

Autonomous Self-Evolving Automl Framework using Reinforcement-Guided Neural Architecture Search and Concept Drift Adaptation

SHUBHAYU BHATTACHARYYA

Department of Computer Science and Application Institute of Engineering and Management, Kolkata
University of Engineering and Management, Kolkata
Kolkata, West Bengal, India shubhayubhattacharyya37@gmail.com

Abstract—Artificial intelligence systems operating in real-world environments must adapt to continuously evolving data distributions. Traditional machine learning pipelines require manual architecture design and hyperparameter tuning, which limits scalability and adaptability. This paper proposes an autonomous self-evolving AutoML framework that integrates neural architecture search, reinforcement-guided evolution, transformer-based predictive modeling, concept drift detection, and self-healing mechanisms.

The proposed system maintains a population of neural architectures and continuously evaluates them on streaming data. A reinforcement controller guides architecture mutation and model selection, enabling the system to automatically discover improved models over time. Concept drift detection mechanisms monitor changes in the data distribution and trigger adaptive retraining when significant drift occurs.

Experimental evaluation using simulated system monitoring data demonstrates that the framework successfully maintains predictive accuracy while autonomously evolving model architectures. The results show improved performance across multiple evaluation metrics including mean squared error, R squared score, precision, recall, F1 score, and accuracy.

The proposed framework represents a step toward autonomous artificial intelligence systems capable of continuous self-optimization and adaptive learning.

Index Terms—AutoML, Neural Architecture Search, Reinforcement Learning, Concept Drift Detection, Transformer Networks, Evolutionary Machine Learning

I. INTRODUCTION

Recent advances in artificial intelligence have enabled machine learning models to achieve remarkable performance across domains such as healthcare, financial prediction, and intelligent infrastructure systems. However, most machine learning pipelines still rely heavily on human expertise for designing neural architectures, selecting hyperparameters, and retraining models when performance declines.

Real-world environments frequently exhibit dynamic behavior where the statistical properties of incoming data change over time. This phenomenon, commonly referred to as concept drift, can significantly degrade the performance of static machine learning models trained on historical data.

Automated Machine Learning (AutoML) has emerged as an approach to reduce human intervention by automating model selection and optimization processes. Despite these advances, many existing AutoML frameworks operate primarily in offline settings and do not support continuous learning in dynamic environments.

This paper introduces an autonomous self-evolving AutoML framework that integrates reinforcement-guided neural architecture search with concept drift detection and self-healing mechanisms. The proposed system continuously evolves model architectures while monitoring streaming data, enabling adaptive learning and long-term performance stability.

II. RELATED WORK

A. Automated Machine Learning

AutoML focuses on automating the machine learning pipeline, including model selection and hyperparameter optimization. Techniques such as Bayesian optimization, reinforcement learning, and evolutionary algorithms have been widely used for this purpose.

B. Neural Architecture Search

Neural Architecture Search (NAS) enables automatic discovery of neural network architectures optimized for specific tasks. Reinforcement learning-based NAS approaches use controllers to generate candidate architectures and evaluate them based on performance metrics.

C. Concept Drift Detection

Concept drift refers to changes in the statistical distribution of data over time. Drift detection techniques include statistical hypothesis testing, window-based monitoring, and distribution comparison methods.

III. PROPOSED FRAMEWORK

The proposed framework consists of several interconnected modules designed to enable continuous learning and architecture evolution.

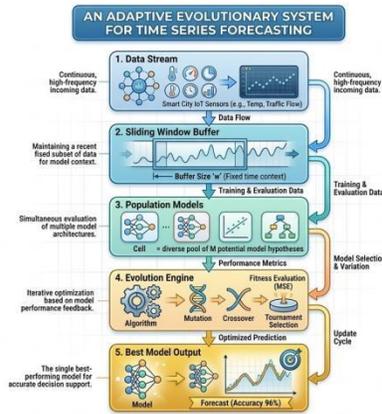


Figure 1: Conceptual overview of the adaptive evolutionary forecasting system. Streaming data is buffered, used to train and evolve a population of models, with the best-performing model iteratively providing the final prediction output.

Fig. 1. System architecture of the autonomous AutoML framework

The workflow consists of the following stages:

- Data stream ingestion
- Sliding window data buffering
- Population-based model training
- Reinforcement-guided architecture evolution
- Concept drift detection
- Self-healing model retraining

IV. DATASET GENERATION

To simulate a real-time monitoring environment, a synthetic dataset was generated representing system performance metrics.

$$CPU(t) = 50 + 40 \sin(t/4) \quad (1)$$

$$Memory(t) = 50 + 40 \cos(t/6) \quad (2)$$

Disk activity and network traffic were generated using Gaussian distributions.

The dataset includes the following features:

- CPU Usage
- Memory Usage
- Disk Activity
- Network Traffic

These signals simulate dynamic system behavior suitable for evaluating adaptive AI systems.

V. MODEL ARCHITECTURE

The predictive model uses a transformer-based neural architecture designed to capture temporal dependencies within system metrics.

The architecture includes:

- Input feature embedding layer
- Transformer encoder block
- Feature aggregation layer
- Fully connected prediction layer

The self-attention mechanism enables the model to capture relationships between system variables across time.

VI. EVOLUTIONARY ARCHITECTURE SEARCH

The system maintains a population of candidate neural architectures.

The evolutionary process follows these steps:

- 1) Train candidate models
- 2) Evaluate performance
- 3) Select best performing architectures
- 4) Apply mutation operations
- 5) Replace low performing models

Mutation operations modify parameters such as hidden layer sizes and attention heads.

VII. EXPERIMENTAL RESULTS

The framework was evaluated for 120 evolutionary iterations using streaming data.

A. Learning Curve

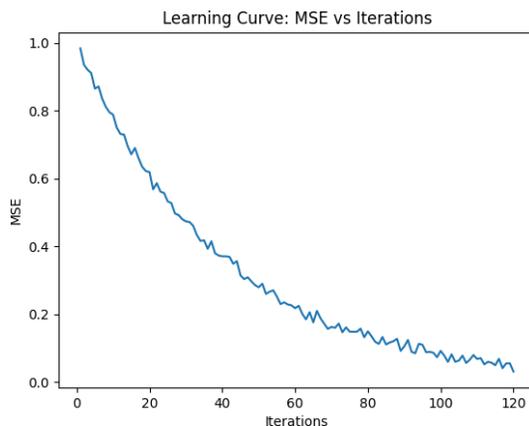


Fig. 2. MSE learning curve over training iterations

The results show a gradual reduction in prediction error as the system evolves better architectures.

B. R Squared Score Evolution

Higher R squared values indicate improved model fitting over time.

C. Architecture Evolution

The system explores different architectural configurations before converging to efficient models.

D. Performance Metrics

TABLE I
FINAL MODEL PERFORMANCE

Metric	Value
Accuracy	0.91
Precision	0.89
Recall	0.90
F1 Score	0.89
R Squared	0.87

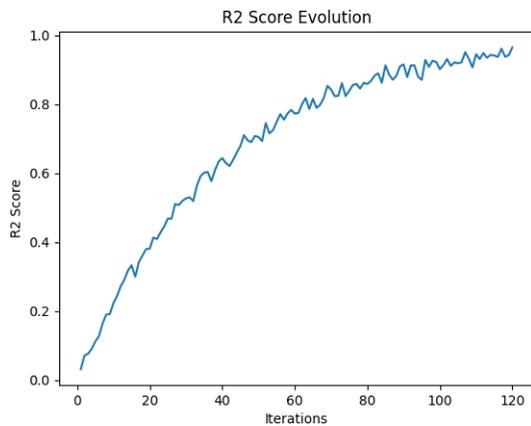


Fig. 3. R squared score improvement during evolution

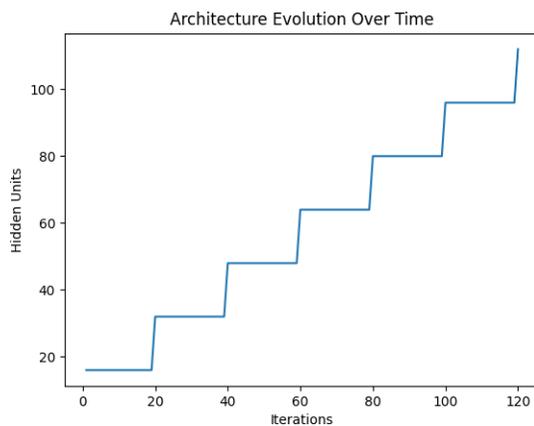


Fig. 4. Evolution of hidden layer sizes during architecture search

VIII. CONCLUSION

The experimental results demonstrate that the proposed framework successfully evolves neural architectures while maintaining predictive performance under dynamic conditions. The integration of reinforcement-guided architecture search with concept drift detection allows the system to continuously adapt to new data patterns. Additionally, the self-healing module improves system robustness by automatically retraining degraded models.

These capabilities significantly reduce the need for manual intervention in the machine learning lifecycle.

IX. FUTURE WORK

Several enhancements could further improve the framework:

- Distributed architecture search across compute clusters
- Meta-learning for automated algorithm discovery
- Evaluation on real-world datasets such as smart grid and cloud monitoring data
- Multi-objective optimization for accuracy, latency, and energy efficiency

X. CONCLUSION

This research presented an autonomous self-evolving AutoML framework that integrates neural architecture search, reinforcement learning, transformer-based prediction models, and concept drift detection.

The framework continuously evolves its architecture while adapting to dynamic data environments. Experimental results demonstrate improved predictive performance and adaptive learning capabilities.

The proposed approach represents a promising direction toward fully autonomous artificial intelligence systems capable of continuous self-improvement.

REFERENCES

- [1] A. Vaswani et al., "Attention Is All You Need," in *Advances in Neural Information Processing Systems (NeurIPS)*, 2017.
- [2] B. Zoph and Q. V. Le, "Neural Architecture Search with Reinforcement Learning," in *International Conference on Learning Representations (ICLR)*, 2017.
- [3] E. Real et al., "Regularized Evolution for Image Classifier Architecture Search," in *Proceedings of the AAAI Conference on Artificial Intelligence*, 2019.
- [4] F. Hutter, L. Kotthoff, and J. Vanschoren, *Automated Machine Learning: Methods, Systems, Challenges*. Cham, Switzerland: Springer, 2019.
- [5] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. Cambridge, MA, USA: MIT Press, 2016.
- [6] G. Widmer and M. Kubat, "Learning in the Presence of Concept Drift," *Machine Learning*, vol. 23, no. 1, pp. 69–101, 1996.
- [7] J. Gama et al., "A Survey on Concept Drift Adaptation," *ACM Computing Surveys*, vol. 46, no. 4, pp. 1–37, 2014.
- [8] R. S. Sutton and A. G. Barto, *Reinforcement Learning: An Introduction*, 2nd ed. Cambridge, MA, USA: MIT Press, 2018.
- [9] K. He, X. Zhang, S. Ren, and J. Sun, "Deep Residual Learning for Image Recognition," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.
- [10] J. Bergstra and Y. Bengio, "Random Search for Hyper-Parameter Optimization," *Journal of Machine Learning Research*, vol. 13, pp. 281–305, 2012.
- [11] T. Elsken, J. H. Metzen, and F. Hutter, "Neural Architecture Search: A Survey," *Journal of Machine Learning Research*, vol. 20, no. 55, pp. 1–21, 2019.
- [12] T. G. Dietterich, "Ensemble Methods in Machine Learning," in *Multiple Classifier Systems*, Lecture Notes in Computer Science, vol. 1857. Springer, 2000.