issue and enhance the understandability of GNN-based models, researchers have begun exploring hybrid approaches that integrate GNNs with symbolic methods [9].

In particular, Fuzzy Logic [5] can express knowledge using fuzzy variables with associated linguistic terms, thereby providing descriptions understandable to humans. This promotes transparency and interpretability in decision-making, which is crucial in many domains. The fuzzy theory has been used in neural networks for many years, and various hybrid neural architectures have been created that incorporate fuzzy components, as the work described in [10]. One of the most famous examples in the literature is the Adaptive Neuro-Fuzzy Inference System (ANFIS) that has found applications in various domains [11, 12, 13].

By incorporating Fuzzy Logic, which deals with uncertainty and imprecision in data, hybrid models aim to provide clearer explanations for AI systems' decisions. This combination allows for a more intuitive understanding of how neural networks process information and make predictions, thus fostering trust and transparency in AI applications. Moreover, leveraging the complementary strengths of neural networks and Fuzzy Logic can lead to more robust and adaptable models that excel in various domains where interpretability is crucial, such as healthcare, finance, and social networks.

However, the achievements in hybrid methods that combine FIS with neural networks have not been mirrored in GNNs. Specifically, using GNNs within neuro-symbolic approaches has not been extensively explored in the literature [9]. In particular, there is a small handful of hybrid systems that combine GNNs and Fuzzy Logic. An example is Fuzzy GNN (FGNN) [14], a meta-learning method for few-shot learning that employs an edge-focused GNN to perform the edge prediction by iteratively updating the edge labels. According to the output of edge prediction, a fuzzy membership function is designed to achieve more exact relationship representations for node classification.

To our knowledge, our work is the first attempt to embed fuzzy theory into GNNs to achieve accurate and interpretable models.

3. Methodology

We introduce a novel method that integrates Fuzzy Logic with GNNs to improve model interpretability while retaining performance. The proposed method involves the following steps:

1. Fuzzification: The first step involves transforming crisp data inputs into fuzzy representations, allowing for the incorporation of uncertainty and imprecision inherent in real-world data.
2. Graph construction: Following fuzzification, the method constructs a graph representation of the data, where nodes represent input features and related linguistic terms.
3. Graph-based classification: Using the constructed graph, the method applies a GNN to make predictions based on the learned patterns and relationships encoded in the graph. This step leverages the rich expressiveness of GNNs to model complex relational structures within the data effectively.
4. Graph-based explanation: One of the distinguishing features of the method is its ability to provide intuitive and simple explanations for the classification decisions made by the model. By analyzing the graph structure and activations of the GNN, the method generates interpretable explanations, shedding light on the reasoning behind each prediction.
5. Fuzzy rule extraction: The method extracts salient fuzzy rules from the learned graph-based model, encapsulating the decision-making logic in a human-understandable format. These fuzzy rules capture the underlying patterns and relationships the model identifies, offering valuable insights into the data and the decision-making process.

In the following, we formalize each step of the proposed method.

3.1. Fuzzification

In this section, we detail the process of *fuzzification* [15], a critical initial step in our approach to integrating Fuzzy Logic with GNNs for XAI. We begin with a labeled dataset $D = \{(\mathbf{x}_d, y_d)\}_{d=1}^{N_D}$, where $\mathbf{x}_d \in \mathbb{R}^n$ represents the input feature vector corresponding to the d-th data point and y_d denotes the associated ground truth label for the class of $\mathbf{x}_d$.

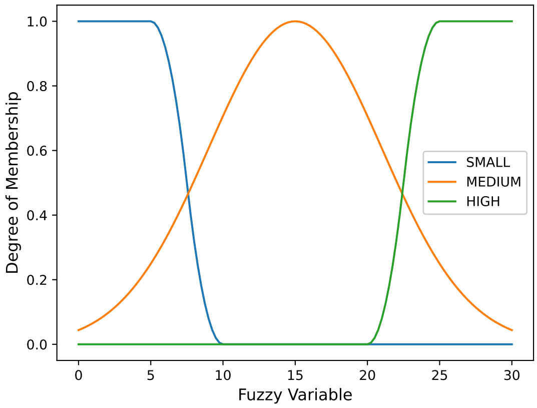


Figure 1: Membership functions for fuzzy sets corresponding to the linguistic terms SMALL, MEDIUM, and HIGH.

We initially fuzzify the input values to address the inherent uncertainty in the data. Specifically, each input feature $x_i, i = 1 \ldots n$, is granulated into M_i fuzzy sets $A_{i1}, A_{i2}, \ldots, A_{iM_i}$. For this study, we maintain a uniform number of fuzzy sets [5] across all input features, thereby setting $M_i = M$ for all $i = 1 \ldots n$. Each fuzzy set A_{ij} is characterized by a membership function $\mu_{ij} : X_i \subseteq \mathbb{R} \rightarrow [0, 1]$, with $\mu_{ij}(x_i) = m_{ij}$, which calculates membership values for the input features. Consequently, each x_i is associated with M membership values m_{ij}, where $j = 1 \ldots M$. In this preliminary investigation, we opt for $M = 3$ and label the fuzzy sets A_{i1}, A_{i2}, A_{i3} with the linguistic terms SMALL, MEDIUM, and HIGH, respectively.

The membership functions for these fuzzy sets are illustrated in Fig. 1. Specifically, we employ a *z-function* for the LOW term, a *gaussian function* for the MEDIUM term, and a *s-function* for the HIGH term. These functions were selected to reflect the domain of discourse accurately. The deployment of "open-ended" fuzzy sets at both extremities of the domain implies that our model can accommodate values that extend beyond the observed range in the dataset, thereby ensuring a holistic representation of the phenomenon under investigation.

To establish the parameters for the fuzzy sets, we determine the 25th, 50th, and 75th percentiles of the data distribution. These percentiles serve as critical reference points for defining the core and support of each fuzzy set, ensuring that the membership functions are appropriately aligned with the underlying data distribution.

The outcome of the fuzzification phase is a "fuzzified" dataset $D_F = \{(\mathbf{x}_d, \mathbf{m}_d, y_d)\}_{d=1}^{N_D}$, wherein the original data points are augmented with fuzzy membership values for each input feature. This fuzzified dataset forms the basis for subsequent phases of our approach.

3.2. Graph Construction

As a further data engineering phase, we represent each data point as a direct graph, defined as a couple $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, where $\mathcal{V}$ is the set of vertices and $\mathcal{E} \subseteq \mathcal{V} \times \mathcal{V}$ is the set of edges. To this end, given a data point $(\mathbf{x}, \mathbf{m}) \in D_F$, the associated graph instance is created by composing a graph

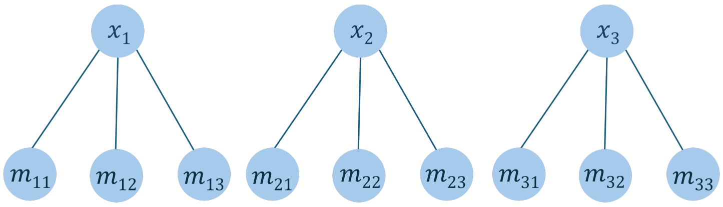


Figure 2: Example of a generic data point transformed in a graph instance. Each feature x_i is linked to all the membership value nodes m_{ij}, symbolizing the membership of the given feature to the corresponding fuzzy set.

structure with nodes for each feature x_i and each fuzzy set A_{ij}. Nodes corresponding to fuzzy sets have the membership value m_{ij} as feature value. Edges are established by connecting each feature node to its corresponding fuzzy set nodes. Formally, for each data point, we define $\mathcal{V} = \{x_1, x_2, \ldots, x_n, m_{11}, m_{12}, \ldots, m_{nM}\}$ and $\mathcal{E} = \{(x_i, m_{ij})$ with $i = 1, \ldots, n, j = 1, \ldots, M\}$.

Figure 2 shows an example of a graph created with three input features x_1, x_2, x_3 each connected to three fuzzy sets, leading to the following configuration: $x_1, x_2, x_3, m_{11}, m_{12}, m_{13}, m_{21}, m_{22}, m_{23}, m_{31}, m_{32}, m_{33}$.

3.3. Graph-Based Classification

After applying the fuzzification process to the dataset D and creating the corresponding graphs, a new dataset, denoted as D_G, is generated as a set of labeled graphs. Formally, we have $D_G = \{(\mathcal{G}_d, y_d) \mid \mathcal{G}_d = (\mathcal{V}_d, \mathcal{E}_d)\}_{d=1}^{N_D}$. This transformation converts the initial classification task into a graph classification task. We employ a graph-based methodology to address this issue, training a Graph Convolutional Network (GCN) [16]. As depicted in Fig. 3, the input graph instance $\mathcal{G}_d$ is fed into the GCN to encode its contextual information. Specifically, the GCN learns iterative node representations, namely for each node $x_i \in \mathcal{G}_d$, a node embedding h_i is computed by aggregating its neighborhoods, for each layer l designed in the GCN. Formally:

$$h_i^{(l+1)} = \sigma\left(W_i^{(l)} \sum_{j \in \mathcal{N}(i) \cup \{i\}} \frac{\hat{A}_{j,i}}{\sqrt{\hat{D}_{jj}\hat{D}_{ii}}} h_i^{(l)}\right),$$

or in matrix formulation:

$$H^{(l+1)} = \sigma\left(\hat{D}^{-\frac{1}{2}}\hat{A}\hat{D}^{-\frac{1}{2}}H^{(l)}W^{(l)}\right),$$

where σ is a general activation function (e.g., ReLU), $\hat{A} = A + I$ is the adjacency matrix of the input graph with inserted self-loops, $\hat{D}$, with $\hat{D}_{ii} = \sum_{j=0} \hat{A}_{ij}$ is the diagonal degree matrix associated with $\hat{A}$, $H^{(l)}$ is the node embedding matrix calculated at the l-th layer, $W^{(l)}$ is a trainable parameter matrix, and $\mathcal{N}$ is the neighborhood function defined as $\mathcal{N} : \mathcal{V} \mapsto \mathcal{V}^*$ such that $\mathcal{N}(v_i) = \{v_j | (v_i, v_j) \in \mathcal{E} \lor (v_j, v_i) \in \mathcal{E}\}$.

The GCN acts as an encoder that, given an instance graph $\mathcal{G}_d$, refines its initial node feature matrix X thanks to the graph convolutions to produce another instance graph $\mathcal{Z}_d$, which has

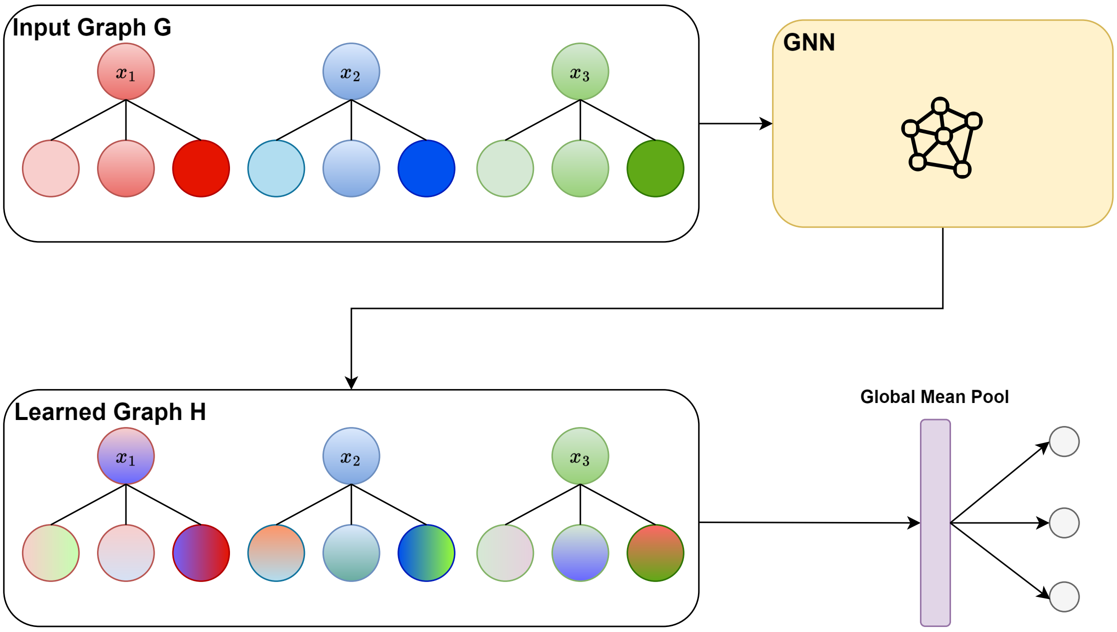


Figure 3: Our proposed method. The input graph corresponding to a fuzzified data point is fed to the GNN. Then, a global mean pool layer compacts all the learned node embeddings into a single feature vector, which is passed to a final linear layer for computing the final classification.

the same structure of $\mathcal{G}_d$ but has a different node feature matrix, denoted as H. Generating a distinct feature vector representing the graph in a vector space is essential to classify an entire input graph instance. To achieve this, we compress the entire refined feature matrix $H \in \mathbb{R}^{n \times h}$, where h denotes the hidden embedding dimensionality, applying a pooling layer to compute a feature-wise average and producing a comprehensive graph feature vector $\mathbf{p}_d \in \mathbb{R}^h$. Finally, $\mathbf{p}_d \in \mathbb{R}^h$ is fed to a classification head layer to compute the output class, represented as a softmax-activated class probability distribution. After training the GCN, we obtain a base model Φ that can classify test instances represented as graphs.

3.4. Graph-Based Explanation

This phase of our framework is devoted to deriving explanations from the graphs corresponding to test instances' classifications using GNNExplainer [6] to determine the significance of edges in the most influential subgraph for a given prediction. GNNExplainer operates with a given input graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, its associated feature matrix $X \in \mathbb{R}^{n \times d}$, where d denotes the input node feature dimensionality (in this work it is set to 1, since we represent each node with a scalar feature), the adjacency matrix $A \in \{0, 1\}^{n \times n}$, the base model Φ, and its prediction $\hat{y} = \Phi(X, A)$. Specifically, this tool learns jointly two sets of parameters, namely $X_F \in [0, 1]^d$, defined as a feature mask vector, and $A_F \in [0, 1]^{n \times n}$ representing the adjacency mask. The process begins by slightly modifying the input graph's structure. This involves updating its feature matrix $\tilde{X} \leftarrow X \odot X_F$, and its adjacency matrix $\tilde{A} \leftarrow A \odot A_F$, where $\odot$ represents the element-wise matrix product. Subsequently, A_F and X_F undergo optimization using cross-entropy loss between $\hat{y}$ and $\tilde{y}$, where $\tilde{y} = \Phi(\tilde{X}, \tilde{A})$.

After finalizing this optimization phase, parameters nearing 1.0 accentuate pivotal node characteristics or connections within the input graph, delineating the most significant subgraph for a specific prediction. Therefore, given any test instance, we obtain the corresponding subgraph containing only edges crucial for the specific prediction.

This corresponds to pruning off useless fuzzy sets for each input variable. Indeed, for all the data in the test set, we calculate the average of the activation values of the edges and prune off the edges with a value above a threshold (defined empirically) of 0.50. Due to edge pruning, scenarios may arise where all linguistic terms associated with an input variable are pruned. In such cases, the variable is consequently eliminated, leading to automatic feature selection.

3.5. Fuzzy Rule Extraction

Given a specific input instance, the final step is to convert the subgraph obtained from the graph-based explanation phase into a linguistic fuzzy rule. To achieve this, we assign the relevant fuzzy linguistic terms for each specific instance in the prediction to every node corresponding to a fuzzy variable. This is done using the “is” connector, which signifies the membership of a specific instance to a fuzzy linguistic term. In cases where multiple fuzzy linguistic terms for the same fuzzy variable are required for prediction, we employ the “OR” connector. Otherwise, we apply the “AND” connector. If an input variable’s value lacks representation by any linguistic term with sufficient degree, it is deemed irrelevant and excluded from the antecedent of the fuzzy rule. This process continues until all input nodes have been processed, thereby establishing the antecedent of the fuzzy rule. For the definition of the fuzzy rule’s consequent, we append the name of the output variable, succeeded by the prediction outcome, following the “THEN” connector.

An example of extracting a fuzzy rule is shown in Fig. 4. The example has the same structure as Fig. 2, but since it is in the prediction phase of the result, the entire subgraph is represented, including the target class (y) with its corresponding result (y_1). GNNExplainer identifies the most important edges, which are colored in red. These edges are the connections from fuzzy variable x_1 to linguistic term m_{13}, from fuzzy variable x_2 to linguistic terms m_{22} and m_{23}, and from fuzzy variable x_3 to linguistic term m_{32}. Subsequently, the significant edges are connected to the target variable. The final result obtained is the following fuzzy rule:

IF x_1 is m_{13} AND x_2 is m_{22} OR m_{23} AND x_3 is m_{32}
THEN y is y_1

4. Experimental Results

To show the effectiveness of the proposed graph-based neuro-symbolic method, preliminary experiments were carried out on two standard datasets: the Iris dataset[1] and the Haberman

[1] Iris dataset: https://www.kaggle.com/datasets/arshid/iris-flower-dataset

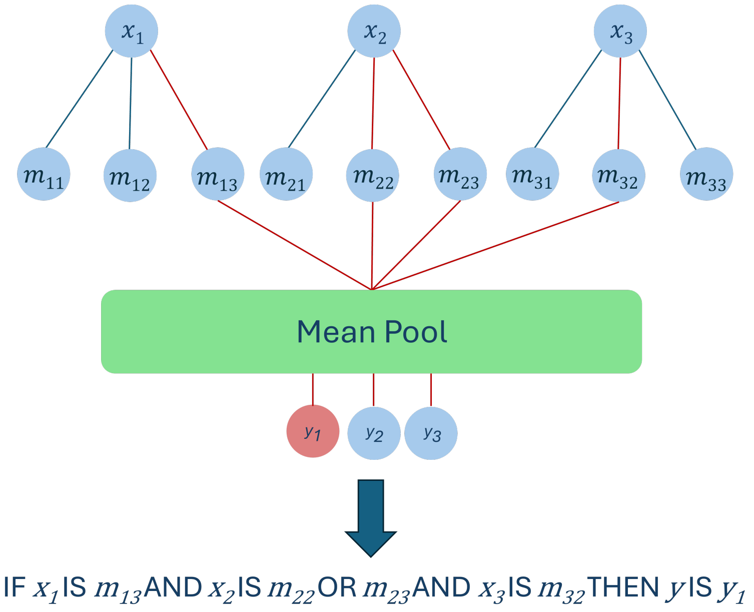


Figure 4: Example of fuzzy rule extracted from a given data point. Edges highlighted in red symbolize the most influential connection in the graph, reflecting that the feature x_i takes as membership value m_{ij}. On the contrary, blue edges represent non-influential links in the graph.

dataset.[2] To better evaluate the quality of the models produced by the proposed method in terms of accuracy and interpretability, they were compared with models generated by the ANFIS neuro-fuzzy network. This four-layer feed-forward neural network reflects a fuzzy rule base in its parameters and topology [17].

The same experimental setup was used to train the ANFIS neuro-fuzzy network for a fair comparison. The dataset was divided into 60% training set, 10% validation set, and 30% test set using the holdout method. A search for optimal hyperparameters was performed using the Hyperopt algorithm[3] [18]. The following hyperparameters of the GCN were optimized: number of layers, number of hidden channels, aggregation function, batch size, and learning rate. To train the GCN, we used the PyTorch Geometric (PyG)[4] library that offers various types of architectures. In this preliminary study, we limited our investigation to the basic Graph Convolutional Network, without testing different types of graph neural layers.

4.1. Classification of Iris Flowers

The Iris dataset consists of 150 samples of Iris flowers, each belonging to one of three species: *Setosa*, *Versicolor*, or *Virginica*. For each sample, four features were measured: *sepal length*, *sepal width*, *petal length*, and *petal width*. The goal is to predict the species of Iris flowers based on their feature measurements.

[2] Haberman dataset: https://www.kaggle.com/datasets/gilsousa/habermans-survival-data-set
[3] Hyperopt library: https://github.com/hyperopt/hyperopt
[4] PyG library: https://pytorch-geometric.readthedocs.io/en/latest/

Dataset	**Method**	**Accuracy**	**Precision**	**Recall**	**F1**
Iris	*Our method*	0.91	0.92	0.91	0.91
	ANFIS	0.93	0.93	0.93	0.93
Haberman	*Our method*	0.68	0.54	0.53	0.53
	ANFIS	0.70	0.54	0.52	0.52

Table 1
Comparison between our proposed method and ANFIS.

As shown in Table 1, the proposed graph-based approach and ANFIS exhibited similar performance levels in accuracy, with our method yielding a marginally lower accuracy score than ANFIS. However, our method overcomes ANFIS in terms of readability. Indeed, the model generated by ANFIS includes $81 (= 3^4)$ rules, and each rule contains all the input variables in the antecedent part. Due to this high number of fuzzy rules, the ANFIS model can be hard to read and interpret. Figure 5 shows some of the 81 rules generated by ANFIS. Conversely, the model generated by our method is highly interpretable since it provides a single rule for classifying a test instance, and each rule contains only significant input variables and useful fuzzy sets. For example, Fig. 6 shows the sub-graph generated by GNNExplainer for classifying a specific test instance. It can be seen that only edges with strong activation (red edges) are retained because they are considered significant for the final classification. The input variable *petal width* is weakly represented by its linguistic terms (blue edges), and it is not connected to the output node, indicating that it does not affect the final prediction. The sub-graph can be easily translated into a compact linguistic form as follows:

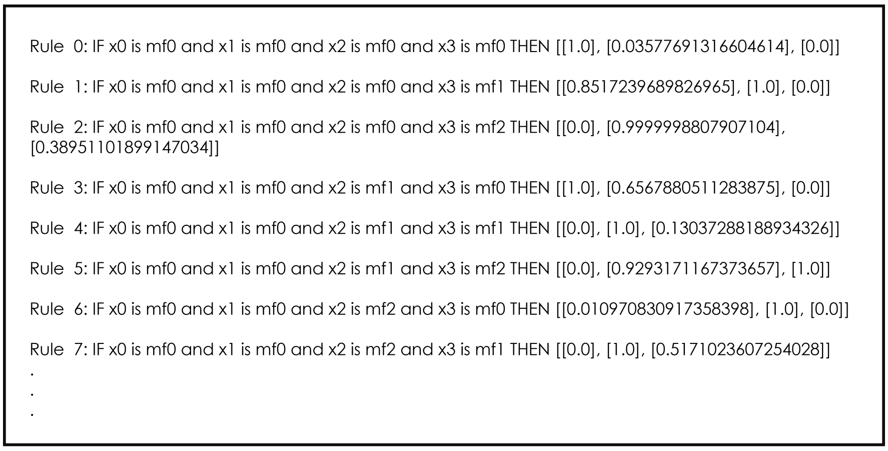
IF *sepal lenght* is *low* OR *medium* AND *sepal width* is *low* OR *medium*
AND *petal length* is *low* OR *high*
THEN *type* is *Setosa*

Our method gives users an understandable description of the prediction for a given test instance, presented in both graphical form (the sub-graph) and textual form (the linguistic IF-THEN rule).

4.2. Classification of Survival of Patients

The Haberman dataset is a classic dataset in the field of medical research and machine learning, containing information about patients who underwent surgery for breast cancer at the University of Chicago's Billings Hospital between 1958 and 1970. It consists of 306 instances and three input features: *age* (the age of the patient at the time of surgery), *year* (the year of the surgery), and *nodes* (the number of positive axillary lymph nodes detected). The target variable is the *status*, i.e., the survival status of the patient after surgery, categorized as either 0 (survived for 5 or more years) or 1 (died within 5 years).

Table 1 summarizes the comparative results. It can be seen that our method has slightly lower accuracy than ANFIS. However, our model outperforms ANFIS in terms of interpretability as it provides a synthetic rule for a given instance. Figure 7 shows an example of an explanation generated by our method for an instance of the Haberman dataset. It can be seen that the



```
Rule  0: IF x0 is mf0 and x1 is mf0 and x2 is mf0 and x3 is mf0 THEN [[1.0], [0.0357769131 6604614], [0.0]]

Rule  1: IF x0 is mf0 and x1 is mf0 and x2 is mf0 and x3 is mf1 THEN [[0.8517239689826965], [1.0], [0.0]]

Rule  2: IF x0 is mf0 and x1 is mf0 and x2 is mf0 and x3 is mf2 THEN [[0.0], [0.9999998807907104],
[0.38951101899147034]]

Rule  3: IF x0 is mf0 and x1 is mf0 and x2 is mf1 and x3 is mf0 THEN [[1.0], [0.6567880511283875], [0.0]]

Rule  4: IF x0 is mf0 and x1 is mf0 and x2 is mf1 and x3 is mf1 THEN [[0.0], [1.0], [0.13037288188934326]]

Rule  5: IF x0 is mf0 and x1 is mf0 and x2 is mf1 and x3 is mf2 THEN [[0.0], [0.9293171167373657], [1.0]]

Rule  6: IF x0 is mf0 and x1 is mf0 and x2 is mf2 and x3 is mf0 THEN [[0.010970830917358398], [1.0], [0.0]]

Rule  7: IF x0 is mf0 and x1 is mf0 and x2 is mf2 and x3 is mf1 THEN [[0.0], [1.0], [0.5171023607254028]]
.
.
.
```

Figure 5: Some rules generated by ANFIS.

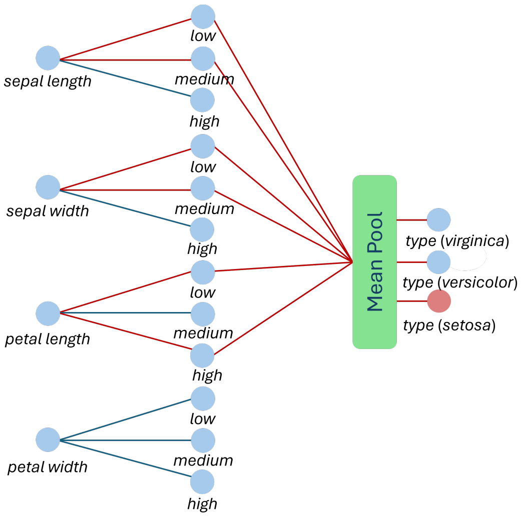


Figure 6: Example of sub-graph explaining the classification of an instance belonging to class *Setosa*.

pruning mechanism has effectively removed the linguistic term *high* from the fuzzy partition of the variable *age* due to its activation exceeding the threshold of 0.50. Starting from this sub-graph, the following linguistic rule is derived:

IF *age* is *low* OR *medium* AND *years* is *medium* OR *high* AND *nodes* is *medium* THEN *status* is 0

5. Conclusions

In this study, we proposed a new model that combines Graph Neural Networks with Fuzzy Logic to enhance the interpretability of deep learning models. To test our model, we conducted

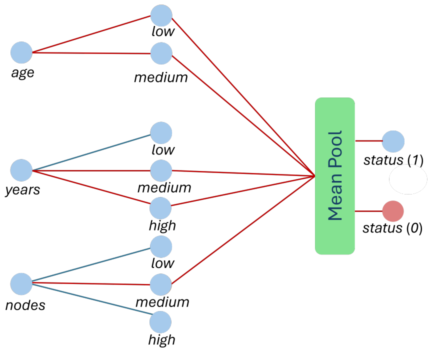


Figure 7: Example of sub-graph generated by our method for an instance of the Haberman dataset.

a preliminary study comparing our method with the ANFIS neuro-fuzzy network using two benchmark datasets (Iris and Haberman). The quantitative results indicate that our model achieved results similar to those of the neuro-fuzzy network. In detail, there is a 2% accuracy difference in favor of the neuro-fuzzy model for both tested datasets. We can confirm that our model achieves levels of robustness comparable to those of ANFIS.

Regarding the qualitative analysis, our model features a "one-shot" rule generation mechanism, explaining the prediction for each instance. On the other hand, the neuro-fuzzy model does not operate this way. It generates different fuzzy rules according to the number of fuzzy variables and fuzzy linguistic terms. Thus, ANFIS obtained 81 and 27 fuzzy rules for the Iris and Haberman datasets, respectively. Therefore, a significant reduction in interpretive complexity can be observed. Using our model in a real-world context would allow domain experts to explain the prediction outcome instantly.

In future work, it will be essential to compare our model with traditional machine learning models to extend the experiments by increasing the number of datasets. Lastly, particular attention will be given to scalability and computational complexity, critical aspects of the proposed model's applicability in real-world contexts.

6. Acknowledgments

Giovanna Castellano and Gianluca Zaza acknowledge funding support from the FAIR - Future AI Research (PE00000013) project, Spoke 6 - Symbiotic AI (CUP H97G22000210007), under the NRRP MUR program funded by NextGenerationEU. The research objectives of this paper are in partial fulfilment of the project EXPLICIT (CUP H93C23000890005). A Ph.D. fellowship funds Raffaele Scaringi's research within the Italian "D.M. n. 352, April 9, 2022" – under the NRRP, Mission 4, Component 2, Investment 3.3 – Ph.D. project "Automatic analysis of artistic heritage via Artificial Intelligence", co-supported by Exprivia S.p.A. (CUP H91I22000410007). All authors are members of the INdAM GNCS research group.

References

[1] D. Gunning, E. S. Vorm, J. Y. Wang, M. Turek, DARPA's Explainable AI (XAI) program: A retrospective, Applied AI Letters (2021).

[2] N. Díaz-Rodríguez, J. Del Ser, M. Coeckelbergh, M. López de Prado, E. Herrera-Viedma, F. Herrera, Connecting the dots in trustworthy Artificial Intelligence: From AI principles, ethics, and key requirements to responsible AI systems and regulation, Information Fusion 99 (2023) 101896. doi:{https://doi.org/10.1016/j.inffus.2023.101896}.

[3] A. Sheth, K. Roy, M. Gaur, Neurosymbolic Artificial Intelligence (Why, What, and How), IEEE Intelligent Systems 38 (2023) 56–62. doi:{10.1109/MIS.2023.3268724}.

[4] L. Wu, P. Cui, J. Pei, L. Zhao, X. Guo, Graph neural networks: foundation, frontiers and applications, in: Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, 2022, pp. 4840–4841.

[5] L. Zadeh, Fuzzy sets, Information and Control 8 (1965) 338–353. doi:{https://doi.org/10.1016/S0019-9958(65)90241-X}.

[6] Z. Ying, D. Bourgeois, J. You, M. Zitnik, J. Leskovec, GNNExplainer: Generating Explanations for Graph Neural Networks, in: H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, R. Garnett (Eds.), Advances in Neural Information Processing Systems, volume 32, Curran Associates, Inc., 2019.

[7] F. Scarselli, M. Gori, A. C. Tsoi, M. Hagenbuchner, G. Monfardini, The Graph Neural Network Model, IEEE Transactions on Neural Networks 20 (2009) 61–80. doi:{10.1109/TNN.2008.2005605}.

[8] J. Zhou, G. Cui, S. Hu, Z. Zhang, C. Yang, Z. Liu, L. Wang, C. Li, M. Sun, Graph neural networks: A review of methods and applications, AI Open 1 (2020) 57–81. doi:{https://doi.org/10.1016/j.aiopen.2021.01.001}.

[9] L. C. Lamb, A. Garcez, M. Gori, M. Prates, P. Avelar, M. Vardi, Graph neural networks meet neural-symbolic computing: A survey and perspective, arXiv preprint arXiv:2003.00330 (2020).

[10] N. Talpur, S. J. Abdulkadir, H. S. A. Alhussian, M. H. B. Hasan, N. Aziz, A. M. Bamhdi, Deep Neuro-Fuzzy System application trends, challenges, and future perspectives: a systematic survey, Artificial Intelligence Review 56 (2022) 865–913.

[11] M. Tektaş, Weather forecasting using ANFIS and ARIMA models, Environmental Research, Engineering and Management 51 (2010) 5–10.

[12] R. Nagpal, D. Mehrotra, A. Sharma, P. Bhatia, ANFIS method for usability assessment of website of an educational institute, World Applied Sciences Journal 23 (2013) 1489–1498.

[13] G. Casalino, G. Castellano, U. Kaymak, G. Zaza, Balancing accuracy and interpretability through neuro-fuzzy models for cardiovascular risk assessment, in: 2021 IEEE Symposium Series on Computational Intelligence (SSCI), IEEE, 2021, pp. 1–8.

[14] T. Wei, J. Hou, R. Feng, Fuzzy Graph Neural Network for Few-Shot Learning, in: 2020 International Joint Conference on Neural Networks (IJCNN), 2020, pp. 1–8. doi:{10.1109/IJCNN48605.2020.9207213}.

[15] G. J. Klir, B. J. C. Yuan, Fuzzy sets and fuzzy logic, 1995.

[16] T. N. Kipf, M. Welling, Semi-supervised classification with graph convolutional networks, arXiv preprint arXiv:1609.02907 (2016).

[17] J.-S. Jang, C.-T. Sun, Neuro-fuzzy modeling and control, Proceedings of the IEEE 83 (1995) 378–406.

[18] J. Bergstra, D. Yamins, D. D. Cox, Making a Science of Model Search: Hyperparameter Optimization in Hundreds of Dimensions for Vision Architectures, in: International Conference on Machine Learning, 2013.