Impact Factor (QJIF): 8.47 E-ISSN: 2583-1615, P-ISSN: 3049-3498

International Journal of Research in Academic World

Received: 04/December/2025 **IJRAW:** 2026; 5(1):51-53 **Accepted:** 09/January/2026

Use of Learned Data Structures in Machine Learning and AI Algorithms

[1]Rishik Jariwala and [*2]Radhika Patwardhan

[1]Student, Diploma in Information Technology, SVKM,s Shri Bhagubhai Mafatlal Polytechnic & College of Engineering, Vile Parle, Mumbai, Maharashtra, India.

[*2]Professor, Faculty in Information Technology, SVKM,s Shri Bhagubhai Mafatlal Polytechnic & College of Engineering, Vile Parle, Mumbai, Maharashtra, India.

Abstract

The rapid growth of modern data has driven a shift from static storage designs to intelligent, data-aware architectures. Traditional storage systems struggle with the sparse and high-dimensional data used in AI, creating the need for advanced solutions such as multidimensional indexing, sparse tensors, and graph-based frameworks. This evolution introduced learned data structures, which apply machine learning to predict data locations by modeling underlying data distributions. Predictive indexes like the PGM-index and Recursive Model Index replace rigid tree structures, significantly improving memory efficiency and search performance.[2] However, these advances also demand changes in data science education. Academic curricula are moving away from outdated procedural languages toward flexible platforms like Python, enriched with real-world case studies from distributed systems and search technologies. Hands-on, multi-level experimental environments prepare students to manage large-scale data effectively. Integrating predictive storage architectures with modern teaching approaches equips future professionals to handle today's complex and data-intensive information landscape efficiently.

Keywords: Data-aware architectures, learned data structures, machine-learning indexing, PGM-index, Recursive Model Index, sparse high-dimensional data.

1. Introduction

The rapid growth of large-scale, high-dimensional data is driving a structural shift in high-performance computing and data management. Traditional data structures, designed for static and general-purpose workloads, struggle to meet the efficiency requirements of modern machine learning and AI systems. To address these limitations, specialized architectures such as sparse tensors and graph frameworks have emerged alongside learned data structures that integrate machine learning into world,

- Inability to exploit data-specific patterns, leading to suboptimal time and space efficiency.
- Poor handling of sparse data, causing unnecessary memory usage and slower computation.
- High procedural overhead in structures such as B-trees due to repeated comparisons.
- Lack of predictive logic for faster data.

2. Limitations in Traditional Data Structures

Conventional data structures are increasingly inadequate for modern AI and ML workloads due to their rigid and general-purpose design. These frameworks struggle to efficiently handle large-scale, high-dimensional, and sparse datasets commonly used in intelligent systems. Key limitations include:

- **Indexing as Regression:** Search is modeled as learning the data's cumulative distribution function (CDF).
- **Predictive Search:** ML models predict key positions, reducing comparison overhead.
- **Error Correction:** Local binary or exponential search handles prediction inaccuracies.
- **Key Architectures:** Recursive Model Index (RMI), PGM-index, ALEX, and Learned Bloom Filters.
- **Performance Trade-offs:** Significant speed and memory gains, but sensitivity to data distribution changes remains a challenge.

3. Learned Data Structures (Concept)

i). Learned Data Structures (LDS) represent a paradigm shift in data management by integrating machine learning models into traditional indexing and storage logic.[1] Scope of the Study.

ii). **Predictive Search:** Key positions are estimated, minimizing traversal and comparisons.

***Corresponding Author:** Radhika Patwardhan

iii). **Error Correction:** Local binary or exponential search compensates for model inaccuracies.

iv). **Hierarchical Models:** Recursive Model Index (RMI) routes queries through layered models.

v). **Piecewise Models:** PGM-index and FITing-tree approximate distributions with bounded error [2].

vi). **Dynamic Indexing:** ALEX supports updates and adaptive restructuring for evolving datasets [4].

4. Predictive Indexing Techniques

Predictive indexing replaces comparison-based search with model-driven estimation of data locations by learning underlying data distributions. Instead of traversing static trees, these techniques predict the position of keys directly, enabling faster access and reduced memory usage.

Some Techniques are:

- **Regression-Based Indexing:** Indexes are modeled as regression functions that learn the CDF or rank of keys.[1]
- Predictive indexing using CDF-based regression models.
- RMI, PGM-index, and ALEX for efficient and adaptive data access.
- Learned Bloom Filters for space-efficient membership testing [3].
- Learned Count-Min sketches for accurate frequency estimation.
- Performance gains with sensitivity to data distribution changes.

5. Learned Data Structures in ML

Learned data structures play a significant role in modern machine learning systems by replacing generic, rule-based logic with data-aware predictive models. Instead of relying on fixed comparison strategies, these structures learn statistical patterns in data to optimize access, storage, and computation. By modeling indexes as regression problems, LDS improve efficiency in ML pipelines where large-scale, dynamic datasets are common. Their integration enhances training speed, inference latency, and memory utilization, making them well suited for AI-driven workloads.

Some key applications and mechanisms include:

- **Computer Vision & Spatial AI:** Predictive indexing enhances nearest-neighbor search, outperforming KD-trees and R-trees.
- **Natural Language Processing:** Trie-based and learned inverted indexes improve string matching and query retrieval.
- **Web Intelligence:** Learned Bloom Filters reduce memory usage for large-scale membership testing.[3]
- **Network Traffic Analysis:** Learned Count-Min sketches accurately detect frequent data patterns.
- **System-Level AI:** Predictive models optimize operating systems and query planners for autonomous infrastructure [5].

6. Application in Artificial Intelligence

Learned and specialized data structures have become integral to artificial intelligence systems, enabling faster data access, efficient memory usage, and intelligent decision-making across diverse domains. By embedding predictive logic into storage and retrieval mechanisms, AI applications can scale effectively with massive and complex datasets.

Some major applications include:

- **Memory Efficiency:** Learned indexes can be orders of magnitude smaller than B+-trees; sparse tensors and learned Bloom filters further reduce space usage [2].
- **Latency Improvements:** Predictive indexing achieves 1.5–3× faster lookups by replacing multi-level searches with model predictions.
- **Inference Overhead:** Model execution introduces latency compared to simple hash functions.
- **Accuracy vs Guarantees:** Performance depends on data distribution, requiring correction mechanisms.
- **Training Cost:** Learned structures incur additional training and maintenance overhead.

7. Performance Benefits and Tradeoffs

Learned and specialized data structures offer notable gains in storage optimization and data retrieval efficiency; however, these gains come with trade-offs in robustness and computational overhead. Their effectiveness depends on workload characteristics, data stability, and

- **Robustness Gaps:** Performance degrades when query distributions shift from training data [6].
- **Dynamic Updates:** Insertions and deletions require retraining, adding computational overhead [4, 6].
- **Inference Latency:** Model execution is slower than traditional hash functions.
- **Algorithmic Limits:** Efficient handling of variable-length keys remains unresolved.
- **Security Risks:** Learned structures are vulnerable to adversarial input patterns.
- **Educational Gap:** Traditional curricula lag behind industry-scale, data-aware architectures.

8. Challenges and Open Research Issues

Despite their advantages, learned data structures face several technical, practical, and educational challenges that limit widespread adoption. These issues define the active research frontier in data-aware systems.

- **Dependence on Data Distribution:** Learned structures rely heavily on training data. Performance may degrade when query or data distributions shift significantly.
- **Dynamic Updates and Retraining:** Insertions and deletions often require model retraining or structural reorganization, increasing computational overhead.
- **Inference Latency:** Model-based predictions can introduce higher latency compared to traditional hash functions or tree traversal methods.
- **Limited Support for Variable-Length Keys:** Most current learned models are designed for fixed-length keys, making extension to variable-length data an open research problem.
- **Algorithmic Limits:** Efficient handling of variable-length keys remains unresolved.
- **Security Risks:** Learned structures are vulnerable to adversarial input patterns.

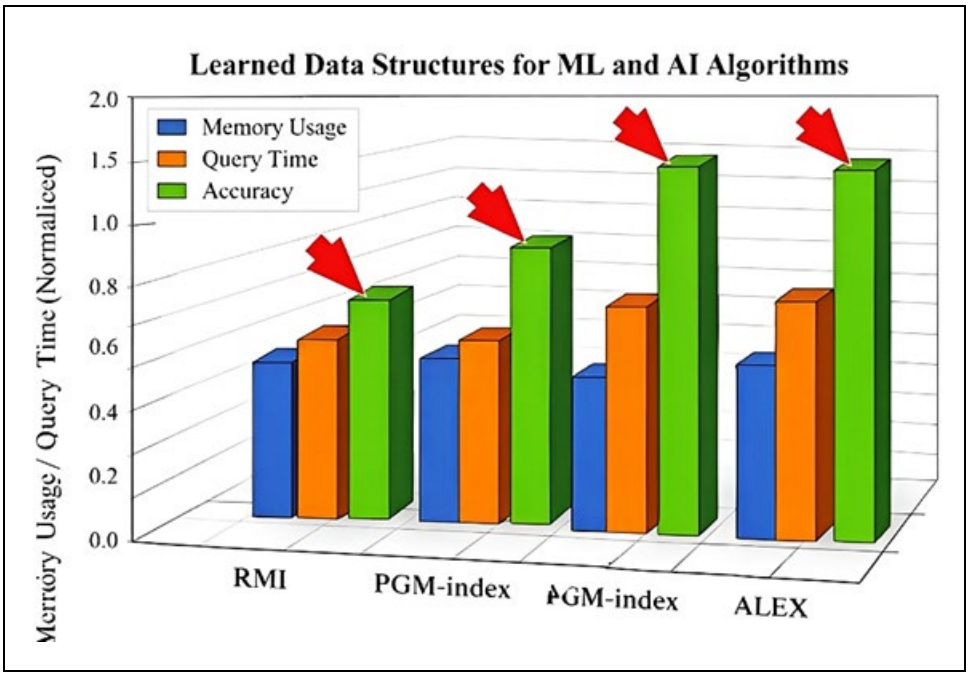

Fig 1: Performance comparison of learned data structures.

9. Conclusion

This project demonstrates how learned data structures can be effectively integrated into machine learning and AI systems to improve data access efficiency beyond traditional indexing methods. By organizing data in a hierarchical manner and using learned models to approximate data distributions at different results, the system achieves faster query performance and better adaptability to real-world data patterns. Unlike classical data structures that rely on rigid rules and worst-case guarantees, learned data structures exploit statistical regularities in data, making them well suited for modern, data-intensive AI applications. The study shows that combining machine learning models with indexing mechanisms can significantly enhance the quality.

References

1. Kraska T, Beutel A, Chi EH, Dean J, Polyzotis N. The Case for Learned Index Structures. *Proceedings of the 2018 International Conference on Management of Data (SIGMOD)*. 2018.
2. Ferragina P, Vinciguerra G. The PGM-Index: A Fully-Dynamic Compressed Learned Index with Provable Worst-Case Bounds. *Proceedings of the VLDB Endowment*. 2020.
3. Mitzenmacher M. A Model for Learned Bloom Filters. *Proceedings of the 2018 IEEE Conference on Data Compression*. 2018.
4. Ding J, Chen U, Wei V. ALEX: An Updatable Adaptive Learned Index. *Proceedings of the 2020 ACM SIGMOD Conference*. 2020.
5. Marcus R, Negi P, Mao H, Tatbul N, Kraska T. Neo: A Learned Query Optimizer. *Proceedings of the VLDB Endowment*. 2019.
6. Bu S, Howe B, Balazinska M, Ernst MD. HAWK: Efficiently Supporting Ad Hoc Analytics on Evolving Data. *Proceedings of the VLDB Endowment*. 2010.