2025 International Conference on Intelligent Computing
July 26-29, Ningbo, China
https://www.ic-icc.cn/2025/index.php

Neural Rule Learning with Network Architecture Search for Interpretable Classification

Xincheng He[1], Xueting Jiang[1], Haoran Liu[2], Shuo Guan[3], Jiayu Xue[1], Bowen Shen[1], and Yuangang Wang[1](✉)

[1] SEAC Key Laboratory of Big Data Applied Technology, Dalian Minzu University, Dalian Liaoning 116600, China
wyg@dlnu.edu.cn
[2] College of Computer Science and Technology, Chongqing University of Posts and Telecommunications, Chongqing 400065, China
[3]School of Computer Science and Engineering, Northeastern University, Shenyang Liaoning 110819, China

Abstract. Deep neural network models have achieved unprecedented success in modeling unstructured data across tasks such as computer vision, speech recognition, and natural language processing. However, their inherent limitations in model transparency and interpretability hinder the analysis of underlying mechanisms, prediction processes, and decision rationales, restricting their application in critical domains such as healthcare, finance, and judiciary. Traditional rule-based models exhibit strong transparency and interpretability. However, they are overly dependent on feature engineering, which significantly increases the cost of human intervention. Furthermore, their limited capability to represent data restricts the efficient and comprehensive utilization of large-scale datasets. While ensemble learning methods can enhance predictive performance, they often do so at the expense of model interpretability. To address the challenge of balancing predictive performance and interpretability in structured data classification tasks, we proposes an interpretable classification method that integrates neural rule learning with network architecture search in this paper. On the one hand, this method automatically learns interpretable logical rules to represent and classify the data. On the other hand, by incorporating network architecture search techniques, the model adapts to the characteristics of the dataset and determines the optimal network structure to achieve superior predictive performance. Through comparative experiments with different types of classification models across multiple datasets, we find that the proposed method demonstrates strong competitiveness in both predictive accuracy and interpretability. It is capable of generating highly interpretable logical rules while maintaining excellent predictive performance.

Keywords: Neural Rule Learning, Network Architecture Search, Interpretable Classifier, Tabular Data, Deep Neural Network.

1 Introduction

Deep learning technologies, represented by deep neural networks, have dominated the field of machine learning due to their powerful representation learning capabilities and have driven technological innovations in numerous complex tasks [11]. However, in some fields where model decision results require high levels of trust and transparency, the "black box" nature of deep neural networks has become a major obstacle to their large-scale application [9]. In the healthcare field, doctors require models to provide accurate diagnostic results. They also need to understand how the model derives conclusions based on the patient's specific physiological characteristics. This transparency is crucial for enhancing the reliability of the diagnostic results and supporting better medical decision-making [23]. Similarly, in the financial field, interpretability is crucial for the model's compliance and credibility because financial decisions are subject to strict audits and regulations. Only transparent and easily understandable model decisions can gain trust and meet policy and regulatory requirements[3]. In the judicial field, decisions often involve complex legal provisions and case backgrounds. Judgments not only need to be accurate and fair but also require a transparent reasoning process, so that they can be reviewed and questioned when necessary [6]. Therefore, striking a balance between model performance and decision-making transparency, while ensuring interpretability and auditability of model outputs, has emerged as a pivotal challenge in these high-stakes domains.

Research on interpretable learning predominantly focuses on post-hoc interpretability methods and the enhancement of inherently transparent models. Post-hoc interpretability methods aim to elucidate the relationship between input features and prediction outcomes by analyzing the behavior of complex models. Prominent approaches include LIME [19], which provides instance-specific explanations by constructing local linear approximations; Grad-CAM [21], which visualizes the regions of deep learning models that contribute to predictions through heatmaps; and rule extraction techniques that translate predictive logic into symbolic representations, such as decision trees [32] or logical expressions. A significant advantage of post-hoc methods is their broad applicability to nearly all model types, allowing for rapid deployment without modifying the model architecture or training process. However, as these methods lack direct access to the internal mechanisms of the models, the explanations they generate may not fully capture the true decision-making logic. Moreover, most post-hoc methods are limited to local analysis, which lacks global interpretability and comprehensive transparency [20].

Meanwhile, improving the predictive performance of transparent models represents another significant approach in interpretable modeling. In recent years, many researchers have applied the stacked generalization principle in ensemble learning to enhance traditional transparent models, such as decision trees and rule-based systems. This strategy enables these models to process data layer by layer, perform built-in feature transformations, and achieve sufficient model complexity. Representative methods include deep forest [35] and multi-layer gradient boosting decision trees [7] based on decision tree, and deep fuzzy system based on Takagi-Sugeno-Kang [33,34] and Wang-Mendel fuzzy models [25,26]. Although these methods have significantly improved the predictive performance of transparent models, the original interpretability of the base models tends to diminish as the deep models become more complex through multi-layer stacking. Consequently, such methods are often unsuitable for applications that demand high levels of transparency and interpretability.

Neuro-symbolic learning has recently been viewed as a promising approach to bridging the gap between model predictive performance and interpretability. In particular, neural rule learning integrates the powerful representational capacity of neural networks with the transparency of rule-based models. By introducing the representation and learning of logical rules, these methods enable models to automatically generate formalized rule sets. Representative approaches include DRNet [15], RRL [27], and ENRL [22], which transform rule generation into a differentiable optimization problem by simulating the behavior of logical operators, thereby efficiently generating rule sets using gradient-based mechanisms. However, these methods still face challenges such as high rule complexity, limited capability in modeling feature interactions, and reliance on manually designed model structures. Therefore, developing more effective strategies for model structure determination to achieve a balance between predictive performance and interpretability has become a key focus of current research.

Neural architecture search has emerged as a innovative method that leverages strategies such as reinforcement learning, evolutionary algorithms, or gradient-based optimization to automatically explore network hyperparameters and structural configurations, thereby identifying optimal architectures for specific tasks [18]. Compared to traditional manual model design, it significantly reduces human intervention and dynamically adjusts network depth and width according to task requirements, which enhances both model adaptability and generalization capabilities. On the basis of the aforementioned research trends and developments, We propose an interpretable classification method that integrates neural rule learning with neural architecture search in this paper. The main contributions of this study are summarized as follows.

- We propose an innovative classification method that integrates neural rule learning with neural architecture search. The method effectively balances predictive performance and interpretability by automatically generating logical rule sets to represent structured data while optimizing the network structure to adapt to different datasets.
- The proposed method enables the automatic extraction of interpretable logical rules from data, which provides human-understandable explanations for the model's predictions. Unlike traditional rule-based models, this method achieves both high transparency and low human intervention cost, and it is more suitable for large-scale structured data applications.
- Through extensive comparative experiments on multiple structured datasets, the proposed method demonstrates superior performance in both predictive accuracy and interpretability compared to traditional rule-based models, ensemble methods, and other kinds of classification approaches. The results highlight the model's capability to generate highly interpretable logical rules without compromising predictive performance.

The remainder of this paper is organized as follows. Section 2 reviews related works on neural rule learning and neural architecture search. Section 3 introduces the proposed method including overall structure and implementation details. Section 4 presents the experimental setup and result analysis. Finally, Section 5 concludes the paper and outlines future research directions.

2 Related Works

2.1 Neural Rule Learning

In recent years, significant progress has been made in neural rule learning. For instance, Qiao $et\ al.$ propose a two-layer neural network-based learning model that automatically generates disjunctive normal form rule sets, which can achieve a balance between predictive accuracy and rule simplicity in classification tasks [15]. Wang $et\ al.$ propose a rule-based representation learner for classification tasks, which leverages an innovative gradient grafting training method to enable the automatic learning of non-fuzzy rules, thereby improving model performance while balancing interpretability and scalability [27]. Shi $et\ al.$ propose an explainable neural rule learning method, which integrates the expressive power of neural networks with the interpretability of rule-based systems. By employing a rule voting mechanism, it enhances model transparency and robustness, while significantly mitigating performance degradation on out-of-distribution data [22]. Yu $et\ al.$ propose a feature interactive neural rule learning method, which formulates rule learning as a differentiable discrete combination encoded by a feedforward neural network and incorporates contextual embeddings to represent both rules and conditions [31]. Despite the significant potential of these methods in improving both model performance and interpretability, neural rule learning still faces several challenges. On one hand, the generated rule sets may become overly complex in certain cases, which increases the difficulty of interpretation. On the other hand, most existing approaches have limited capability in capturing deep feature relationships. Moreover, the rule generation process heavily relies on the model architecture design, which constrains its adaptability across multi-task and multi-domain scenarios. Future research could focus on simplifying and optimizing rule sets, exploring automated rule generation systems, and enhancing the model's generalization and robustness in complex tasks to achieve a better balance between performance and interpretability

2.2 Neural Architecture Search

Neural architecture search is an automated method for generating neural network architectures by optimizing a predefined search space. This approach significantly reduces the reliance on manual expertise while improving design efficiency and model performance. Consequently, many researchers have conducted in-depth studies on key aspects of neural architecture search, including search space design, optimization of search strategies, and performance prediction methods. Real $et\ al.$ employ a controller network to generate candidate architectures and dynamically adjust the policy to optimize network performance [17]. Pham $et\ al.$ efficiently select optimal architectures from a large pool of candidate networks by introducing genetic mutation and selection mechanisms from evolutionary algorithms [14]. Liu *et al* adopt a differentiable search framework, which significantly improves search efficiency by formulating the architecture search process as a differentiable optimization problem [12]. Neural architecture search has achieved significant success in unstructured data modelling tasks and has been extensively applied in fields such as computer vision and natural language processing. Therefore, leveraging neural architecture search within neural rule learning models to address structured data classification problems presents substantial potential for further advancement. This integration not only facilitates the automated generation of highly interpretable rule sets but also enhances the model's adaptability across various tasks and data distributions.

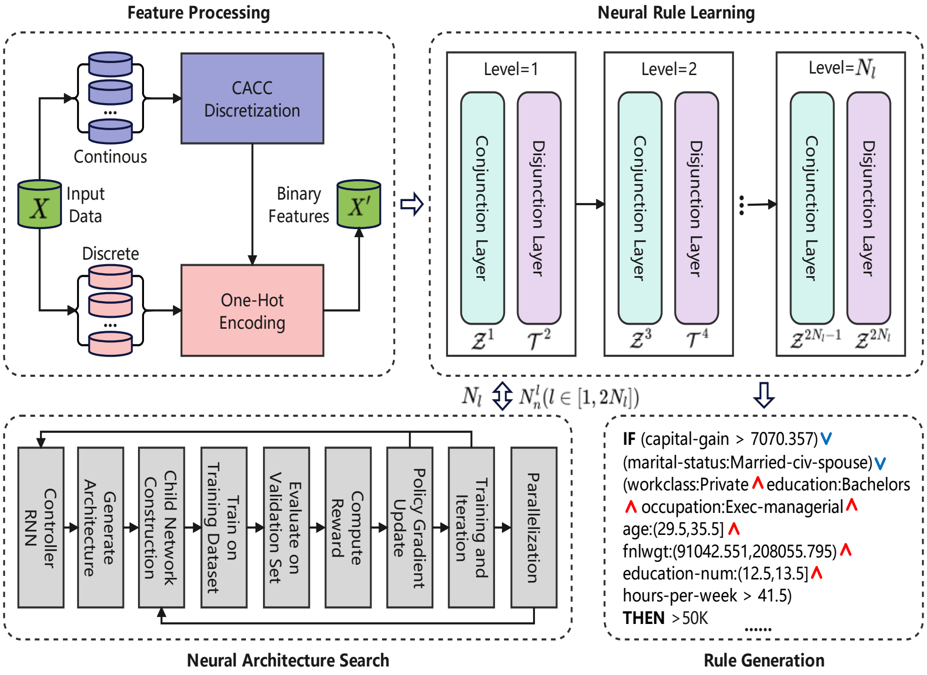

Fig. 1. The model structure of our proposed method

3 Methodology

This paper proposes an interpretable classification method based on neural rule learning with neural architecture search(NRL-NAS), which aims to address the trade-off between performance and interpretability in traditional classification methods. By integrating the transparency of rule generation with the flexibility of network architecture optimization, this method is capable of delivering efficient and interpretable classification results in complex tasks. As shown in **Fig. 1.** The model structure of our proposed method, the proposed method is consisted of feature preprocessing, neural rule learning, neural architecture search, and rule generation.

3.1 Feature Processing

Suppose $X = \{(x_1, y_1), \dots, (x_{N_s}, y_{N_s})\}$ represents the training dataset, where N_s is the number of samples. Here, x_i denotes the feature vector of the i-th sample, $x_{i,j}$ denotes the j-th feature value of the i-th sample, and y_i denotes the class label of the training sample x_i. All features in the training dataset X can be either continuous or discrete values, while all class variables are discrete. Because continuous features cannot be directly used in logic-based rules, we must perform a discretization process on the continuous feature values in the training dataset X. To improve the model's predictive performance, we use a continuous feature discretization method based on the class-attribute contingency coefficient [24], which is more suitable for classification tasks.

For a continuous attribute a to be discretized, we first determine its minimum value a_{min} and maximum value a_{max}. We then collect all distinct values of a and sort them in ascending order to form the set Ω_a. For every pair of adjacent data points in Ω_a, we compute their midpoint to form the candidate boundary set B, and we also include a_{min} and a_{max} in B. We initialize the discretization scheme $D = \{[a_{min}, a_{max}]\}$ and set the evaluation metric $G_{cacc} = 0$, which measures the quality of the current scheme. Subsequently, the algorithm enters an iterative process. In each iteration, it sequentially attempts to insert any boundary $b \in B$ that has not been included in the current discretization scheme D, thereby forming a new partition D'. The corresponding CACC value of this new scheme is then computed according to equation (1) and (2).

$$CACC = \sqrt{\frac{\alpha}{\alpha + N_s}} \tag{1}$$

$$\alpha = \frac{N_s[(\sum_{p=1}^{N_c} \sum_{q=1}^{N_i} \frac{{h_{p,q}}^2}{N_{p,+} N_{+,q}}) - 1]}{\log(N_i)} \tag{2}$$

Here, N_c is the number of classes, N_i is the number of intervals, $h_{p,q}$ is the number of samples of the p-th class in the q-th interval, $N_{p,+}$ is the number of samples that belong to p-th class, $N_{+,q}$ is the number of samples whose values of attribute A fall in q-th interval. After all candidate boundaries are attempted, the solution D' maximizing the CACC value is selected and compared with the current optimal solution. The current solution $D = D'$ and global optimum $G_{cacc} = CACC_{D'}$ are updated, with the interval count N_i incremented by 1, if $CACC_{D'} > G_{cacc}$ or the interval limit is not yet reached. Otherwise, the iteration is terminated. The process continues iteratively until no further improvement in G_{CACC} is observed or the interval limit is met, finally outputting the discretization scheme D'.

After the continuous attributes in dataset X have been discretized using the aforementioned method, we employ one-hot encoding to transform them into binary features. The discrete attributes and the class labels in X are similarly processed by means of one-hot encoding. Consequently, we convert X into $X' = \{(x_1', y_1'), \ldots, (x_{N_s}', y_{N_s}')\}$, where $x_i' \in \{0,1\}^{N_b}$ is a binary feature vector of length N_b, and $y_i' \in \{0,1\}^{N_c}$ is a binary vector whose dimension equals the number of classes N_c.

3.2 Rule Learning and Generation

To achieve semantic rules that simultaneously possess interpretability and classification capability, we propose and construct a neural rule learning model inspired by the Concept Rule Sets (CRS) and the Multi-layer Logical Perceptron (MLLP). The CRS architecture consists of N_l hierarchical levels, each comprising a conjunction layer followed immediately by a disjunction layer, resulting in a total of $2N_l$ layers in the model. Formally, for any $l \in \{1,2N_l\}$, if l is odd, then $\mathcal{Z}^l$ denotes the conjunction layer, whereas if l is even, then $\mathcal{T}^l$ denotes the disjunction layer.

Except for the input layer, each of the remaining layers contains a predetermined number of nodes, as well as edges that connect these nodes to the nodes in the previous layer. Let N_n^l denote the number of nodes in the l-th layer of the CRS, and let W^l represent the adjacency matrix between the l-th and $(l-1)$-th layers, with a shape of $N_n^l \times N_n^{l-1}$, where $W_{i,j}^l \in \{0,1\}$. If there exists an edge connecting the i-th node in the l-th layer and the j-th node in the $(l-1)$-th layer, then $W_{i,j}^l = 1$; otherwise, $W_{i,j}^l = 0$.

Assume that z_i^l denotes the i-th node in layer $\mathcal{Z}^l$, which represents a conjunction operation over all the nodes in layer $\mathcal{T}_{l-1}$ that are connected to z_i^l. Similarly, t_i^l denotes the i-th node in layer $\mathcal{T}^l$, which represents a conjunction operation over all the nodes in layer $\mathcal{Z}_{l-1}$ that are connected to t_i^l. The definitions of these two types of nodes are shown in equation (3).

$$\begin{cases} z_i^l = \bigwedge_{W_{i,j}^l=1} t_j^{l-1} \\ t_i^{l+1} = \bigvee_{W_{i,j}^{l+1}=1} z_j^l \end{cases} \tag{3}$$

If the number of nodes in the final layer of the CRS is set to the number of categories N_c, then the model can be regarded as a classifier $\mathcal{C}\colon \{0,1\}^{N_b} \to \{0,1\}^{N_c}$. Although the CRS has good characteristics in terms of model expressiveness and transparency, its weights are discrete, making it impossible to train the model using gradient descent. Therefore, we adopt MLLP as a surrogate model to mimic the behavior of CRS. The number of layers and the nodes in each layer of MLLP correspond one-to-one with those in CRS. The difference lies in that MLLP is a fully connected neural network model.

Let $\hat{z}_i^l$ and $\hat{t}_i^l$ denote the neurons in MLLP corresponding to the CRS nodes z_i^l and t_i^l, respectively. $\widehat{W}^l$ represents the weight matrix of the l-th layer in MLLP, where $\widehat{W}_{i,j}^l \in [0,1]$. To enable MLLP to mimic the logical disjunction and conjunction operations in CRS, we adopt the continuous logic activation function proposed in [13], which is defined as shown in equation (4).

$$\begin{cases} \hat{z}_i^l = Conjunction(\hat{t}_i^{l-1}, \widehat{W}_i^l) = \prod_{j=1}^{N_n^{l-1}} 1 - \widehat{W}_{i,j}^l (1 - \hat{t}_{i,j}^{l-1}) \\ \hat{t}_i^l = Disjunction(\hat{z}_i^{l-1}, \widehat{W}_i^l) = 1 - \prod_{j=1}^{N_n^{l-1}} (1 - \hat{z}_{i,j}^{l-1} \cdot \widehat{W}_{i,j}^l) \end{cases} \tag{4}$$

When simulating disjunction and conjunction operations, it is necessary to constrain the output within the range [0,1]. Therefore, we apply the $Clip(\widehat{W}_{i,j}^l) = max(0, min(1, \widehat{W}_{i,j}^l))$ function to clip the weights. MLLP can exactly replicate the behavior of CRS when its weights are identical to those of the corresponding CRS. Let $\hat{\mathcal{C}}$ denote the MLLP model and $\widehat{\mathcal{W}}$ represent all weights of the model $\hat{\mathcal{C}}$. The loss function for training the MLLP is defined as shown in equation (5).

$$Loss = \frac{1}{N_s} \sum_{i=1}^{N_s} MSE(Y_i', \hat{\mathcal{C}}(X_i', \widehat{\mathcal{W}})) + \lambda \mathcal{E}(\widehat{\mathcal{W}}) \tag{5}$$

where $MSE(\cdot)$ is the mean squared error, and $\mathcal{E}(\widehat{W}))$ denotes the L_2 regularization term.

3.3 Neural Architecture Search

Both the discrete CRS and its continuous version, MLLP, feature a fixed model architecture with alternating layers of conjunction and disjunction operations. The number of layers and the nodes in each layer are manually preset, lacking the capability for automatic optimization or architecture search. As demonstrated in literature [28], different model architecture designs significantly impact predictive performance. Specifically, the configuration of hidden layer nodes and the depth of layers directly affect classification accuracy. To enable automatic discovery of high-performing and interpretable neural rule structures, we incorporate a neural architecture search framework based on reinforcement learning [36]. The goal is to find the optimal configuration including the number of layers and the number of nodes in each layer, so that the resulting MLLP achieves the best classification performance while maintaining interpretability. Moreover, the random binarization proposed in [28] is used to train a differentiable MLLP, which can then be transformed into a discrete CRS. Neural architecture search and random binarization are not mutually exclusive. Instead, their combination involves first identifying the optimal structure and then training the parameters based on that structure. This joint optimization mechanism of structure and parameters can effectively advance the automated design of interpretable models.

We employ a recurrent neural network as a controller, which sequentially generates architectural descriptions of candidate MLLP models. At each step, the controller predicts architectural hyperparameters. Each sampled architecture is then instantiated into a MLLP model, trained using the surrogate loss function defined in equation (5). After training, the validation accuracy of the instantiated model is used as the reward signal to update the controller parameters. Specifically, we optimize the expected reward using the REINFORCE proposed in [29]. To preserve the interpretability of the rule-based structure, we constrain the search space to ensure that all generated architectures strictly adhere to the alternating pattern of conjunction and disjunction layers. Additionally, to maintain the logical consistency of the model, we enforce the use of binary or clipped continuous weights, thereby enabling a faithful approximation of logical operations within a differentiable framework. Furthermore, we impose upper bounds on the depth and width of the network to prevent the construction of overly complex rule structure, which could undermine the model's transparency and comprehensibility. To address the high computational cost of evaluating each candidate architecture, we adopt a distributed training scheme where multiple candidate MLLP models are trained in parallel. Additionally, early stopping and parameter sharing strategies are incorporated to accelerate convergence during the search phase. After the search process converges, the top-k architectures with the highest validation performance are selected for further fine-tuning. The best-performing architecture is then extracted and interpreted as an optimized rule-based classifier, with logical expressions derived from its learned structure.

4 Experimental Study

4.1 Datasets

As shown in **Table 1**, we select 12 benchmark datasets with domain representativeness from the UCI Machine Learning Repository, which have formed a broad research consensus in terms of classification performance evaluation and model interpretability verification. To comprehensively test the generalization ability of the method, the selected datasets exhibit significant differences in three dimensions including sample size, class distribution, and feature space. The sample sizes range from 100 to 67,557, covering scenarios from small-sample learning to large-scale data analysis. The number of classes varies between 2 and 26, encompassing binary classification, multi-class classification, and high-dimensional classification tasks. The feature dimensions are distributed between 4 and 42, including numerical, categorical, and mixed-type features. This multi-dimensional data construction strategy effectively avoids potential biases in experimental evaluation.

Table 1. The datasets used in experimental study.

Dataset	Instances	Classes	Features
adult	32,561	2	14
bank-marketing	45,211	2	16
banknote	1,372	2	4
blogger	100	2	5
chess	28,056	18	6
connect-4	67,557	3	42
letRecog	20,000	26	16
magic04	19,020	2	10
mushroom	8,124	2	22
nursery	12,960	5	8
tic-tac-toe	958	2	9
wine	178	3	13

To address the issue of missing data, this study design a feature-type-sensitive differential processing pipeline. For continuous variables, a Gaussian assumption-based mean imputation method is employed, using the average value of observed samples for unbiased estimation. When significant skewness is detected, the system automatically switches to a median imputation mechanism to maintain the authenticity of the data distribution. For categorical variables, a maximum likelihood estimation-based mode imputation method is used. When multiple modes exist, a random sampling strategy is employed to select the imputation value to maintain the entropy of the original distribution. All imputation operations is completed independently on the training set, and a data isolation mechanism is strictly followed to prevent information leakage to the validation set.

4.2 Evaluation Metrics

To objectively evaluate model performance in class-imbalanced scenarios, this study constructed a multi-dimensional evaluation system. The macro F1-score served as the evaluation metric, calculated as the unweighted average of F1-scores across all classes to eliminate the impact of sample size differences. The experiment adopts a stratified 5-fold cross-validation method, ensuring that the class distribution in each fold strictly matched that of the original dataset. In each validation round, 80% of the data was used for model training, and 20% served as an independent validation set for hyperparameter tuning. The final experimental results were reported as the mean and standard deviation of the five-fold test set metrics, with statistical significance tests ensuring the reliability of conclusions.

4.3 Compared Methods

In this study, we select nine categories of benchmark models encompassing diverse modeling paradigms for systematic comparison, aiming to comprehensively validate the proposed method's capability to balance classification performance and model interpretability. The baseline models includes CRS [28], Decision tree (C4.5) [16], classification and regression trees (CART) [2], scalable bayesian rule lists (SBRL) [30], logistic regression (LR) [10], piecewise linear neural network (PLNN) [4], support vector machine (SVM) [5], gradient boosted decision trees (GBDT) [8], and two variants of random forest ($\text{RF}_{e=10}$ and $\text{RF}_{e=100}$) [1]. These baseline models exhibit gradient characteristics along the interpretability spectrum, establishing a multidimensional reference framework for evaluation.

Regarding interpretability dimensions, C4.5, CART, and SBRL serve as rule-based models that provide intrinsic interpretability through explicit decision paths or probabilistic rule sets. As representative generalized linear models, LR offers intuitive statistical interpretations through parameter weights. Such models are frequently adopted as benchmarks in interpretable machine learning research due to their transparent decision mechanisms. In contrast, PLNN employs piecewise linear activation functions to construct multilayer perceptrons. While demonstrating superior nonlinear modeling capabilities, its implicit distributed representations render decision logic inherently opaque. Ensemble methods including GBDT and RF achieve exceptional classification accuracy, yet their inherent model complexity and black-box nature significantly limit trustworthiness in high-stakes decision-making scenarios such as medical diagnosis and financial risk assessment.

The experimental design specifically examines models' differential performance in the accuracy-interpretability trade-off. By controlling the number of estimators in random forests (10 vs. 100), this study further elucidates the impact mechanism of model complexity on interpretability. Systematic comparisons not only validate the proposed method's capability to maintain classification performance comparable to black-box models while achieving decision process transparency, but also demonstrate its generalization potential across diverse task scenarios through cross-dataset robustness analysis.

4.4 Experimental Results

As shown in **Table 2**, the proposed NRL-NAS model demonstrates competitive classification performance across 12 benchmark datasets, achieving a mean macro F1-score of 87.23%. This represents a 0.25 percentage point improvement over the CRS baseline (86.98%) and significantly outperforms traditional decision tree models (C4.5: 84.48%, CART: 84.44%) and shallow learning methods (SVM: 75.56%, LR: 74.27%) with statistical significance ($p < 0.05$, Wilcoxon signed-rank test). Notably, NRL-NAS exhibits systematic advantages on datasets containing high-dimensional heterogeneous features, including *adult* (81.65% vs. 80.95%), *bank-marketing* (74.76% vs. 73.34%), and *wine* (98.87% vs. 98.10%). The perfect classification accuracy (100%) achieved on the *mushroom* dataset by both NRL-NAS and six baseline models confirms the theoretical soundness of our method in linearly separable scenarios. Compared to complex ensemble models, NRL-NAS achieves superior performance while maintaining interpretability. For instance, the random forest ensemble requires 100 base estimators (RF$_{e=100}$) to reach 86.82% mean performance, whereas NRL-NAS attains 87.23% F1-score through dynamic rule generation via neural architecture search. This efficiency makes NRL-NAS particularly suitable for domains requiring model transparency, such as medical diagnosis and financial risk assessment. Despite overall strong performance, statistically significant gaps exist on the *nursery* (97.17% vs. CRS 99.69%, ($p = 0.032$)) and *chess* (78.63% vs. CRS 80.21%, ($p = 0.041$)) datasets. We attribute these limitations to the multi-objective optimization trade-offs in dynamic rule generation: when handling strongly coupled features (e.g., chess move patterns), the neural architecture search may overfit local feature interactions, thereby reducing global rule generalizability. Furthermore, the inferior performance on *letRecog* (83.86% vs. C4.5 88.20% and RF$_{e=100}$ 90.31%) suggests potential quantization error accumulation in the CACC discretization algorithm when processing continuous handwritten features.

Table 2. Comparative results of classification performance between NRL-NAS and other methods.

Dataset	NRL-NAS	CRS	C4.5	CART	SBRL	LR	SVM	PLNN	GBDT	RF$_{e=10}$	RF$_{e=100}$
adult	**81.65**	80.95	75.40	74.77	79.88	78.43	63.63	73.55	80.36	77.48	78.83
bank-marketing	**74.76**	73.34	71.24	70.21	72.67	69.81	66.78	72.40	75.28	69.89	72.01
banknote	96.27	94.93	98.45	97.85	94.44	98.82	**100.00**	**100.00**	99.48	99.11	99.19
blogger	83.61	**85.33**	75.90	78.27	67.64	55.55	62.11	56.24	67.58	77.33	85.17
chess	78.63	**80.21**	79.90	79.15	26.44	33.06	36.83	77.85	71.41	66.38	74.25
connect-4	69.45	65.88	61.66	61.24	48.54	49.87	50.17	64.55	71.41	66.38	**75.35**
letRecog	83.86	84.96	**88.20**	86.90	64.32	72.05	74.90	80.92	87.32	88.43	90.31
magic04	83.53	80.87	80.31	80.05	82.52	75.72	75.64	83.07	**96.51**	93.61	96.15
mushroom	**100.00**	**100.00**	**100.00**	99.98	**100.00**	**100.00**	**100.00**	**100.00**	**100.00**	**100.00**	**100.00**
nursery	97.17	**99.69**	95.55	95.47	71.32	64.64	82.48	79.71	87.32	88.43	90.31
tic-tac-toe	**99.79**	99.77	91.70	94.21	98.39	98.12	98.07	98.26	99.19	94.85	98.37
wine	**98.87**	98.10	95.48	94.39	95.84	95.16	96.05	76.07	98.44	96.90	98.31
Average	**87.23**	86.98	84.48	84.44	75.17	74.27	75.56	81.17	85.56	84.24	86.82

In this study, interpretable rule sets were extracted from datasets in the form of logical expressions. **Table 3** presents the concise and complete rule sets generated for three representative classification tasks: *adult* (mixed-type data), *blogger* (categorical data), and wine (numerical data). Taking the adult income prediction task as an example, the objective is to determine whether an individual's annual income exceeds USD 50K. Rule 1, "capital-gain > 7070.357," indicates that individuals with capital gains exceeding approximately USD 7070 are highly likely to have an income above USD 50K. This single-condition rule highlights a strong positive correlation between high capital gains and high income levels. Rule 2 characterizes the typical attributes of high-income individuals, suggesting that young, well-educated employees occupying managerial positions within the private sector and working extended hours are more likely to belong to the high-income group. Rule 3 predicts high income based solely on the marital status of being married to a U.S. citizen, potentially reflecting the real-world observation that married individuals tend to exhibit greater economic stability. Collectively, these rule sets integrate key features such as occupation type, educational attainment, and economic status, thereby providing verifiable and interpretable reasoning pathways to support decision-making processes.

The performance of the model under varying data complexities was further validated through a systematic analysis of rule lengths extracted from 12 benchmark datasets. As summarized in **Table 4**, the model demonstrates the ability to generate highly concise rule sets for low-dimensional datasets, with the average rule lengths for magic04 and tic-tac-toe being 2.11 and 2.43, respectively. Conversely, for high-dimensional and complex datasets, rule complexity exhibits a positive correlation with data dimensionality; for example, the average rule length for connect-4 reaches 10.87. Similar trends are observed in real-world datasets such as bank-marketing and banknote, underscoring the model's capacity to preserve rule simplicity while maintaining robust classification

Table 3. Explainable rules generated by NRL-NAS with different datasets.

Dataset	Class	Rules
adult	<=50K	(relationship:Not-in-family ∧ capital-gain <= 5161.948) ∨ (relationship:Unmarried ∧ capital-loss <= 1820.5) ∨ (occupation:Other-service ∧ capital-gain <= 5161.948) ∨ (workclass:Self-emp-not-inc ∧ occupation:Farming-fishing ∧ capital-loss <= 1820.5) ∨ (marital-status:Never-married ∧ capital-gain <= 5161.948 ∧ capital-loss <= 1820.5) ∨ (education-num <= 12.5 ∧ capital-gain <= 5161.948 ∧ capital-loss > 1978.5) ∨ (age <= 29.5 ∧ capital-gain <= 5161.948) ∨ (capital-gain <= 5161.948 ∧ capital-loss <= 1820.5 ∧ hours-per-week <= 34.5) ∨ (workclass:Self-emp-not-inc ∧ capital-gain <= 5161.948 ∧ capital-loss <= 1820.5)
	>50K	(capital-gain > 7070.357) ∨ (marital-status:Married-civ-spouse) ∨ (workclass:Private ∧ education:Bachelors ∧ occupation:Exec-managerial ∧ age:(29.5,35.5] ∧ fnlwgt:(91042.551,208055.795) ∧ education-num:(12.5,13.5] ∧ hours-per-week > 41.5)
blogger	pb_no	(Degree:medium ∧ caprice:right ∧ lpss:yes) ∨ (caprice:middle ∧ topic:impression ∧ lmt:no lpss:yes) ∨ (Degree:low ∧ topic:impression ∧ lmt:yes) ∨ (Degree:medium ∧ topic:scientific ∧ lpss:yes) ∨ (Degree:low ∧ caprice:right ∧ lmt:no ∧ lpss:yes)
	pb_yes	(Degree_medium ∧ caprice:middle ∧ lmt:yes) ∨ (caprice:right ∧ topic:political ∧ lpss:yes) ∨ (Degree_medium ∧ topic:political ∧ lpss:yes) ∨ (Degree:medium ∧ caprice:left ∧ topic:impression) ∨ (Degree:medium ∧ lpss:no) ∨ (lmt:no ∧ lpss:no) ∨ (caprice:left ∧ topic:news)
wine	Class_1	(Alcalinity_of_ash <= 17.9 ∧ Magnesium:(99.5, 133.0] ∧ Total_phenols > 2.335 ∧ Flavanoids > 2.276) ∨ (Alcohol:(12.919, 13.535] ∧ Magnesium:(99.5, 133.0] ∧ Total_phenols > 2.335 ∧ Flavanoids > 2.2762 ∧ Color_intensity:(3.491, 4.968]) ∨ (Proline > 905.163)
	Class_2	(Color_intensity <= 3.491) ∨ (Malic_acid <= 1.546 ∧ Color_intensity:(3.491, 4.968] ∧ Hue:(1.005, 1.295]) ∨ (Alcohol <= 12.729 ∧ Total_phenols > 2.335 ∧ Proanthocyanins > 1.651 ∧ Proline <= 511.479) ∨ (Malic_acid <= 1.546 ∧ Magnesium <= 88.5 ∧ Proline <= 511.479)
	Class_3	(Hue <= 0.785 ∧ OD280/OD315_of_diluted_wines <= 2.155 ∧ Proline:(511.479, 768.010]) ∨ (Malic_acid > 2.224 ∧ Alcalinity_of_ash > 19.45 ∧ OD280/OD315_of_diluted_wines <= 2.155) ∨ (Flavanoids <= 0.913 ∧ Hue <= 0.785)

performance. Further analysis of the distribution of rule lengths reveals that atomic core rules, typically comprising one to two conditions, are consistently generated across all datasets, whereas the maximum rule lengths vary significantly. For instance, maximum rule lengths of 19 and 20 are observed in the high-dimensional datasets connect-4 and mushroom, respectively, markedly exceeding those of lower-dimensional datasets. Of particular note is the blogger dataset, which exhibits a broad distribution of rule lengths. This, combined with the model's dynamic adjustment mechanism, confirms its ability to adaptively balance rule complexity and classification accuracy based on the strength of feature interactions, thereby demonstrating robust adaptability to diverse data patterns.

Table 4. The length of rules generated by NRL-NAS with various datasets.

Dataset	Rule length		
	Min	Max	Average
adult	1	9	5.36
bank-marketing	1	8	2.87
banknote	2	6	2.91
blogger	2	13	3.67
chess	1	15	6.64
connect-4	1	19	10.87
letRecog	1	9	5.05
magic04	1	17	2.11
mushroom	2	20	5.33
nursery	2	9	3.46
tic-tac-toe	1	6	2.43
wine	1	5	3.78

5 Conclusion and Future Work

In this study, we propose an interpretable classification framework that integrates neural rule learning with network architecture search. The proposed method is capable of automatically learning logical rules to represent and classify structured data while adaptively determining the optimal network architecture to maximize predictive performance. Extensive experiments on multiple benchmark datasets demonstrate that our approach achieves a strong balance between predictive accuracy and model interpretability, generating concise and highly interpretable logical rules without sacrificing classification performance. As future work, we intend to extend the proposed framework to unstructured data domains and explore more advanced strategies for fully automating rule extraction, thereby further reducing the dependence on manual feature engineering.

Acknowledgments. This work is supported by the National Natural Science Foundation of China (Grant No. 62476046), the Liaoning Provincial Joint Science and Technology Program (Grants No. 2024JH2/102600111 and 2024JH2/102600108), the Key Laboratory of Advanced Control

System of Industrial Equipment of Liaoning Province (Grant No. LACS202101), the Research Project of Liaoning Provincial Education Department (Grant No. JYTMS20231803), and the Fundamental Research Funds for the Central Universities (Grant No. 04442024051).

References

1. Breiman, L.: Random forests. Machine Learning 45(1), 5–32 (2001)
2. Breiman, L., Friedman, J.H., Olshen, R.A., Stone, C.J.: Classification and Regression Trees. Wadsworth International Group (1984)
3. Černevičienė, J., Kabašinskas, A.: Explainable artificial intelligence (xai) in finance: A systematic literature review. Artificial Intelligence Review 57(8), 216 (2024)
4. Chu, L., Hu, X., Hu, J., Wang, L., Pei, J.: Exact and consistent interpretation for piecewise linear neural networks: A closed form solution. In: Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining. pp. 1244–1253 (2018)
5. Cortes, C., Vapnik, V.: Support-vector networks. Machine Learning 20(3), 273–297 (1995)
6. Deeks, A.: The judicial demand for explainable artificial intelligence. Columbia Law Review 119(7), 1829–1850 (2019)
7. Feng, J., Yu, Y., Zhou, Z.H.: Multi-layered gradient boosting decision trees. In: Bengio, S., Wallach, H., Larochelle, H., Grauman, K., Cesa-Bianchi, N., Garnett, R. (eds.) Advances in Neural Information Processing Systems. vol. 31. Curran Associates, Inc. (2018)
8. Friedman, J.H.: Greedy function approximation: A gradient boosting machine. Annals of Statistics 29(5), 1189–1232 (2001)
9. Gunning, D., Stefik, M., Choi, J., Miller, T., Stumpf, S., Yang, G.Z.: Xai-explainable artificial intelligence. Science Robotics 4(37), eaay7120 (2019)
10. Kleinbaum, D.G., Dietz, K., Gail, M., Klein, M., Klein, M.: Logistic regression. Springer (2002)
11. LeCun, Y., Bengio, Y., Hinton, G.: Deep learning. Nature 521(7553), 436–444 (2015)
12. Liu, H., Simonyan, K., Yang, Y.: Darts: Differentiable architecture search. arXiv preprint arXiv:1806.09055 (2018)
13. Payani, A., Fekri, F.: Learning algorithms via neural logic networks. arXiv preprint arXiv:1904.01554 (2019)
14. Pham, H., Guan, M., Zoph, B., Le, Q., Dean, J.: Efficient neural architecture search via parameters sharing. In: International Conference on Machine Learning. pp. 4095–4104. PMLR (2018)
15. Qiao, L., Wang, W., Lin, B.: Learning accurate and interpretable decision rule sets from neural networks. In: Proceedings of the AAAI Conference on Artificial Intelligence. vol. 35, pp. 4303–4311 (2021)
16. Quinlan, J.R.: C4.5: Programs for Machine Learning. Morgan Kaufmann (1993)
17. Real, E., Liang, C., So, D., Le, Q.: Automl-zero: Evolving machine learning algorithms from scratch. In: International Conference on Machine Learning. pp. 8007–8019. PMLR (2020)
18. Ren, P., Xiao, Y., Chang, X., Huang, P.Y., Li, Z., Chen, X., Wang, X.: A comprehensive survey of neural architecture search: Challenges and solutions. ACM Computing Surveys (CSUR) 54(4), 1–34 (2021)
19. Ribeiro, M.T., Singh, S., Guestrin, C.: 'why should I trust you?' explaining the predictions of any classifier. In: Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. pp. 1135–1144 (2016)
20. Rudin, C.: Stop explaining black box machine learning models for high stakes decisions and use interpretable models instead. Nature Machine Intelligence 1(5), 206–215 (2019)

21. Selvaraju, R.R., Cogswell, M., Das, A., Vedantam, R., Parikh, D., Batra, D.: Gradcam: visual explanations from deep networks via gradient-based localization. International Journal of Computer Vision 128, 336–359 (2020)
22. Shi, S., Xie, Y., Wang, Z., Ding, B., Li, Y., Zhang, M.: Explainable neural rule learning. In: Proceedings of the ACM Web Conference 2022. pp. 3031–3041 (2022)
23. Tjoa, E., Guan, C.: A survey on explainable artificial intelligence (xai): Toward medical xai. IEEE Transactions on Neural Networks and Learning Systems 32(11), 4793–4813 (2020)
24. Tsai, C.J., Lee, C.I., Yang, W.P.: A discretization algorithm based on class-attribute contingency coefficient. Information Sciences 178(3), 714–731 (2008)
25. Wang, L.X.: Fast training algorithms for deep convolutional fuzzy systems with application to stock index prediction. IEEE Transactions on Fuzzy Systems 28(7), 1301–1314 (2020)
26. Wang, Y., Liu, H., Jia, W., Guan, S., Liu, X., Duan, X.: Deep fuzzy rule-based classification system with improved wang–mendel method. IEEE Transactions on Fuzzy Systems 30(8), 2957–2970 (2022)
27. Wang, Z., Zhang, W., Liu, N., Wang, J.: Scalable rule-based representation learning for interpretable classification. Advances in Neural Information Processing Systems 34, 30479–30491 (2021)
28. Wang, Z., Zhang, W., Wang, J., et al.: Transparent classification with multilayer logical perceptrons and random binarization. In: Proceedings of the AAAI conference on artificial intelligence. vol. 34, pp. 6331–6339 (2020)
29. Williams, R.J.: Simple statistical gradient-following algorithms for connectionist reinforcement learning. Machine learning 8, 229–256 (1992)
30. Yang, H., Rudin, C., Seltzer, M.: Scalable bayesian rule lists. In: International conference on machine learning. pp. 3921–3930. PMLR (2017)
31. Yu, L., Li, M., Zhang, Y.L., Li, L., Zhou, J.: Finrule: Feature interactive neural rule learning. In: Proceedings of the 32nd ACM International Conference on Information and Knowledge Management. pp. 3020–3029 (2023)
32. Zhang, Q., Yang, Y., Ma, H., Wu, Y.N.: Interpreting cnns via decision trees. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 6261–6270 (2019)
33. Zhang, Y., Ishibuchi, H., Wang, S.: Deep takagi–sugeno–kang fuzzy classifier with shared linguistic fuzzy rules. IEEE Transactions on Fuzzy Systems 26(3), 1535–1549 (2017)
34. Zhang, Y., Wang, G., Zhou, T., Huang, X., Lam, S., Sheng, J., Choi, K.S., Cai, J., Ding, W.: Takagi-sugeno-kang fuzzy system fusion: A survey at hierarchical, wide and stacked levels. Information Fusion 101, 101977 (2024)
35. Zhou, Z.H., Feng, J.: Deep forest. National Science Review 6(1), 74–86 (2019)
36. Zoph, B., Le, Q.V.: Neural architecture search with reinforcement learning. arXiv preprint arXiv:1611.01578 (2016)