

A Rule-Guided Neural Network for Explainable Enterprise Architecture Decision Support

Rahul Ramesh P
dept. Artificial Intelligence and Data
Science SRM University, Kattankalthur
mail.rahul.pragada@gmail.com

A. Shanthini
Professor Department of Data Science and
Business Systems SRM Institute of Science
and Technology
shanthia@srmist.edu.in

Abstract - Enterprise Architecture (EA) decision-making is not a simple task. It involves dealing with multiple organizational factors such as business objectives, technology constraints, regulatory requirements, and scalability needs. In most cases, EA has traditionally relied on expert-driven frameworks and rule-based governance models. These approaches are quite useful because they provide clarity and traceability, but at the same time, they tend to be rigid and are not always able to keep up with fast-changing enterprise environments.

In recent years, machine learning-based decision support systems have come into the picture, offering a more data-driven way to model complex enterprise relationships. While these systems give good predictive performance, they often behave like black-box models, making it difficult to clearly understand how decisions are being made. In an EA context, this becomes a concern, because explainability, governance alignment, and accountability are very important.

Because of this, we can clearly see a gap between traditional rule-based approaches, which are easy to interpret but not flexible, and modern data-driven models, which are flexible but lack transparency. To address this problem, this work proposes a neuro-symbolic approach that combines symbolic architectural rules with neural representation learning. Here, domain-specific constraints are encoded as rule-based features and combined with enterprise attributes in a shared latent representation, learned using a multi-head neural network.

The approach is evaluated using a synthetic dataset of 3,000 enterprise scenarios. The results show that the neuro-symbolic model achieves an accuracy of 98.29%, compared to 92.12% for a neural-only baseline. More importantly, it supports decision-making that is not only accurate but also explainable and aligned with enterprise governance expectations.

Keywords - Enterprise Architecture Decision-Making, Neuro-Symbolic AI, Decision Support Systems

I. INTRODUCTION

Enterprise Architecture (EA) gives a methodical way to align an organization's business goals with its information systems and technology setup. By organizing capabilities across business, data, application, and technology layers, EA helps in better coordination of planning, smoother system evolution, and governance-driven decision-making. Common frameworks, especially those from organizations like The Open Group, focus on proper documentation, traceability, and maintaining architectural consistency to support enterprise transformation.

At the same time, enterprise environments today are becoming more and more complex. With the growing adoption of cloud-native architectures, distributed systems, and continuously changing regulatory requirements, architectural decision-making is not that straightforward anymore. Decisions like choosing deployment strategies, integration patterns, or data management approaches have to consider multiple dependent factors such as scalability, security, compliance.

Traditional EA methodologies mostly depend on expert-driven rules and evaluation frameworks. These approaches are good in terms of governance transparency and clarity, but in practice they tend to be quite static and not easy to adapt when enterprise environments keep changing quickly. On the other hand, machine learning-based decision support systems offer more flexibility and can model complex relationships effectively. However, they often behave like black-box models, making it difficult to understand how decisions are actually made. This becomes a concern, especially in governance-critical (finance regulatory etc.) Architectural decisions where interpretability is very important.

To address this challenge, this work proposes a neuro-symbolic decision support framework that combines symbolic architectural rules with neural representation learning. The idea is to bring together the strengths of both approaches. Rule-based reasoning helps maintain transparency and alignment with governance, while neural models add flexibility and adaptability. Using

this approach, the objective is to provide enterprise architecture recommendations that are not only adaptive but also explainable and practical in real-world scenarios.

II. RELATED WORK

Enterprise Architecture (EA) has slowly evolved as a structured discipline to manage organizational complexity through layered abstractions, governance mechanisms, and systematic planning. Early work by Winter and Fischer defined EA as a hierarchical, multi-layered system integrates business, process, and technology artefacts to maintain alignment and consistency [1]. Their model focused on structural dependencies and followed the principle that “IT supports business,” establishing EA as a governance-driven, multi-level framework.

Building on this, Buckl et al. introduced an information model for managing application landscape evolution, highlighting the importance of temporality, traceability, and historization in EA [2]. In simple terms, their work emphasized that architectural decisions should be traceable across time, so that accountability is maintained through different planning cycles. While these approaches brought strong structure and clarity, they were mostly based on deterministic models and required considerable manual effort.

To improve architectural decision-making, Kazman et al. proposed the Architecture Tradeoff Analysis Method (ATAM), a scenario-based framework used to evaluate decisions against quality attributes such as performance, availability, and modifiability [3]. ATAM made architectural reasoning more systematic by identifying risks and trade-offs through stakeholder-driven analysis. However, in practice, it still depends heavily on expert input and manual evaluation, which limits its scalability in rapidly changing environments.

With the rise of data-driven approaches, machine learning (ML) has started influencing decision support systems. Recent work on Explainable Artificial Intelligence (XAI)-based DSS highlights the increasing use of deep learning models in enterprise decision contexts [4]. These models offer strong predictive capabilities but introduce a trade-off between accuracy and explainability. As noted in studies on supervised ML explainability, neural networks often behave like “black boxes,” making their decisions difficult to interpret [5]. This lack of transparency is a major concern in EA, where auditability, governance alignment, and accountability are critical.

To address this shortcoming, neuro-symbolic AI has emerged as a hybrid approach that combines neural learning with symbolic reasoning. Nawaz et al. describe this paradigm as the integration of data-driven learning with logic-based inference, allowing both adaptability

and structured reasoning [6]. Techniques such as Logic Tensor Networks and differentiable logic programming are examples where symbolic constraints are embedded within neural architectures.

In enterprise and software decision-making contexts, recent studies show that integrating rule-based knowledge directly into neural models can improve interpretability and data efficiency [7]. Similar hybrid approaches have also been explored in ethical and risk-sensitive domains, where transparency and compliance are essential [8]. These works indicate that embedding domain knowledge into learning systems enhances both robustness and trust.

Further, research on hybrid neural models for complex environments emphasizes the need for systems that are interpretable by design, rather than relying only on post-hoc explanations [9]. Such approaches aim to combine human-understandable variables with adaptive learning mechanisms, ensuring both clarity and reliability in decision-making.

Despite these developments, the application of neuro-symbolic methods to enterprise architecture decision support is still limited. Traditional EA approaches focus on structured documentation and trade-off reasoning [1]–[3], while AI-based decision support systems focus more on predictive performance and post-hoc explainability [4], [5]. Although neuro-symbolic AI shows strong potential [6]–[9], it has not yet been fully adapted to complex EA decision scenarios involving governance constraints, architectural styles, deployment strategies, and compliance requirements.

This gap motivates the present research.

III. METHODOLOGY

Architecture diagram

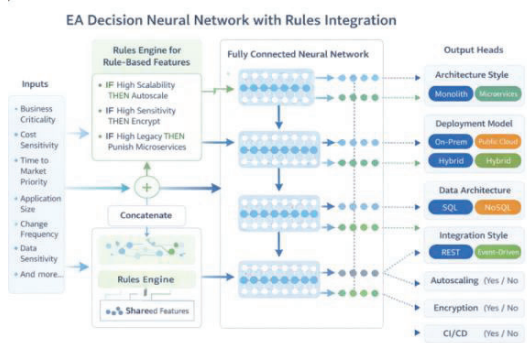


Fig. 1. Neuro-Symbolic Architecture.

The proposed neuro-symbolic decision support framework (Fig.1) Consists of three main components: a rule-engine layer, a feature encoding module, and a multi-head feedforward neural network. Let the

enterprise scenario be represented as an input feature vector.

$$X = \{x_1, x_2, \dots, x_n\}, X \in R^n \quad (1)$$

Where $n = 20$ denotes the number of enterprise attributes.

A symbolic rule set $R = \{r_1, r_2, \dots, r_m\}$ is defined using domain-informed architectural constraints. Each rule r_j evaluated against the input vector X , producing a binary or probabilistic activation value:

$$\phi_{j(X)} \in [0,1] \quad (2)$$

The rule activations are aggregated into a rule feature vector:

$$\Phi(X) = \{\phi_{-1}(X), \phi_{-2}(X), \dots, \phi_{-m}(X)\} \quad (3)$$

The original feature vector and rule-derived features are concatenated to form an augmented representation:

$$X' = [X \parallel \Phi(X)] \quad (4)$$

Where \parallel denotes vector concatenation.

This augmented representation is passed through a shared neural representation layer:

$$H = f_{\theta}(X') \quad (5)$$

Where f_{θ} denotes a feedforward neural transformation parameterized by weights θ . The latent representation H captures both symbolic constraints and learned enterprise patterns.

The model employs a multi-head output architecture to predict seven coordinated architectural decisions:

$$Y = \{y_1, y_2, y_3, \dots, y_7\} \quad (6)$$

Each output head applies a softmax activation to produce class probabilities:

$$\hat{y}_k = \text{softmax}(W_k H + b_k) \quad (7)$$

For $k = 1, 2, 3, 4, 5, 6, 7$.

The network is trained using categorical cross-entropy loss aggregated across all output heads:

$$L = \sum_{k=1}^7 L_k \quad (9)$$

This formulation allows the model to learn across multiple architectural dimensions at the same time, while still maintaining interpretability through explicit integration of rule-based features. In other words, even

though the model is learning in a data-driven way, it does not lose alignment with predefined architectural rules.

Since there are no publicly available datasets specifically for enterprise architecture (EA) decision-making, a structured synthetic dataset was created to simulate realistic enterprise design scenarios. The dataset generation process was guided by established EA principles, including layered enterprise architecture modelling [1], managed application landscape structures [2], and scenario-based evaluation approaches such as ATAM [3]. These references helped in defining domain-relevant constraints and trade-off patterns, which were then encoded into the dataset to ensure architectural consistency and governance alignment.

The final dataset consists of 3,000 enterprise scenarios, with each scenario represented as a 20-dimensional feature vector covering four major categories **Business attributes** like organizational scale, domain criticality, growth rate, **Technical attributes** like legacy dependency, scalability requirements, performance sensitivity, **Security and compliance factors** like regulatory constraints, data sensitivity levels finally **Organizational maturity indicators** like DevOps capability, governance maturity

Each feature was encoded as categorical or ordinal variables and normalized where appropriate.

To avoid deterministic rule replication, probabilistic perturbations were introduced during label assignment. Specifically, rule-based architectural recommendations were treated as primary decision anchors, and controlled stochastic variations were applied to simulate real-world ambiguity and expert disagreement. This ensured that the dataset preserved domain consistency while enabling learning generalization.

Each scenario was labelled with seven outputs:

- Architecture style
- Deployment model
- Data architecture pattern
- Integration strategy
- Auto Scale
- Encryption
- CI/CD

Class distributions were kept in check to reduce extreme imbalance and to ensure meaningful training and evaluation across all output dimensions.

IV. RESULT & DISCUSSION

An 80–20 train–test split was applied, resulting in 2,400 training samples and 600 testing samples. All categorical features were encoded using one-hot Data collection encoding.

In the neuro-symbolic configuration, additional rule-derived features were generated using predefined architectural constraints and appended to the input feature space. The model architecture consists of a shared feedforward neural network with two hidden layers (128 and 64 neurons respectively), followed by seven parallel decision heads corresponding to Architecture style, Deployment model, Data architecture, Integration style

Training Behaviour

Training was carried out for purely neural network and neuro-symbolic models, (Fig. 2) captures behaviour indicating both models improve over time while Neuro-symbolic model demonstrates faster convergence with notable progress in performance gap.

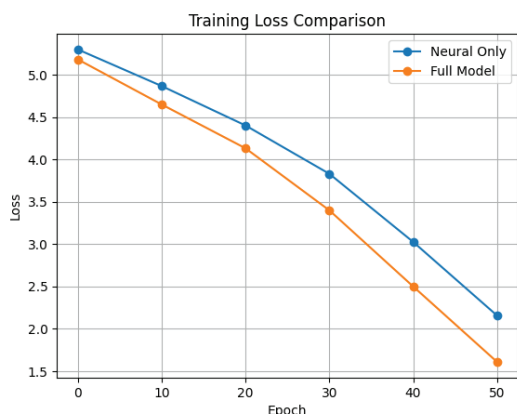


Fig. 2. Training Loss Comparison.

(Table.1) Indicates improved handling of imbalanced architectural patterns while Deployment remains the most complex decision Neuro-symbolic approach improves constraint –driven reasoning

Component	Neural Only	Full Model
Architecture	0.8633	1.0000
Deployment	0.8217	0.9317
Data	1.0000	1.0000
Integration	1.0000	1.0000
Auto Scale	1.0000	1.0000
Encryption	1.0000	1.0000
CI/CD	1.0000	1.0000
Average	0.955	0.990

Table 1. Accuracy comparison

For **Architecture** and **Deployment**, accuracy is strong, but the lower recall and F1 scores (Table 2.) suggest the model misses some relevant cases despite high precision. Overall, the model is highly precise but slightly less balanced in identifying all relevant instances for certain components.

Component	Accuracy	Precision (Macro)	Recall (Macro)	F1 Score (Macro)
Architecture	0.8633	0.93	0.60	0.63
Deployment	0.8217	0.91	0.71	0.69
Data	1.0000	1.00	1.00	1.00
Integration	1.0000	1.00	1.00	1.00
Auto Scale	1.0000	1.00	1.00	1.00
Encryption	1.0000	1.00	1.00	1.00
CI/CD	1.0000	1.00	1.00	1.00

Table 2. Neuro-Symbolic Model Metrics

Explainability

Explainability is achieved through rule-derived features that integrate with neural learning process. These features explain domain knowledge such as:

- Regulatory restrictions
- Change frequency impact
- Legacy dependency constraints
- Cloud readiness considerations

Because the rule activations are deterministic and easy to understand, each architectural recommendation can be traced back to specific conditions. So, it becomes clearer why a particular decision is taken. Compared to purely neural models, this makes the system more transparent, while still keeping the learning going.

At the same time, purely neural models are good at picking patterns, but they often get biased towards the dominant classes, especially in complex architectural and deployment cases. When rule-based features are added, this problem reduces to some extent. We can see this from better minority class recall and improved macro F1-scores, which means the decisions are more balanced overall.

In the end, the neuro-symbolic approach manages to include architectural knowledge into the learning process without making the system too rigid. It keeps a good balance — not fully rule-based, not fully black-box — which makes it more practical for real-world use.

Limitations and Future Work

Even though the proposed neuro-symbolic framework shows strong performance, there are a few limitations that need to be considered.

First, the dataset used in this work is synthetically generated using deterministic rule mappings. This helps maintain consistency, but at the same time, it may not fully reflect the complexity and unpredictability we usually see in real enterprise environments. Second, some of the decision outputs, like data architecture, integration style, and CI/CD configuration, show very high accuracy. This is mainly because the labels are simplified and follow clear deterministic patterns, so it doesn't fully test how well the model generalizes in more complex cases.

Third, the symbolic rules used in the model are manually defined based on domain knowledge. While this helps with interpretability and keeps things aligned with governance, it may not cover all the real-world variations and evolving best practices in enterprise architecture. Also, the current framework does not consider time-based changes, such as system evolution, migrations, or continuous transformations that happen in real organizations.

To improve on these points, future work will focus on making the framework more practical and closer to real-world scenarios. One important step is to test the model using actual enterprise architecture case studies to better understand its real applicability. In addition, introducing dynamic policies and allowing rules to adapt over time can help the system stay relevant as enterprise needs change.

Further improvements can include using graph-based representations to better capture relationships between

different architectural components. Also, attention-based neural models can be explored to improve both interpretability and context understanding. Another useful direction is to automatically extract rules from past architectural decisions, so that the system can keep learning and updating its knowledge base.

Finally, extending the framework to support multi-cloud and distributed environments will make it more suitable for modern enterprise setups, where systems are often spread across different platforms and locations.

V. CONCLUSION

This paper presented a neuro-symbolic decision support framework for enterprise architecture design. By combining rule-based knowledge with multi-head neural learning, the system is able to improve prediction performance in complex architectural and deployment decisions.

From the experimental results, it can be seen that the neuro-symbolic model performs better than the neural-only baseline, achieving an overall accuracy of 97.76%. At the same time, it also improves minority class recall and macro F1-scores, especially in scenarios where constraints play an important role.

Overall, the results show that bringing together structured domain knowledge and data-driven learning works well in enterprise architecture settings. The approach gives a good starting point for building AI-based systems that are not only accurate, but also explainable and aligned with architectural constraints and governance needs.

VI. REFERENCES

- [1] R. Winter and R. Fischer, "Essential layers, artifacts, and dependencies of enterprise architecture," *Journal of Enterprise Architecture*, May 2007.
- [2] S. Buckl, A. M. Ernst, F. Matthes, and C. M. Schweda, "An information model for managed application landscape evolution," *Journal of Enterprise Architecture*, Feb. 2009.
- [3] R. Kazman, M. Klein, and P. Clements, *ATAM: Method for Architecture Evaluation*, CMU/SEI-2000-TR-004, Software Engineering Institute, Carnegie Mellon University, Aug. 2000.
- [4] G. Kostopoulos, G. Davrazos, and S. Kotsiantis, "Explainable artificial intelligence-based decision support systems: A recent review," *Electronics*, vol. 13, no. 14, p. 2842, 2024. doi: 10.3390/electronics13142842.
- [5] N. Burkart and M. Huber, "A survey on the explainability of supervised machine learning," *Journal of Artificial Intelligence Research*, vol. 70, pp. 245–317, 2021.
- [6] . Nawaz, M. Anees-ur-Rahaman, and Z. Saeed, "A review of neuro-symbolic AI integrating reasoning and learning for advanced cognitive systems," *Intelligent Systems with Applications*, vol. 26, 2025.

[7] N. S. K. Sastry and V. Sastry, "Neuro-symbolic artificial intelligence in software systems: Integrating neural learning and symbolic reasoning for explainable decision-making," *International Journal of Managerial Studies and Research*, vol. 14, no. 1, pp. 10–17, Jan. 2026.

[8] C. K. Kolli, "Hybrid neuro-symbolic models for ethical AI in risk-sensitive domains," arXiv:2511.17644, 2025.

[9] R. D'Elia, A. Termine, and F. Flammini, "Towards explainable decision support using hybrid neural models for logistic terminal automation," arXiv:2509.07577, 2025.