AI-Driven Database Management: A Survey

www.surveyx.cn

Abstract

This survey paper explores the transformative role of artificial intelligence (AI) and machine learning (ML) in enhancing database management systems (DBMS). By integrating AI-driven methodologies, modern DBMS have achieved significant advancements in query optimization, resource allocation, and anomaly detection, addressing the limitations of traditional systems. Key innovations include learned indexes, such as the PGM-index, which improve query latency and space efficiency, and AI-driven frameworks like openGauss that enhance performance through innovative architectural designs. The paper also examines the dual perspective of AI for databases (AI4DB) and databases for AI (DB4AI), highlighting how AI augments database functionalities and supports AI model deployment. Additionally, the survey discusses the integration of multi-task meta-learning frameworks, which facilitate transferable knowledge across tasks and databases, and the evolution of AI-driven text-to-SQL systems. Challenges such as model interpretability and the integration of heterogeneous data sources are addressed, emphasizing the need for more dynamic and adaptable data management solutions. The paper concludes by underscoring the potential of AI and ML to revolutionize database management, paving the way for more intelligent, scalable, and efficient data-driven applications.

1 Introduction

1.1 Significance of AI and ML in Database Management

The integration of artificial intelligence (AI) and machine learning (ML) into database management systems (DBMS) represents a significant advancement in modern data handling and processing frameworks. Traditional database systems often face challenges in scalability, reliability, and efficiency, particularly in big data and dynamic cloud contexts [1]. AI and ML effectively address these challenges by automating complex tasks such as query optimization and resource allocation, thereby enhancing DBMS efficiency and scalability.

AI-driven techniques, including learned query superoptimization, allow query optimizers to learn from historical performance, improving efficiency for repetitive queries within modern analytics systems [2]. Frameworks like Hydro utilize real-time execution data to dynamically adjust query plans, reducing execution time and improving resource utilization [3]. These advancements highlight AI's crucial role in optimizing database operations and enhancing overall system performance.

Additionally, the application of deep learning for cardinality estimation significantly improves the accuracy and efficiency of query optimization processes [4]. This enhancement is particularly vital in environments that require precise query execution plans to manage complex data interactions effectively.

The adoption of AutoML techniques is pivotal for automating ML processes, especially in IoT data analytics, improving efficiency in handling dynamic datasets [5]. This automation is essential for managing the vast and complex datasets characteristic of modern DBMS [6]. However, challenges remain, such as model interpretability and the integration of multiple data sources, as evidenced by limitations in current tabular question-answering systems [7]. Addressing these challenges is

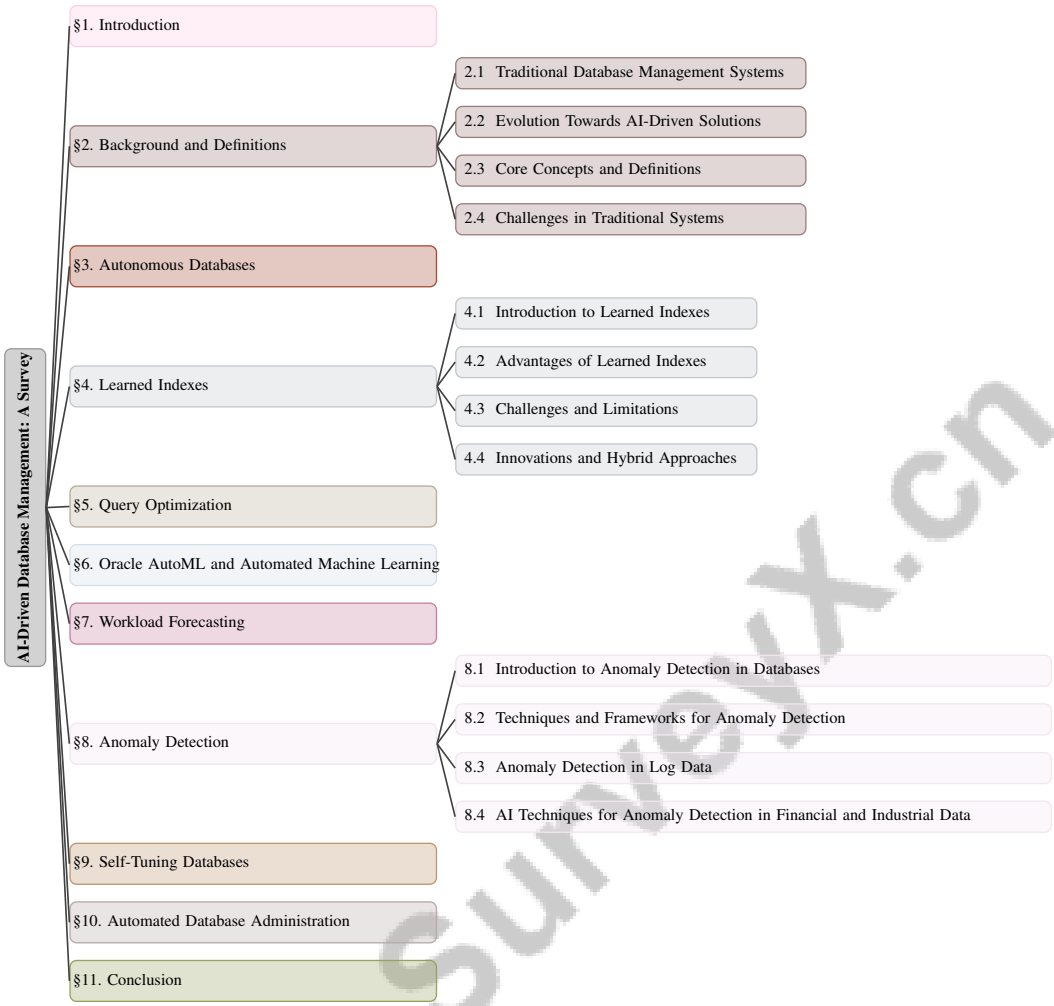


Figure 1: chapter structure

crucial for the effective deployment of AI and ML in DBMS, paving the way for more dynamic and adaptable data management solutions [8].

The transformative impact of AI and ML on database management lies in their ability to modernize DBMS through automation and optimization, enhancing performance and scalability while addressing traditional systems' limitations. Nevertheless, the lack of public large-scale real-world datasets has hindered AIOps development, complicating fair comparisons and ensuring generality and real performance [9]. Moreover, integrating heterogeneous production data is essential for enhancing industrial analytics, emphasizing the need for AI-driven solutions to manage diverse data sources effectively [10].

1.2 Objectives of the Survey

This survey aims to illuminate the transformative role of AI-driven methodologies in enhancing database management systems (DBMS) by integrating AI and ML to mitigate traditional systems' limitations. Key objectives include improving incident management and system supervision, which are essential components of modern IT operations [11]. The survey elucidates the dual perspectives of AI for databases (AI4DB) and databases for AI (DB4AI), highlighting how AI can augment database functionalities and how database techniques can support AI model deployment and optimization [12].

Another critical objective is to explore multi-task meta-learning frameworks that capture transferable knowledge across tasks and databases, promoting efficient learning processes [13]. This aligns with

the broader goal of developing interconnected, end-to-end frameworks that enhance big data quality through improved assessment, anomaly detection, and correction processes [14].

The survey further aims to enhance the interpretability of ML models by proposing a synthesis of k-nearest neighbors with information theory [15]. By addressing challenges faced by DBMS users, the research uncovers opportunities for automating DBMS administration [16]. Additionally, the survey examines the challenges of applying ML techniques to IoT data analytics, focusing on model selection, tuning, and updating processes [5].

Moreover, the survey provides a comprehensive overview of the evolution of AI-driven text-to-SQL systems, emphasizing advancements in large language model architectures and the pivotal role of datasets in fostering progress [17]. It investigates approximation schemes for many-objective query optimization (MOQO), which efficiently generate near-optimal plans, significantly reducing the time required for exhaustive optimization [18]. Furthermore, the survey addresses challenges in ML-based resource management in cloud computing, underscoring the need for intelligent resource management strategies that adapt to dynamic workloads [6]. It encompasses the evolution of database systems, including object-relational integration, web services, transaction processing, data mining, and the incorporation of various data types [8].

1.3 Overview of Key Terms

In the realm of AI-driven database management, several key terms are pivotal to understanding the integration of AI and ML technologies within database systems. The concept of AI for Databases (AI4DB) encompasses learning-based techniques for configuration tuning, query optimization, index and view advising, and enhancing security measures [12]. These methodologies aim to automate and optimize database operations by leveraging AI capabilities to improve performance and resilience.

Conversely, Databases for AI (DB4AI) refers to the development of AI-oriented declarative languages and data governance practices that facilitate efficient data management. This includes accelerating both the training and inference phases of AI models, enabling swift deployment and execution within database environments [12]. The synergy between AI4DB and DB4AI underscores the dual role that databases play in supporting AI applications and benefiting from AI-driven enhancements.

AutoML signifies the automation of ML processes, particularly relevant in IoT data analytics. This involves the automatic selection, tuning, and updating of models to manage concept drift, ensuring analytics remain accurate amid changing data patterns [5]. The application of AutoML in database systems is crucial for handling the dynamic and complex datasets characteristic of modern IoT environments.

The integration of AI and ML into DBMS holds significant transformative potential, as evidenced by recent advancements and research findings. These technologies promise to enhance the adaptability and efficiency of DBMS, enabling intelligent systems capable of effectively managing the increasing volume and complexity of contemporary data-driven applications. For instance, the development of unified models like MTMLF showcases the capability to transfer knowledge across tasks and databases, addressing limitations in traditional ML approaches. Furthermore, AI-driven frameworks are being designed to improve data quality in big data ecosystems, emphasizing the importance of accurate data for informed decision-making. As automation tools for DBMS administration evolve, they present both challenges and opportunities for industry-wide adoption, ultimately leading to more responsive and automated data management solutions [13, 17, 16, 14, 19].

1.4 Structure of the Survey

The survey is meticulously structured to provide a comprehensive exploration of AI-driven database management, beginning with an introduction that establishes the significance of integrating AI and ML into DBMS. This section highlights the transformative impact of these technologies on modernizing data handling frameworks. Following this, the survey delineates its objectives, emphasizing the dual perspective of AI for databases (AI4DB) and databases for AI (DB4AI), and explores the integration of multi-task meta-learning frameworks and the evolution of AI-driven text-to-SQL systems.

The second major section provides a background on traditional DBMS and their evolution towards AI-driven solutions, defining core concepts such as autonomous databases, learned indexes, and query

optimization. It also addresses the challenges faced by traditional systems, which AI solutions aim to mitigate.

Subsequent sections delve into specific aspects of AI-driven database management. Section three focuses on autonomous databases, discussing their self-management capabilities and the AI-driven frameworks that support these functionalities. The fourth section examines learned indexes, highlighting their role in improving data retrieval speeds and exploring recent innovations.

The fifth section explores query optimization through AI and ML, discussing learned models, neuro-symbolic approaches, and reinforcement learning. Section six introduces Oracle AutoML, detailing its role in automating ML tasks within databases and its impact on workload efficiency.

In section seven, the survey discusses workload forecasting techniques, emphasizing the limitations of traditional methods and introducing innovative models like QueryBot 5000. Section eight examines anomaly detection methods in database operations, focusing on techniques applicable to log data and specific industry contexts.

Section nine explores self-tuning databases, highlighting frameworks that enable automatic configuration adjustments for optimal performance. Finally, section ten discusses automated database administration, exploring tools and techniques that reduce manual intervention and the challenges involved in transitioning to automated systems.

The survey concludes with a comprehensive summary of key findings that reflect the current landscape of AI-driven database management, emphasizing the dual benefits of AI enhancing database intelligence and databases optimizing AI models. It also highlights persistent challenges, such as limited industry adoption of automation tools and the need for improved text-to-SQL systems, while suggesting future research directions that include extending capabilities to NoSQL databases, addressing domain generalization, and enhancing scalability for real-world applications [17, 12, 16]. This structured approach ensures a thorough examination of the integration of AI and ML into database management, providing valuable insights for researchers and practitioners alike.The following sections are organized as shown in Figure 1.

2 Background and Definitions

2.1 Traditional Database Management Systems

Traditional database management systems (DBMS) have been pivotal in data storage and retrieval, relying on structured query language (SQL) and predefined schemas to maintain data consistency and integrity [8]. These systems emphasize transactional consistency, durability, and isolation, adhering to the ACID properties crucial for reliable transaction processing. However, they face significant challenges in big data and cloud computing contexts, such as scalability issues that lead to performance bottlenecks when managing vast data volumes [1]. Their static schemas hinder flexibility, complicating adaptation to evolving data needs [8]. Additionally, extensive manual intervention for configuration, tuning, and maintenance makes traditional DBMS resource-intensive and error-prone [20], limiting real-time resource optimization [3]. Moreover, they often lack advanced analytics capabilities necessary for extracting insights from complex and unstructured data sources [10].

These limitations underscore the need for adaptive and intelligent solutions that integrate seamlessly with modern data ecosystems. The shift towards AI-driven DBMS, leveraging machine learning and automation, is essential for enhancing performance, scalability, and operational efficiency. Autonomous architectures, such as Peloton, exemplify this transition by optimizing themselves for current and anticipated workloads without human intervention, addressing the constraints of traditional systems reliant on manual decision-making [21, 16].

2.2 Evolution Towards AI-Driven Solutions

The evolution of database management systems (DBMS) towards AI-driven solutions marks a shift from static architectures to dynamic frameworks capable of managing the complexity and volume of modern data [21]. Self-driving DBMS, which autonomously optimize performance without human oversight, exemplify this transformation [21]. A key advancement is the development of learned indexes, which use machine learning models to enhance data retrieval based on distribution and query patterns, outperforming conventional static indexing methods [22, 23]. AI technologies have also been

integrated into query optimization, employing learning-to-rank approaches to rectify inefficiencies in traditional methods [24].

AI-driven methodologies extend to multimodal data retrieval platforms, efficiently extracting information from diverse sources and managing data growth through intelligent handling [25]. Systems like openGauss demonstrate this integration, catering to high performance, availability, security, and intelligence in modern database services [26]. AI solutions address traditional system limitations in managing complex queries, particularly in text-to-SQL scenarios, enhancing query processing efficiency and accuracy [7]. The incorporation of AI into DBMS signifies a transformative advancement in overcoming traditional system challenges, promising intelligent, scalable, and efficient data management solutions [8].

2.3 Core Concepts and Definitions

Understanding the integration of artificial intelligence (AI) and machine learning (ML) technologies in database management is essential. Core concepts include AI for Databases (AI4DB), which refines traditional optimization techniques for large-scale data, and Databases for AI (DB4AI), which facilitates model deployment and accelerates algorithm performance. These concepts highlight the dual role of databases in supporting AI applications and benefiting from AI-driven enhancements [13, 17, 12, 16, 19].

Autonomous databases, characterized by self-management, self-tuning, and self-repair capabilities, reduce manual oversight and minimize human error by leveraging AI and ML algorithms. This autonomy allows for dynamic adaptation to changing workloads and real-time resource optimization, enhancing system performance and reliability [27]. Learned indexes employ machine learning models to predict data positions within sorted arrays, adapting to data distribution and query patterns for improved retrieval speeds and reduced memory usage [28, 22]. Benchmarks for learned indexes provide open-source implementations and real-world dataset repositories, enabling fair performance comparisons [29]. However, challenges such as inefficiencies in managing real-world data distributions persist [29].

Query optimization aims to determine the most efficient execution plan for queries. AI and ML techniques revolutionize this field by introducing models that predict query performance metrics and select optimal execution strategies. Cardinality estimation benefits from machine learning approaches that provide accurate predictions of the number of rows returned by a query [30, 31]. The integration of neural networks into learned indexes further enhances performance through improved time and space efficiency [32]. The integration of AI and ML into DBMS holds transformative potential, fostering intelligent, adaptive, and efficient systems capable of addressing contemporary data-driven application complexities [14, 13, 19, 16].

2.4 Challenges in Traditional Systems

Traditional database management systems (DBMS) face numerous challenges in adapting to rapidly evolving data environments. A primary issue is the static nature of query optimization, leading to performance bottlenecks, particularly with user-defined functions (UDFs) where execution costs and selectivity vary based on input data [3]. Fixed parameter values exacerbate this problem, resulting in suboptimal query plans [33]. Cardinality estimation is another challenge, as traditional methods often fail to accurately model complex data distributions, leading to inefficiencies and inaccuracies, especially with increasing query joins [34].

Learned indexes, while promising, reveal challenges as they are primarily designed for in-memory use and do not effectively reduce block reads and writes crucial for disk operations [35]. This limitation is evident in traditional schema-on-write systems, multi-model databases, vector databases, and data lakes, which struggle with multimodal data and rich hybrid queries [25]. Traditional query optimizers often make locally optimal decisions, overlooking globally optimal plans, particularly when common subexpressions are present [36]. Probabilistic inference methods in these systems either operate in exponential time or require numerous iterations to converge, making them impractical for large-scale applications [37].

Additional challenges include high read latencies, inefficient resource utilization during failover, and inadequate support for modern hardware architectures, limiting traditional methods' effectiveness

in meeting evolving user requirements [26]. Existing benchmarks, often based on synthetic data, fail to accurately represent real-world system behavior, complicating the evaluation and comparison of traditional systems [9]. Traditional data integration methods also face challenges in aligning ontologies with industrial data, leading to mismatches that complicate analytics and integration processes [10]. These challenges highlight the need for AI-driven solutions to enhance DBMS adaptability, efficiency, and intelligence, overcoming traditional approaches' limitations and paving the way for advanced data management frameworks.

3 Autonomous Databases

Autonomous databases signify a major advancement in data management, characterized by their ability to function with minimal human intervention through sophisticated automation and optimization techniques. This section delves into their self-management and optimization capabilities, highlighting the integration of machine learning and control theory to anticipate workloads and optimize configurations proactively. Figure 2 illustrates the hierarchical structure of autonomous databases, emphasizing two primary aspects: Self-Management and Optimization, and AI-Driven Frameworks and Architectures. The first category explores the integration of machine learning and resource allocation, showcasing advancements such as openGauss and AutoTQA. The second category focuses on optimization frameworks and performance enhancements, emphasizing tools like OnlineTune and PandaDB. Together, these elements underscore the transformative impact of AI and machine learning in enhancing database performance and adaptability.

3.1 Self-Management and Optimization

Autonomous databases excel in self-management and optimization, reducing human dependency by integrating advanced machine learning and control theory. Systems like openGauss exemplify this integration with NUMA-aware processing and AI-driven optimization to enhance database performance [26]. Machine learning models improve query planning and execution, as demonstrated by BN-QP, which optimizes query execution by inferring missing attribute values [34]. Ontology reshaping in Knowledge Graphs simplifies query structures, enhancing industrial analytics [10].

As illustrated in Figure 3, the hierarchical structure of self-management and optimization in autonomous databases highlights the key areas of machine learning integration, resource allocation, and query processing enhancements. Innovations like Meta Model-based Predictive Autoscaling (MMPA) use reinforcement learning for resource allocation in cloud environments, enabling databases to adapt dynamically to workload changes [38]. Multimodal data retrieval platforms emphasize efficient management of diverse data types, essential for modern databases [25]. AutoTQA leverages large language models for text-to-SQL queries, enhancing query processing [7]. These advancements demonstrate how AI and machine learning empower autonomous databases to adapt to changing workloads and optimize resources, significantly enhancing data management performance and reliability.

3.2 AI-Driven Frameworks and Architectures

AI-driven frameworks and architectures are crucial for autonomous databases, enabling efficient management of complex data environments. OnlineTune exemplifies this with contextual Bayesian optimization and safety assessment, enhancing database reliability [39]. Planter's modular design supports diverse machine learning algorithms, offering flexibility in managing workloads [40]. GFTR framework improves performance by minimizing random memory accesses, enhancing cache locality [41].

Instance-optimized components adapt mechanisms for optimal performance based on datasets and workloads, crucial for dynamic environments [42]. Workload-aware vector data partitioning and multi-query optimization improve query throughput [43]. PandaDB uses advanced indexing for semantic understanding and accelerated query execution [44]. MQRLD supports transparent data storage and hybrid queries, essential for multimodal data [25].

The openGauss system incorporates NUMA-aware processing and adaptive symmetric multiprocessing for enhanced performance and security [26]. AutoTQA uses large language models for efficient multi-table query processing [7]. These frameworks and architectures enable autonomous databases

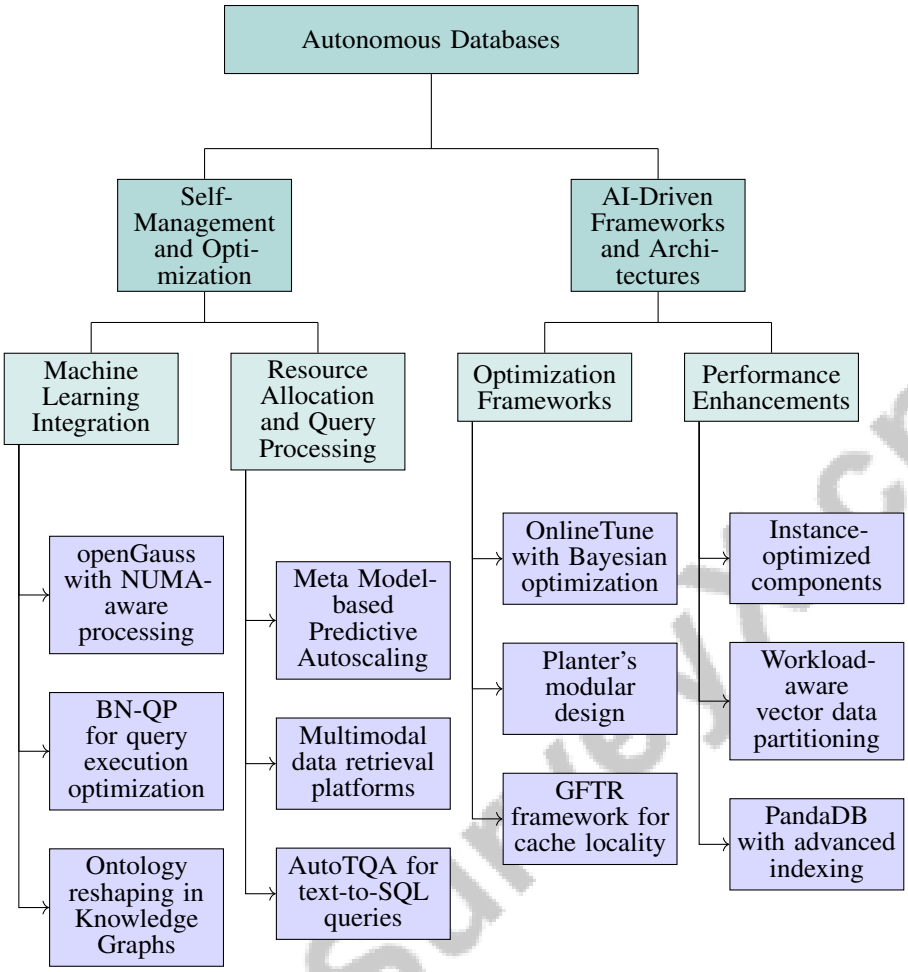


Figure 2: This figure illustrates the hierarchical structure of autonomous databases, highlighting two primary aspects: Self-Management and Optimization, and AI-Driven Frameworks and Architectures. The first category delves into machine learning integration and resource allocation, showcasing advancements like openGauss and AutoTQA. The second category focuses on optimization frameworks and performance enhancements, emphasizing tools such as OnlineTune and PandaDB. Together, these elements underscore the transformative impact of AI and machine learning in enhancing database performance and adaptability.

to optimize performance, predict workloads, and adapt to complex data environments, streamlining management and supporting advanced features like in-database AI analytics and self-optimization, enhancing efficiency and reliability across industries [21, 19, 8, 45].

As depicted in Figure 4, AI-driven frameworks and architectures are pivotal in enhancing data systems' efficiency and intelligence. The first component, "User Requests and Next-generation Data System," illustrates the dynamic interaction between user input and data innovation, highlighting seamless integration of user-driven insights. The second component, "A Diagram of a Machine Learning Infrastructure," details a machine learning infrastructure incorporating a PostgreSQL host with a BAIHE extension, showcasing data flow from inference to database. These examples highlight AI-driven architectures' transformative potential in creating responsive and intelligent data systems capable of meeting modern data environments' demands [19, 46].

AI-generated, for reference only.

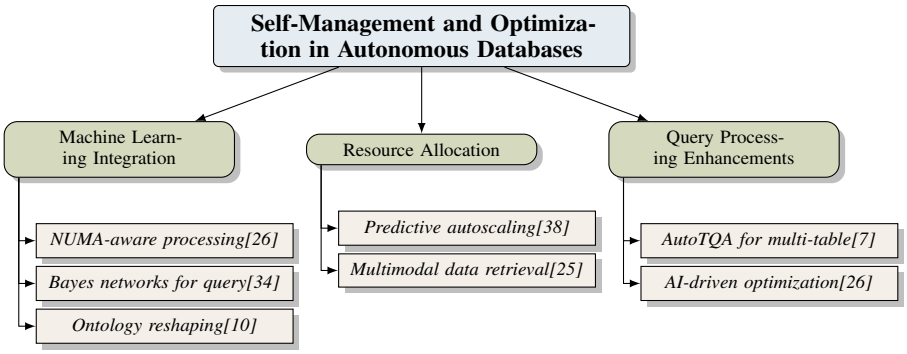
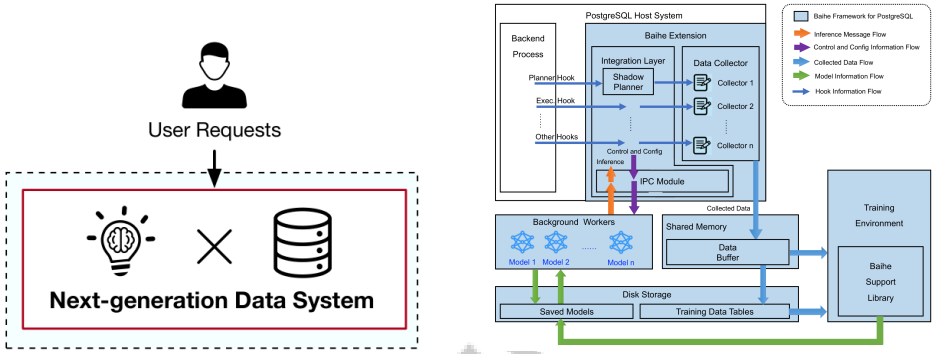


Figure 3: This figure illustrates the hierarchical structure of self-management and optimization in autonomous databases, highlighting key areas of machine learning integration, resource allocation, and query processing enhancements.



(a) User Requests and Next-generation Data System[19]

(b) A Diagram of a Machine Learning Infrastructure[46]

Figure 4: Examples of AI-Driven Frameworks and Architectures

4 Learned Indexes

4.1 Introduction to Learned Indexes

Learned indexes revolutionize data indexing by leveraging machine learning models to predict data positions, offering a dynamic alternative to traditional methods like B-trees and binary search. Unlike static structures that struggle with dynamic datasets due to their inability to adapt to varying data distributions or query patterns [22], learned indexes frame indexing as a machine learning problem, training models to map keys to positions in a sorted array based on rank [47].

A key advantage of learned indexes is their adaptability to diverse data distributions and query workloads. For instance, the PGM-index adapts to dictionary key distributions and access frequencies, functioning as a distribution-aware learned index [22]. Similarly, Doraemon optimizes performance for dynamic workloads by incorporating access patterns and caching trained models [47].

Learned indexes have also expanded to multi-dimensional data, with structures like LIMS using pivot-based mapping to enhance query processing in metric spaces [25]. In external-memory contexts, approaches like AULID optimize on-disk learned indexes to reduce block I/O operations [25].

Despite their potential, the learning processes of learned indexes require further exploration, as current methodologies lack formalized frameworks for evaluating learning objectives and effectiveness, especially as datasets grow in complexity [23, 48]. Establishing systematic approaches to measure and enhance their capabilities is crucial for their broader application in database management systems, offering intelligent, adaptive, and efficient data management solutions.

4.2 Advantages of Learned Indexes

Learned indexes advance database management by using machine learning models to enhance data retrieval efficiency and adaptability, often surpassing traditional indexing methods. They can achieve O(1) expected query time with linear space usage, significantly improving efficiency over structures like B+Trees [49], which is particularly advantageous in complex data environments [22].

Their adaptability to specific data distributions results in faster, more memory-efficient indexing. Innovations like AirIndex optimize hierarchical index designs to create heterogeneous structures tailored to specific environments, enhancing query performance [50]. Frameworks such as Doraemon reduce re-training time by caching and incrementally adapting models for similar access patterns [47].

Learned indexes also excel in managing diverse multimodal data types and supporting complex hybrid queries. The MQRLD platform, for instance, optimizes retrieval through learned indexes [25], while AULID introduces a novel node structure to lower I/O costs in disk-based operations [35].

In dynamic environments, systems like UpLIF employ reinforcement learning to self-tune model structures, ensuring sustained optimal performance without manual intervention [51]. Learned indexes like WaZI demonstrate flexibility in adapting to varying query workloads, reducing latencies and improving retrieval efficiency [52].

The integration of learned indexes into database management systems presents a promising direction for achieving efficient, scalable, and adaptable solutions, addressing traditional indexing limitations and meeting modern data environment demands [6].

4.3 Challenges and Limitations

Implementing learned indexes in database systems presents challenges affecting their performance and applicability. A significant challenge is their dependency on probabilistic assumptions about data distribution, which may not hold in all scenarios, leading to suboptimal performance [49]. This reliance complicates operations in dynamic environments with unpredictable data distributions.

Retraining complexity is another obstacle, increasing linearly with the number of keys and their lengths, causing performance bottlenecks [53]. This complexity is pronounced in dynamic environments with frequent data updates. Benchmark methodologies often focus on read-only workloads, neglecting mixed read/write scenarios, limiting real-world applicability [29].

Disk-based learned indexes like AULID face challenges under high write workloads, where I/O overhead may escalate due to frequent structural modifications [35]. Inaccuracies in block size predictions can also lead to performance bottlenecks [54].

As illustrated in Figure 5, the primary challenges and limitations associated with implementing learned indexes in database systems can be categorized into three main areas: reliance on probabilistic assumptions, complexity of retraining processes, and specific issues related to disk-based learned indexes. Each category highlights key issues such as suboptimal performance, frequent updates, high I/O overhead, and block size inaccuracies, emphasizing the need for ongoing research to enhance the adaptability and efficiency of learned indexes.

Managing learned index structures, as seen in MQRLD, requires careful tuning based on varying query workloads, which can be resource-intensive [25]. The effectiveness of cached models is crucial, as inaccuracies may hinder adaptation to new data distributions [47]. Implementations like AutoIndex primarily focus on read-only indexes, limiting utility in environments with frequent data modifications [55].

Key challenges include defining effective error-correction mechanisms for mispredictions and structuring data layouts to facilitate effective learning by machine learning models [48]. Ongoing research is needed to enhance the adaptability, efficiency, and robustness of learned indexes for diverse and dynamic data environments.

4.4 Innovations and Hybrid Approaches

Recent innovations in learned indexes have significantly improved their adaptability, efficiency, and applicability across varied data environments. The Shift-Table method exemplifies advancements

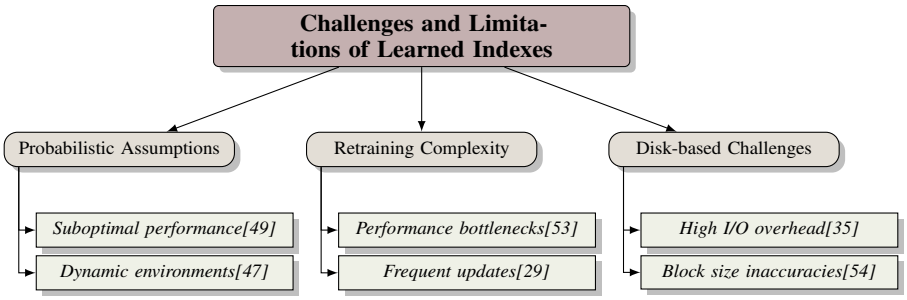


Figure 5: This figure illustrates the primary challenges and limitations associated with implementing learned indexes in database systems. The challenges are categorized into three main areas: reliance on probabilistic assumptions, complexity of retraining processes, and specific issues related to disk-based learned indexes. Each category highlights key issues such as suboptimal performance, frequent updates, high I/O overhead, and block size inaccuracies, emphasizing the need for ongoing research to enhance the adaptability and efficiency of learned indexes.

in optimizing learned index structures, achieving performance improvements up to threefold on real-world datasets compared to existing methods [56].

Hybrid approaches, combining learned indexes with traditional methods, address their respective limitations, offering balanced solutions for complex data environments [48]. Future research is expected to explore these hybrid structures, particularly in disk-resident settings, optimizing performance and resource utilization [57].

Integrating reinforcement learning into learned index structures represents another innovative approach. For example, UpLIF dynamically optimizes index structures through reinforcement learning, ensuring optimal performance in dynamic environments [51]. This facilitates continuous adaptation to changing data patterns and workloads, enhancing database operation efficiency.

Frameworks proposed by Li et al. demonstrate significant improvements in learning efficiency and effectiveness, achieving up to 78x construction speedup and 1.59x query speedup [58]. These advancements enhance scalability and reduce overhead, particularly in dynamic environments requiring frequent updates.

The MQRLD experimental setup includes a comparative analysis using various multimodal datasets to evaluate query performance [25], highlighting the potential of learned indexes in supporting diverse data types and complex queries.

Future research will focus on optimizing learned indexes for external-memory contexts and exploring hybrid approaches that blend learned and traditional methods to leverage their strengths [29]. These innovations and hybrid approaches advance learned indexes, offering promising solutions for efficient, scalable, and adaptable data management systems, transforming database management practices into more intelligent and responsive data systems.

5 Query Optimization

5.1 AI and Machine Learning in Query Optimization

The incorporation of AI and ML into query optimization has significantly advanced DBMS by improving the efficiency, accuracy, and adaptability of data querying. Traditional methods, which rely on static heuristics and fixed cost models, often result in suboptimal execution plans for complex queries [59]. AI and ML introduce adaptive learning and dynamic tuning, markedly enhancing query optimization. Learned Query Optimizers (LQOs), leveraging deep learning, exemplify this shift by capturing complex data patterns and generalizing across unseen queries, leading to more accurate predictions of query execution plans [31, 4]. The PGM-index, for instance, uses linear models to approximate key positions in sorted arrays, aiding efficient data retrieval and improved query optimization [22].

In industrial analytics, reshaping ontologies aligns them with industrial data, optimizing query execution by ensuring data representations meet analytical needs [10]. The COOOL framework applies learning-to-rank techniques to query optimization, significantly improving performance by reducing execution latencies and minimizing regressions [24]. As AI and ML research progresses, their integration into query processes is expected to enhance efficiency, scalability, and adaptability, addressing challenges in data quality management and performance prediction across various industries [14, 60, 61].

5.2 Learned Models and Cardinality Estimation

Learned models are crucial in improving cardinality estimation, a vital component of query optimization in DBMS. Traditional methods often falter with modern data distributions and query patterns, leading to inefficiencies. In contrast, learned models employ advanced ML techniques to provide dynamic and adaptable solutions, significantly enhancing accuracy and efficiency. These models outperform conventional estimators in static environments but face challenges in dynamic settings, particularly with rapid data updates. Tools like AutoCE optimize the selection of learned models based on dataset characteristics, achieving up to 27

The MCESS method uses a deep learning-based regression model to estimate cardinalities, enhancing query optimization efficiency [4]. The BitE model classifies workloads into Light and Heavy categories using database statistics, optimizing query execution more efficiently than traditional methods [62]. Despite advancements, query-driven learned cardinality estimators often struggle with out-of-distribution queries, highlighting the need for domain knowledge integration to improve accuracy and reliability [24]. The MAQO method, combining cost-based and heuristic techniques, illustrates the potential of merging learned models with traditional optimization methods to enhance query execution plans [63]. Ongoing research aims to refine these models for practical deployment in real-world systems [64, 65, 66].

5.3 Neuro-symbolic and Hybrid Approaches

Neuro-symbolic and hybrid approaches in query optimization merge symbolic reasoning with neural networks to enhance adaptability and efficiency. The FOOP optimizer employs deep reinforcement learning (DRL) to develop a data-adaptive optimizer, effectively avoiding exhaustive join order enumeration, a common challenge in traditional methods [67, 68]. This approach enhances execution efficiency while addressing the limitations of conventional optimizers, which often rely on inaccurate cost estimations.

Parallel processing capabilities further enhance query optimization by utilizing multiple cores and processing nodes, improving performance in complex queries within large datasets [69, 70, 71, 61]. The MPDP algorithm exemplifies this by efficiently pruning the search space and leveraging cluster resources, eliminating synchronization overhead. The FactorJoin method showcases the advantages of hybrid approaches in estimating join cardinalities, significantly outperforming both traditional and advanced learning-based methods by reducing estimation latency and model size while maintaining accuracy [69, 72, 73, 74]. This framework allows rapid estimation of numerous sub-plan queries, demonstrating the potential of combining symbolic reasoning with ML.

The integration of domain knowledge into learned models, as seen in CORDON, introduces differentiable constraints that enhance model accuracy and robustness [64, 36, 60]. The COOOL framework exemplifies learning-to-rank techniques in query optimization, transforming cost estimation into predicting relative cost orders, thus enhancing stability against execution latency variations [75, 76, 77, 78].

This discussion is further illustrated in Figure 6, which depicts the hierarchical structure of neuro-symbolic and hybrid approaches in query optimization. The figure categorizes methods into neuro-symbolic methods, parallel processing, and domain knowledge integration, providing examples from recent research. Despite progress, challenges remain, such as reliance on well-tuned cost models, which can impact optimizer performance. LEC query optimization minimizes expected query execution costs, while BitE enhances optimization by adapting to workload complexity [33, 62]. The exploration of neuro-symbolic and hybrid approaches in query optimization holds promise for improving adaptability, efficiency, and accuracy in DBMS. As research advances in methodologies

like Retrieval-Augmented Generation (RAG) and learned cardinality estimation, their impact on query optimization processes is anticipated to increase significantly [79, 64, 60].

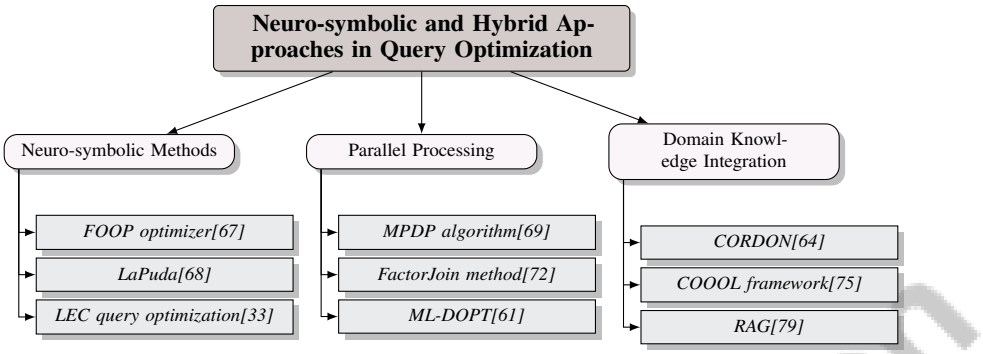


Figure 6: This figure illustrates the hierarchical structure of neuro-symbolic and hybrid approaches in query optimization, categorizing methods into neuro-symbolic methods, parallel processing, and domain knowledge integration, with examples from recent research.

5.4 Reinforcement Learning and Adaptive Optimization

Reinforcement learning (RL) has emerged as a transformative approach in adaptive query optimization, enabling DBMS to dynamically adjust performance by learning from environmental interactions. This allows databases to manage fluctuating workloads and complex data environments effectively. The MADB framework exemplifies independent learning, where multiple agents adapt their policies based on shared observations while maintaining individual goals [80]. RL applications in frameworks like Hydro introduce adaptive query processing (AQP) tailored for machine learning queries, allowing dynamic predicate evaluation reordering and improved resource allocation [3]. However, high training and inference costs associated with RL can hinder real-time applicability, necessitating more efficient learning algorithms and hybrid models combining neural and symbolic strengths [81].

Innovative methods such as FOOP leverage deep reinforcement learning to optimize join orders without exhaustive searches, significantly reducing optimization time [67]. The CardIndex method integrates learned indexing and cardinality estimation into a lightweight structure, enhancing performance and resource efficiency [82]. Learned query superoptimization improves query performance in repetitive workloads, showcasing the potential of RL and advanced AI techniques in optimizing query execution [2]. The integration of RL in adaptive query optimization represents a significant leap forward, enhancing adaptability, efficiency, and accuracy in query execution. Recent advancements allow optimizers to incrementally refine query plans based on past performance, promising improved performance and reduced latency while simplifying the optimization process [2, 83, 84, 59]. As research continues, these methodologies are expected to play a pivotal role in transforming query optimization processes, paving the way for more intelligent and responsive data systems.

6 Oracle AutoML and Automated Machine Learning

The advent of Oracle AutoML marks a significant evolution in harnessing data-driven insights, optimizing machine learning (ML) tasks for enhanced performance and efficiency across diverse applications. This section delves into Oracle AutoML's core features and its role in automating ML processes within database management systems (DBMS), highlighting both challenges and opportunities in DBMS automation. Oracle AutoML's capabilities are pivotal in managing the growing complexity and volume of data, enhancing both efficiency and adaptability [13, 16].

6.1 Introduction to Oracle AutoML

Oracle AutoML revolutionizes ML task automation in database systems by streamlining model selection, tuning, and deployment. It leverages Oracle's infrastructure to automate ML model generation, minimizing the need for extensive human expertise and addressing challenges in dynamic environ-

ments like IoT systems. This automation enhances accessibility and efficiency for organizations lacking deep technical knowledge [85, 13, 86, 5, 87].

Key features include automating feature selection and hyperparameter tuning, crucial for optimizing model performance, especially in rapidly changing data environments. Sophisticated algorithms explore parameter spaces to identify optimal configurations, accelerating model development while ensuring robustness [5, 13, 86, 61].

Oracle AutoML integrates seamlessly with Oracle's database systems, facilitating real-time analytics and decision-making by applying ML models directly to stored data. This reduces latency and enhances data security by minimizing data exportation, ensuring efficient data management [21, 13, 16].

Continuous model monitoring and updating ensure accuracy as data evolves, critical in dynamic environments requiring frequent recalibration. Advanced AI-driven approaches enable effective anomaly detection and correction within big data ecosystems [14, 88, 25]. Oracle AutoML's comprehensive framework simplifies model development and deployment, empowering organizations to leverage data fully and drive innovation [5, 13, 87].

6.2 Automation of Machine Learning Tasks

Oracle AutoML significantly automates ML tasks in database systems, enhancing model selection, hyperparameter tuning, and updating processes. This automation is vital in dynamic environments like IoT, where model adaptability and efficiency are crucial [5]. By leveraging sophisticated algorithms, Oracle AutoML optimizes configurations, enhancing predictive accuracy without extensive human input.

Automation extends to continuous model monitoring and updating, crucial in IoT environments with rapidly changing data patterns. By automating these processes, Oracle AutoML accelerates development cycles and reduces computational demands, addressing DBMS administration challenges like resource allocation and query execution optimization [87, 13, 61, 16].

Oracle AutoML's integration with Oracle's database infrastructure enables direct ML model application to stored data, enhancing real-time analytics and decision-making. This integration improves data quality and database performance, facilitating timely, data-driven decisions [14, 89, 90, 16]. Consequently, Oracle AutoML empowers organizations to harness their data's full potential while maintaining security and governance.

6.3 Integration with Database Systems

Oracle AutoML's integration into DBMS automates ML workflows, enhancing data processing efficiency and leveraging advanced ML techniques to streamline administrative tasks [13, 16]. This integration allows real-time analytics and decision-making by executing ML models directly on stored data, ensuring data security and integrity while reducing latency.

This integration facilitates automated ML model selection and optimization tailored to processed data characteristics, enhancing performance and ensuring precise insights. Techniques like multi-task training and workload forecasting allow AutoML to adapt to varying environments, streamlining operations and reducing human intervention [91, 21, 13, 61, 16].

Oracle AutoML's seamless integration with DBMS enhances model adaptability to evolving data patterns, addressing challenges like concept drift in dynamic environments. By employing multi-task training and pre-train fine-tune procedures, AutoML ensures models remain relevant without extensive retraining [5, 13, 16]. This adaptability is crucial in modern data environments, necessitating frequent recalibration for optimal performance.

Oracle AutoML's integration into DBMS enhances automation and optimization of ML workflows, addressing data complexities and volumes. This framework streamlines DBMS administration, leveraging advanced ML techniques to improve system performance and predictive capabilities, facilitating efficient data management and decision-making [21, 13, 16].

6.4 Impact on Workload Efficiency

Oracle AutoML's integration into DBMS significantly enhances workload efficiency by automating and optimizing ML processes. This reduces manual intervention in model selection, tuning, and deployment, accelerating the ML lifecycle. Advanced algorithms identify optimal configurations, enhancing model robustness and addressing challenges like concept drift in dynamic environments [85, 5].

A key impact is continuous model monitoring and updating, ensuring accuracy as data evolves, crucial in environments with rapidly changing data patterns. Automating recalibration minimizes computational demands, enhancing system efficiency and paving the way for improved database management and query optimization [87, 13, 61, 16].

Oracle AutoML's integration with database systems facilitates real-time analytics, enabling direct ML model execution within the database environment. This reduces latency and enhances operational efficiency, streamlining decision-making and mitigating data quality risks [14, 75, 78]. The seamless interaction between ML models and database infrastructure drives insights and innovation across domains.

The impact is exemplified by Meta Model-based Predictive Autoscaling (MMPA) in Alipay's cloud environment, achieving significant resource savings [38]. Oracle AutoML optimizes resource allocation and improves workload efficiency, enabling data-driven decisions and fostering innovation, particularly in dynamic environments like IoT systems [5, 61, 16].

7 Workload Forecasting

7.1 Limitations of Traditional Forecasting Techniques

Traditional forecasting methods in database management face significant challenges in effectively predicting resource usage. They are often unable to handle the complexities of multi-resource management or adapt to real-time workload fluctuations, particularly in cloud environments characterized by high variability and dimensionality. These heuristic-based approaches frequently rely on oversimplified assumptions, leading to inefficiencies such as resource wastage and Quality-of-Service (QoS) violations. In contrast, recent machine learning advancements show promise in integrating diverse system telemetry data to create models that adapt to dynamic workloads, thereby enhancing resource allocation and management strategies in cloud computing [92, 6, 93, 94].

Furthermore, traditional methods often use static models that fail to fully exploit historical data and system call sequences critical for accurate predictions. This lack of granularity and adaptability worsens their limitations, as these models are not designed to respond dynamically to rapid data pattern changes typical of modern database environments. Additionally, traditional techniques are constrained by predefined assumptions regarding data distributions and relationships, which may not reflect real-world complexities, leading to inaccurate predictions and suboptimal resource management decisions, ultimately impacting database system performance [92].

Recent research advocates for methodologies that leverage AI-driven frameworks and anomaly detection, enabling dynamic assessment and adaptation to the diverse characteristics of big data. This shift enhances forecast accuracy and reliability, highlighting the need for advanced forecasting methods that incorporate machine learning and artificial intelligence technologies to improve adaptability and precision [14, 65, 75, 95].

7.2 Introduction to QueryBot 5000

QueryBot 5000 marks a significant advancement in workload forecasting by utilizing historical data and the logical composition of queries to predict query arrival rates in dynamic environments [96]. This innovative approach enhances workload prediction robustness, addressing challenges posed by traditional techniques that struggle with modern database workload variability and complexity. Figure 7 illustrates the key aspects of QueryBot 5000, highlighting advancements in workload forecasting, complementary systems like Alibaba Workload Miner, and the impact on resource management in cloud environments.

The predictive capabilities of QueryBot 5000 derive from its ability to analyze and categorize queries based on their logical structure, enabling more accurate and context-aware forecasts. This aligns with the broader trend of employing machine learning models to improve resource management efficiency in cloud environments, as evidenced by recent advancements in prediction accuracy [94]. By integrating these models, QueryBot 5000 anticipates workload fluctuations, optimizing resource allocation and enhancing overall system performance.

Complementary systems like Alibaba Workload Miner (AWM) classify queries by business logic and identify patterns in real-time to optimize query processing [97]. This synergy between query classification and pattern discovery further enhances QueryBot 5000's ability to deliver precise workload forecasts, contributing to more efficient resource management in cloud-based database systems.

QueryBot 5000 exemplifies the transformative potential of machine learning-driven solutions in workload forecasting, offering a robust framework for managing the dynamic and complex data environments typical of modern database applications [93].

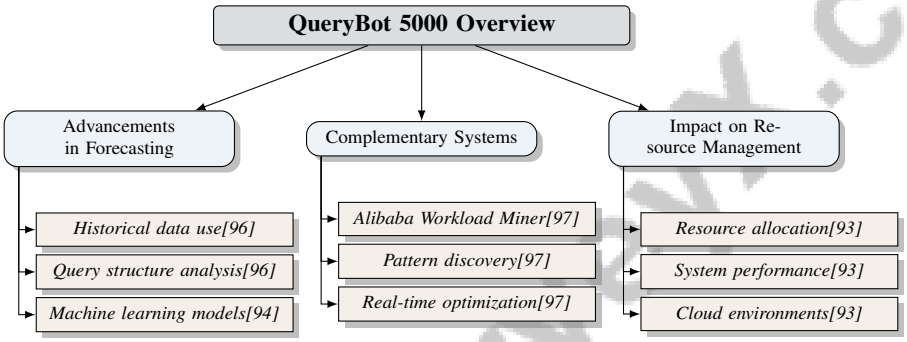


Figure 7: This figure illustrates the key aspects of QueryBot 5000, highlighting advancements in workload forecasting, complementary systems like Alibaba Workload Miner, and the impact on resource management in cloud environments.

7.3 Query Clustering and Forecasting Models

Query clustering and forecasting models significantly enhance workload prediction by categorizing similar queries and applying predictive techniques to these clusters. The QB5000 framework exemplifies this approach by predicting expected query arrival rates through clustering and forecasting [96]. This method leverages the logical composition of queries to provide more accurate workload forecasts, addressing the challenges faced by traditional forecasting techniques in dynamic database environments.

Figure 8 illustrates the hierarchical structure of query clustering and forecasting models, emphasizing key frameworks, machine learning models, and challenges associated with dynamic workload prediction and resource management in database environments. Integrating machine learning models into workload forecasting improves prediction precision and adaptability. A comprehensive survey categorizes existing research into five classes of machine learning models: Evolutionary Learning, Deep Learning, Hybrid Learning, Ensemble Learning, and Quantum Learning [94]. Each model contributes unique strengths to resource management efficiency; for instance, deep learning models capture complex patterns in query workloads, while ensemble methods enhance prediction accuracy through model combination.

Furthermore, systems like Alibaba Workload Miner (AWM) augment query clustering techniques by processing streaming query logs to discover and optimize workload patterns in real-time [97]. This capability allows for dynamic resource allocation adjustments, improving database system performance and efficiency. By merging query clustering with advanced forecasting models, these approaches offer robust solutions for managing the variability and complexity of modern database workloads.

The integration of query clustering and advanced forecasting models marks a pivotal evolution in database management systems (DBMS), significantly enhancing resource allocation accuracy and sys-

tem performance. Frameworks like QueryBot 5000 leverage historical query data to anticipate future workloads, enabling timely optimizations. Focusing on the logical composition of queries rather than mere resource utilization allows these models to adapt effectively to dynamic environments, offering both short- and long-term forecasting capabilities. The application of machine learning techniques to optimize query performance, particularly through parallelism adjustments, underscores the growing importance of proactive resource management in preventing issues such as underutilization and QoS violations in cloud environments [96, 93, 61]. As research progresses, these techniques will increasingly optimize workload forecasting and enhance the efficiency of data-driven applications.

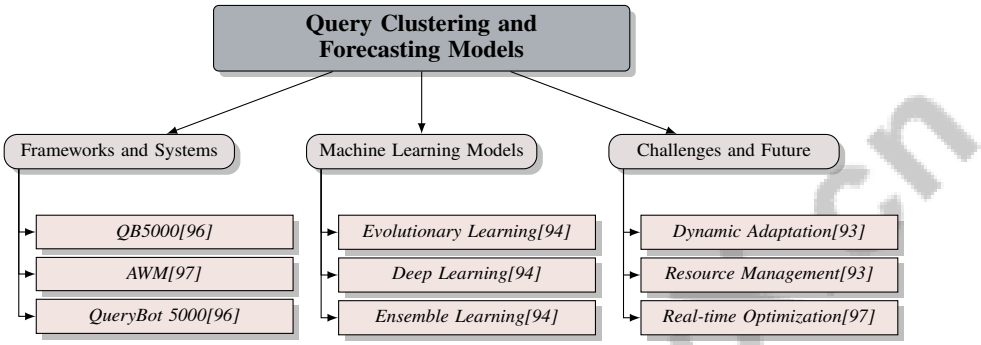


Figure 8: This figure illustrates the hierarchical structure of query clustering and forecasting models, emphasizing key frameworks, machine learning models, and challenges associated with dynamic workload prediction and resource management in database environments.

7.4 Evaluation and Real-World Application

Evaluating workload forecasting techniques, particularly in real-world applications, underscores their critical role in enhancing the efficiency and adaptability of database management systems. The implementation of QueryBot 5000 has demonstrated significant efficacy in predicting future workloads with minimal accuracy loss, allowing self-driving database management systems (DBMSs) to proactively optimize performance [96]. This predictive capability facilitates dynamic resource allocation and improves system responsiveness to fluctuating workloads in modern data environments.

A comprehensive analysis of workload forecasting techniques reveals key insights into existing research gaps and emphasizes the importance of accurate workload predictions [93]. These insights underscore the necessity for integrated approaches that consider multiple resources, enhancing overall resource management effectiveness. By leveraging advanced machine learning models, such as Quantum Neural Networks, future research can further optimize these forecasting techniques, yielding more reliable and interpretable predictions [94].

In real-world applications, integrating workload forecasting models into DBMSs enables informed decision-making processes, allowing systems to anticipate and adapt efficiently to workload changes. This adaptability is crucial for maintaining optimal performance in cloud-based environments, where high variability and complexity are prevalent. By optimizing machine learning models and incorporating Explainable AI principles, future developments can enhance the reliability and interpretability of workload forecasts, driving innovation and efficiency in data-driven applications [94].

The evaluation of workload forecasting techniques in real-world scenarios reveals their significant potential to enhance DBMS efficiency through proactive optimizations. Advanced frameworks like QueryBot 5000 leverage historical data to predict query arrival rates and adapt to dynamic workloads, minimizing human intervention and reducing database administrators' tuning time. This approach anticipates future workload needs while integrating behavior modeling and action planning to facilitate autonomous system management, ultimately leading to improved performance and resource utilization in modern data-driven applications [91, 96]. As research advances, these techniques are expected to play an increasingly pivotal role in enhancing the adaptability and efficiency of modern data environments, ensuring systems effectively manage contemporary workload demands.

8 Anomaly Detection

8.1 Introduction to Anomaly Detection in Databases

Anomaly detection is crucial in database management systems for identifying irregularities that may indicate system malfunctions, security breaches, or data integrity issues. The complexity of modern data environments, characterized by high dimensionality and diverse data types, necessitates advanced methodologies capable of addressing various data anomalies. Techniques such as statistical characterization and machine learning are increasingly relied upon to extract insights from atypical data points. Recent advancements, including the use of Large Language Models (LLMs) for encoding non-semantic data and the development of a taxonomy for log data anomalies, highlight the need for tailored approaches to enhance detection capabilities across diverse datasets. These methodologies improve the reliability of critical applications, such as financial audits and IT operations, by tackling challenges like feature sparsity and the intricacies of heterogeneous data [98, 99, 100, 95].

The integration of Knowledge-Based Temporal Abstraction (KBTA) with temporal pattern mining represents a significant advancement, enabling the identification of normal behavior patterns and facilitating the detection of deviations that suggest anomalies [100]. This approach is particularly effective in environments where temporal data is crucial for understanding system behavior.

In log data contexts, a novel taxonomy categorizes anomalies into point anomalies, such as Template and Attribute Anomalies, and contextual anomalies, providing a structured framework for understanding and addressing anomalies in complex IT systems [98, 101]. Anomalies in computational workflows can indicate underlying hardware or system process issues, necessitating sophisticated detection techniques that navigate feature dimension heterogeneity and sparsity [85, 99].

At an industrial scale, anomaly detection methods effectively identify cases arising from different mechanisms compared to the majority of the dataset, demonstrating their capability in large-scale environments [102]. By interpreting atypical data within extensive datasets, these methods address challenges often neglected in traditional statistical analyses [95].

The advancement of sophisticated anomaly detection techniques is essential for enhancing the reliability and security of database systems. Techniques such as those utilized in PinSQL effectively identify and address anomalies in database queries and performance metrics, thereby improving the integrity and operational efficiency of data-driven applications. This proactive approach mitigates performance issues stemming from anomalous SQL queries and fosters the development of resilient and intelligent database management solutions [103, 95, 16].

8.2 Techniques and Frameworks for Anomaly Detection

Anomaly detection in database management systems is vital for identifying irregularities that may signal issues like security breaches or system malfunctions. Various innovative techniques and frameworks have emerged to address the complexities of detecting anomalies in dynamic data environments. These include Knowledge-Based Temporal Abstraction (KBTA) for time-oriented data, Large Language Models (LLMs) for encoding financial data, and comprehensive taxonomies for log data anomalies, all aimed at enhancing detection accuracy and efficiency [98, 99, 95, 100, 85].

KBTA combined with temporal pattern mining exemplifies a technique that detects anomalies in time-oriented data by identifying normal behavior patterns [100]. This approach is particularly effective in environments where understanding temporal data patterns is crucial.

The ADLILog framework represents an unsupervised log-based anomaly detection method that utilizes log instructions from public code projects alongside target system logs to train a deep learning model [101]. This method effectively addresses the challenge of log data anomalies without requiring labeled data, which is often resource-intensive to produce [98].

Pre-trained sentence-transformer models encode non-semantic categorical data into fixed-size dense vectors, facilitating the detection of anomalies in complex datasets such as financial journal entries [99]. Traditional rule-based systems and machine learning techniques often necessitate extensive data preprocessing and expert knowledge, limiting their effectiveness in detecting novel anomalies [85]. Scalable approaches are increasingly adopted to handle heterogeneous data types, enhancing the robustness and scalability of anomaly detection systems [102].

Certain anomaly detection methods leverage the principle that atypical data can be described with fewer bits than typical data, offering a unique perspective on identifying irregularities [95]. The advancement of sophisticated techniques and frameworks for anomaly detection is crucial for ensuring the reliability and security of database systems, enabling the identification of atypical data patterns that may indicate potential threats. The integration of statistical methods, machine learning algorithms, and LLMs enhances the accuracy of anomaly detection while supporting proactive measures against data breaches and fraud. Domain-specific knowledge and temporal pattern mining further refine the detection process, allowing for a nuanced understanding of normal versus abnormal behavior in complex data environments [104, 98, 99, 95, 100].

8.3 Anomaly Detection in Log Data

Anomaly detection in log data is essential for identifying irregularities that may indicate issues such as security breaches or system malfunctions. The complexity of contemporary data environments, characterized by high-dimensional and diverse log data, necessitates advanced methodologies capable of efficiently processing and analyzing this complexity, including feature mapping, kernel methods, and deep learning-based approaches [105, 104, 98, 95, 75].

The ADLILog framework exemplifies an unsupervised log-based anomaly detection method that processes raw logs through a two-phase learning procedure, leveraging log instructions from public code projects alongside target system logs to train a deep learning model [101]. This method effectively addresses the challenge of detecting anomalies in log data without requiring labeled data.

Evaluating anomaly detection methods typically involves utilizing real-world data to derive normal patterns and detect deviations. For example, Shabtai et al. conducted an evaluation using data from a real server over two weeks, with the first week dedicated to deriving normal patterns and the second week for anomaly detection [100]. This underscores the importance of realistic data environments in assessing the effectiveness of anomaly detection techniques.

Deep learning-based methods, such as DeepLog and A2Log, have demonstrated superior performance in detecting anomalies compared to traditional data mining techniques, excelling across all types of anomalies, particularly template anomalies [98]. The comparative analysis reveals the potential of deep learning models to provide more accurate and reliable anomaly detection in log data.

In industrial applications, the computational intensity of processes like Singular Value Decomposition (SVD) presents challenges for scaling with large datasets [102]. Addressing scalability issues is crucial for implementing effective anomaly detection systems in expansive environments.

Innovative methods that define atypicality axiomatically offer a fresh perspective on anomaly detection, distinguishing themselves from traditional techniques reliant on likelihood measures [95].

As illustrated in Figure 9, the hierarchical categorization of anomaly detection in log data encompasses various detection methods, evaluation techniques, and the associated challenges and innovations. This framework enhances our understanding of the landscape of anomaly detection, emphasizing the need for tailored approaches to address specific challenges in log data analysis.

The advancement of sophisticated anomaly detection techniques in log data is vital for ensuring the reliability and security of database systems, enabling IT operators to identify and classify various types of anomalies effectively. A comprehensive understanding of these anomalies, supported by a newly introduced taxonomy for log data, guides the selection of appropriate detection algorithms. Recent studies indicate that deep learning approaches significantly outperform traditional data mining methods, particularly in detecting contextual anomalies, thereby enhancing log data analysis efficacy and improving system performance and security [98, 104]. By effectively identifying and addressing anomalies, these techniques ensure the integrity and performance of data-driven applications, paving the way for resilient and intelligent database management solutions.

8.4 AI Techniques for Anomaly Detection in Financial and Industrial Data

The application of artificial intelligence (AI) techniques for anomaly detection in financial and industrial data has shown substantial promise in enhancing the accuracy and efficiency of identifying irregularities indicative of fraud, operational failures, or system inefficiencies. In financial contexts, the use of large language models (LLMs) for embedding financial data has significantly improved

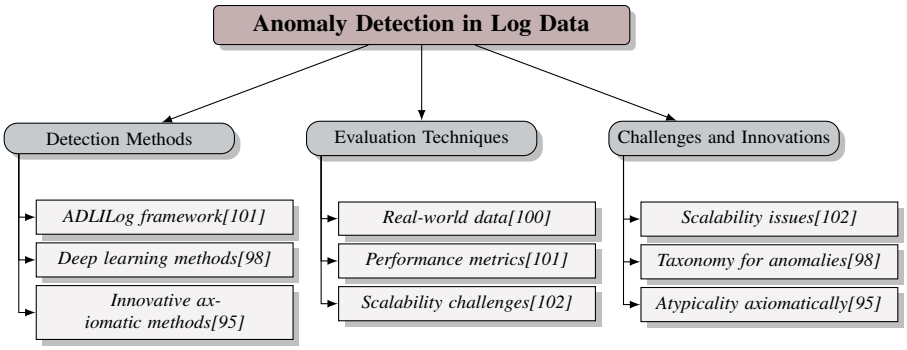


Figure 9: This figure illustrates the hierarchical categorization of anomaly detection in log data, focusing on detection methods, evaluation techniques, and associated challenges and innovations.

anomaly detection capabilities, capturing complex patterns and relationships within the data and outperforming traditional methods [99].

In industrial settings, integrating AI-driven methodologies into anomaly detection frameworks has effectively identified atypical patterns within large datasets. Techniques such as Knowledge-Based Temporal Abstraction (KBTA) combined with temporal pattern mining have successfully detected anomalies by identifying deviations from established normal patterns [100]. This approach is particularly valuable in environments where temporal data is critical for understanding operational behaviors and detecting potential disruptions.

Advanced AI techniques have enhanced the use of log data for anomaly detection in both financial and industrial domains. The ADLILog framework leverages log instructions from public code projects to train deep learning models, significantly outperforming both supervised and unsupervised methods in terms of F1 score [101]. This framework addresses the challenge of detecting anomalies in log data by utilizing deep learning approaches that eliminate the need for costly labeled data, thus enhancing scalability and applicability.

Moreover, applying LLMs in analyzing log files generated during computational workflows offers a novel approach to anomaly detection. By adapting LLMs through supervised fine-tuning and in-context learning, these models can directly detect anomalies, providing a powerful tool for identifying irregularities in complex data environments [85].

Future research in anomaly detection should focus on refining classification methods, exploring additional types of anomalies, and improving the understanding of currently unclassifiable log messages [98]. Furthermore, developing methods that successfully identify atypical subsequences in large datasets lays a solid theoretical foundation for future applications in data analysis, ensuring that AI-driven anomaly detection techniques evolve and adapt to modern data complexities [95].

Integrating AI techniques into anomaly detection frameworks significantly transforms the management of financial and industrial data, enhancing precision and effectiveness in anomaly identification. Recent studies demonstrate that leveraging LLMs for encoding non-semantic financial data improves the detection of irregularities in financial records, outperforming traditional machine learning models. Furthermore, implementing AI-driven frameworks addresses critical data quality challenges in big data ecosystems by employing advanced methodologies for error detection, correction, and metadata integration. This comprehensive approach facilitates the identification of diverse quality anomalies and enhances overall data reliability, ultimately leading to more informed decision-making across various industries [14, 95, 99]. As research progresses in this field, these methodologies are expected to play an increasingly pivotal role in ensuring the reliability and security of data-driven applications across various domains.

9 Self-Tuning Databases

9.1 Frameworks and Methods for Self-Tuning

Self-tuning databases represent a paradigm shift in database management, utilizing advanced architectures and machine learning for autonomous configuration adjustments. Unlike traditional systems reliant on human intervention, modern self-driving databases like Peloton employ predictive planning to optimize current workloads and anticipate future demands. Innovations such as OnlineTune and QueryBot 5000 enhance performance by adapting to dynamic cloud environments and utilizing historical query data for workload forecasting, respectively, thereby reducing manual tuning efforts [21, 96, 42, 39]. These systems leverage machine learning algorithms and optimization techniques to dynamically accommodate changing workloads and system requirements.

A significant method in self-tuning databases is the TGD-rewrite technique, which optimizes query generation through factorization and rewriting, minimizing redundancy and ensuring efficient query processing [106]. Integrating anomaly detection techniques is crucial for maintaining optimal performance, as demonstrated by Caithness et al., who identify anomalies through dataset partition analysis and joint distribution evaluation [102]. Such detection prevents performance degradation and ensures efficient resource allocation.

Advanced techniques like reinforcement learning and predictive modeling forecast future workload patterns, adjusting system configurations accordingly. Frameworks such as QueryBot 5000 predict query arrival rates, enabling proactive optimization actions that reduce costly human intervention [91, 96]. These methodologies ensure databases maintain high performance and reliability in complex data landscapes.

The development of self-tuning frameworks signifies a transformative advancement in database management, yielding intelligent, adaptive, and efficient solutions for performance optimization. As research progresses, particularly with AI and machine learning integration, innovative methodologies are expected to enhance DBMS capabilities and operational efficiency. This evolution encompasses automating administrative tasks, unifying algorithms with data processing, and incorporating advanced analytics, all crucial for managing the increasing volume and complexity of data [104, 8, 16].

9.2 Resource and Query Optimization

Self-tuning databases employ sophisticated methodologies to optimize resource allocation and query performance, ensuring effective data management. They utilize machine learning models and adaptive algorithms to adjust system configurations dynamically in response to shifting workloads and data patterns. Generative models capturing the joint probability distribution of datasets enable accurate query estimates, enhancing query optimization for diverse and complex queries [107].

Beyond query optimization, self-tuning databases predict future workload patterns to optimize resource allocation, supported by anomaly detection techniques that identify deviations from normal behavior. Analyzing dataset partitions and evaluating joint distributions aids in detecting anomalies, ensuring efficient resource allocation [102]. Future research aims to refine these optimization techniques by enhancing SVD implementations and exploring their statistical properties. Bootstrapped resampling techniques are anticipated to improve the robustness and accuracy of resource and query optimization processes [102]. These advancements will contribute to developing more intelligent and adaptive self-tuning databases capable of managing resources and optimizing query performance in dynamic environments.

9.3 Anomaly and Atypicality Detection

Anomaly and atypicality detection are critical for optimizing self-tuning databases, identifying irregularities that may indicate system malfunctions or security breaches. By focusing on atypical data, which can be represented more efficiently than typical data, these methods enhance the understanding of complex datasets. Recent advancements, including large language models (LLMs), improve the recognition of novel anomalies in computational workflows, facilitating effective monitoring and maintenance of system reliability and security. Techniques such as supervised fine-tuning and in-context learning enable sophisticated pattern recognition for anomaly detection [85, 95]. These

mechanisms are integral to the adaptive nature of self-tuning databases, allowing proactive resolution of performance bottlenecks.

In self-tuning databases, anomaly detection frameworks continuously monitor system behavior, identifying deviations from established patterns that may indicate underlying issues. Advanced machine learning techniques, such as Knowledge-Based Temporal Abstraction (KBTA) combined with temporal pattern mining, enhance the ability to detect anomalies in time-oriented data [100]. Utilizing log data for anomaly detection is essential for maintaining database performance. Frameworks like ADLILog leverage log instructions from public code projects to train deep learning models, providing an unsupervised method for detecting anomalies in log data without costly labeled data [101].

The application of LLMs for embedding financial data significantly improves anomaly detection, offering nuanced insights into atypical patterns within datasets [99]. These embeddings capture complex relationships, enhancing self-tuning databases' ability to detect and respond to anomalies effectively.

Integrating anomaly and atypicality detection mechanisms into self-tuning databases is crucial for maintaining performance and ensuring the reliability and security of data-driven applications. By effectively identifying and addressing anomalies, these systems optimize resource allocation, prevent performance degradation, and enhance efficiency. As research progresses, innovative methodologies are expected to play a vital role in the evolution of intelligent and adaptive database management systems. This includes developing self-driving architectures like Peloton, which leverage machine learning for system optimization and workload trend anticipation. Learning-based approaches for database knob tuning and learned indexes, such as ALEX, promise to enhance performance by efficiently managing configurations and optimizing retrieval processes. Collectively, these methodologies aim to reduce human intervention, improve resource utilization, and address modern data management complexities, paving the way for more autonomous and efficient database systems [21, 108, 109, 12, 60].

10 Automated Database Administration

10.1 Automation Techniques and Tools

The evolution of automated database administration is marked by the integration of AI and ML, significantly enhancing system efficiency and reducing manual oversight. AI-driven models such as PUNQ and algorithms like GenCrd advance cardinality estimation, streamlining query optimization and improving performance [110]. The AirIndex framework illustrates automated index optimization through guided graph search, enhancing data retrieval efficiency without manual intervention [50].

Learning-based methodologies, as seen in the openGauss system, dynamically adjust to workload fluctuations, minimizing manual tuning and ensuring high performance [45]. In IoT analytics, AutoML automates ML processes, crucial for managing large, dynamic data volumes [5]. Automated tuning methods, including knob tuning, further reduce the expertise required for configuration adjustments, maintaining optimal performance [109]. The Baihe framework exemplifies the integration of ML into database systems, enhancing automation while ensuring system stability [46].

Moreover, applying relational algebra in process automation reduces execution time and resource usage, improving system efficiency [111]. These advancements represent a leap in database management, leveraging AI and ML for intelligent, adaptive solutions that minimize human intervention. As databases grow in complexity, automated systems optimize operations by predicting workload trends and autonomously managing tasks, addressing traditional methods' limitations [21, 8, 12, 16]. Continued research is poised to further enhance modern database systems' capabilities.

10.2 Unified and Framework-Based Approaches

Unified and framework-based methodologies are pivotal in advancing automated database administration, offering integrated solutions that enhance DBMS efficiency and scalability. By leveraging AI and ML, these approaches address the growing complexity and data volumes managed by DBMS, fostering self-managing systems that reduce manual intervention [16, 13, 12, 8, 14].

The Baihe framework exemplifies AI integration into DBMS, enhancing automation capabilities while maintaining robustness [46]. Framework-based approaches, such as the Planter framework,

emphasize modular architectures that support integrating new technologies, vital for managing diverse workloads [40].

Relational algebra application in process automation further streamlines operations by reducing execution time and resource usage [111]. These unified approaches represent a critical evolution in database administration, leveraging AI and ML to deliver adaptive solutions that enhance performance with minimal manual input. As DBMSs evolve, adopting these solutions is essential for overcoming administrative challenges and unlocking new data management opportunities [8, 16].

10.3 Challenges and Human Factors

The shift to automated database systems presents challenges, particularly concerning dynamic workloads complicating VM consolidation and energy efficiency predictions [6]. Human factors, including user skepticism and lack of experience, significantly impact automation adoption, necessitating adequate training and confidence-building measures [16].

Resistance to change is another hurdle, as stakeholders may be reluctant to relinquish control over manual processes. Effective communication of automation's benefits, such as operational efficiency and error reduction, is crucial for securing user buy-in [7, 79, 16, 14, 87].

Addressing these technical and human challenges is vital for successful automation implementation, significantly improving database management processes' adaptability and efficiency. Integrating advanced AI and ML technologies can further optimize these processes, enabling proactive data management essential for accurate analysis and informed decision-making [21, 8, 16, 14, 60].

11 Conclusion

The integration of artificial intelligence (AI) and machine learning (ML) into database management systems (DBMS) marks a significant advancement in data processing capabilities, enhancing performance, scalability, and adaptability. AI methodologies have been instrumental in optimizing key DBMS components, including query processing, resource allocation, and anomaly detection. Innovations such as learned indexes exemplify AI's potential to transform data management, offering improvements in query latency and space efficiency. Current AI-driven database systems, like openGauss, showcase high performance through innovative techniques and architectural designs that effectively navigate complex data environments.

Significant progress has been achieved in embedding AI techniques within database systems to enhance their performance and efficiency, particularly through hybrid systems that integrate AI with traditional methodologies. The concept of self-driving DBMS architectures illustrates the feasibility of autonomous systems that minimize human intervention while enhancing operational performance. Moreover, the application of reinforcement learning in frameworks has markedly improved resource management, demonstrating AI's capability in optimizing resource allocation and utilization.

Future research should focus on expanding the application of ML-enhanced models and exploring dynamic datasets to refine optimization processes further. Enhancing models to leverage database values and foreign-key relationships presents a promising development path. Additionally, building user trust in automation tools and providing adequate training for DBMS administration automation are critical areas for further exploration. The reshaping of ontologies has been shown to simplify queries and enhance the efficiency of Knowledge Graph generation, which is vital for optimizing industrial analytics.

The integration of AI into DBMS offers a promising pathway for enhancing efficiency, scalability, and adaptability. As research progresses, these methodologies are set to play an increasingly crucial role in shaping the future of data-driven applications, leading to more responsive and intelligent database solutions. The ongoing evolution of these technologies is expected to address the challenges posed by modern data environments, ensuring that DBMS can effectively manage contemporary workload demands.

References

[1] Alekh Jindal, Lalitha Viswanathan, and Konstantinos Karanasos. Query and resource optimizations: A case for breaking the wall in big data systems, 2019.

[2] Ryan Marcus. Learned query superoptimization, 2023.

[3] Gaurav Tarlok Kakkar, Jiashen Cao, Aubhro Sengupta, Joy Arulraj, and Hyesoon Kim. Hydro: Adaptive query processing of ml queries, 2024.

[4] Yaoshu Wang, Chuan Xiao, Jianbin Qin, Xin Cao, Yifang Sun, Wei Wang, and Makoto Onizuka. Monotonic cardinality estimation of similarity selection: A deep learning approach, 2021.

[5] Li Yang and Abdallah Shami. Iot data analytics in dynamic environments: From an automated machine learning perspective, 2022.

[6] Tahseen Khan, Wenhong Tian, Guangyao Zhou, Shashikant Ilager, Mingming Gong, and Rajkumar Buyya. Machine learning (ml)-centric resource management in cloud computing: A review and future directions. *Journal of Network and Computer Applications*, 204:103405, 2022.

[7] Jun-Peng Zhu, Peng Cai, Kai Xu, Li Li, Yishen Sun, Shuai Zhou, Haihuang Su, Liu Tang, and Qi Liu. Autotqa: Towards autonomous tabular question answering through multi-agent large language models. *Proceedings of the VLDB Endowment*, 17(12):3920–3933, 2024.

[8] Jim Gray. The revolution in database system architecture, 2004.

[9] Zeyan Li, Nengwen Zhao, Shenglin Zhang, Yongqian Sun, Pengfei Chen, Xidao Wen, Minghua Ma, and Dan Pei. Constructing large-scale real-world benchmark datasets for aiops, 2022.

[10] Zhuoxun Zheng, Baifan Zhou, Dongzhuoran Zhou, Gong Cheng, Ernesto Jiménez-Ruiz, Ahmet Soylu, and Evgeny Kharlamo. Query-based industrial analytics over knowledge graphs with ontology reshaping, 2022.

[11] Youcef Remil. *A data mining perspective on explainable AIOps with applications to software maintenance*. PhD thesis, INSA de Lyon, 2023.

[12] Xuanhe Zhou, Chengliang Chai, Guoliang Li, and Ji Sun. Database meets artificial intelligence: A survey. *IEEE Transactions on Knowledge and Data Engineering*, 34(3):1096–1116, 2020.

[13] Ziniu Wu, Pei Yu, Peilun Yang, Rong Zhu, Yuxing Han, Yaliang Li, Defu Lian, Kai Zeng, and Jingren Zhou. A unified transferable model for ml-enhanced dbms, 2021.

[14] Widad Elouataoui. Ai-driven frameworks for enhancing data quality in big data ecosystems: Error$_d etection, correction, and metadata integration$, 2024.

[15] Christopher J. Hazard, Christopher Fusting, Michael Resnick, Michael Auerbach, Michael Meehan, and Valeri Korobov. Natively interpretable machine learning and artificial intelligence: Preliminary results and future directions, 2019.

[16] Yifan Wang, Pierre Bourhis, Romain Rouvoy, and Patrick Royer. Challenges & opportunities in automating dbms: A qualitative study. In *Proceedings of the 39th IEEE/ACM International Conference on Automated Software Engineering*, pages 2013–2023, 2024.

[17] Aditi Singh, Akash Shetty, Abul Ehtesham, Saket Kumar, and Tala Talaei Khoei. A survey of large language model-based generative ai for text-to-sql: Benchmarks, applications, use cases, and challenges, 2025.

[18] Immanuel Trummer and Christoph Koch. Approximation schemes for many-objective query optimization, 2014.

[19] Beng Chin Ooi, Shaofeng Cai, Gang Chen, Yanyan Shen, Kian-Lee Tan, Yuncheng Wu, Xiaokui Xiao, Naili Xing, Cong Yue, Lingze Zeng, et al. Neurdb: an ai-powered autonomous data system. *Science China Information Sciences*, 67(10):200901, 2024.

[20] Phanwadee Sinthong and Michael J. Carey. Polyframe: A retargetable query-based approach to scaling dataframes (extended version), 2021.

[21] Andrew Pavlo, Gustavo Angulo, Joy Arulraj, Haibin Lin, Jiexi Lin, Lin Ma, Prashanth Menon, Todd C Mowry, Matthew Perron, Ian Quah, et al. Self-driving database management systems. In *CIDR*, volume 4, page 1, 2017.

[22] Giorgio Vinciguerra, Paolo Ferragina, and Michele Miccinesi. Superseding traditional indexes by orchestrating learning and geometry, 2019.

[23] Tim Kraska, Alex Beutel, Ed H. Chi, Jeffrey Dean, and Neoklis Polyzotis. The case for learned index structures, 2018.

[24] Xianghong Xu, Zhibing Zhao, Tieying Zhang, Rong Kang, Luming Sun, and Jianjun Chen. Coool: A learning-to-rank approach for sql hint recommendations, 2023.

[25] Ming Sheng, Shuliang Wang, Yong Zhang, Kaige Wang, Jingyi Wang, Yi Luo, and Rui Hao. Mqrld: A multimodal data retrieval platform with query-aware feature representation and learned index based on data lake, 2025.

[26] Guo-Liang Li, Jiang Wang, and Guo Chen. opengauss: An enterprise-grade open-source database system. *Journal of Computer Science and Technology*, 39(5):1007–1028, 2024.

[27] Oliver Schulte, Hassan Khosravi, Flavia Moser, and Martin Ester. Learning class-level bayes nets for relational data, 2009.

[28] Yao Tian, Tingyun Yan, Xi Zhao, Kai Huang, and Xiaofang Zhou. A learned index for exact similarity search in metric spaces, 2022.

[29] Ryan Marcus, Andreas Kipf, Alexander van Renen, Mihail Stoian, Sanchit Misra, Alfons Kemper, Thomas Neumann, and Tim Kraska. Benchmarking learned indexes, 2020.

[30] Yuxing Han, Haoyu Wang, Lixiang Chen, Yifeng Dong, Xing Chen, Benquan Yu, Chengcheng Yang, and Weining Qian. Bytecard: Enhancing data warehousing with learned cardinality estimation. *arXiv preprint arXiv:2403.16110*, 2024.

[31] Claude Lehmann, Pavel Sulimov, and Kurt Stockinger. Is your learned query optimizer behaving as you expect? a machine learning perspective, 2024.

[32] Domenico Amato, Giosue' Lo Bosco, and Raffaele Giancarlo. On the suitability of neural networks as building blocks for the design of efficient learned indexes, 2022.

[33] Francis C. Chu, Joseph Y. Halpern, and Praveen Seshadri. Least expected cost query optimization: an exercise in utility, 1999.

[34] Rohit Raghunathan, Sushovan De, and Subbarao Kambhampati. Bayes networks for supporting query processing over incomplete autonomous databases, 2012.

[35] Hai Lan, Zhifeng Bao, J. Shane Culpepper, Renata Borovica-Gajic, and Yu Dong. A simple yet high-performing on-disk learned index: Can we have our cake and eat it too?, 2023.

[36] Prasan Roy, S. Seshadri, S. Sudarshan, and Siddhesh Bhobe. Efficient and extensible algorithms for multi query optimization, 1999.

[37] Wolfgang Gatterbauer and Dan Suciu. Dissociation and propagation for approximate lifted inference with standard relational database management systems, 2016.

[38] Siqiao Xue, Chao Qu, Xiaoming Shi, Cong Liao, Shiyi Zhu, Xiaoyu Tan, Lintao Ma, Shiyu Wang, Shijun Wang, Yun Hu, Lei Lei, Yangfei Zheng, Jianguo Li, and James Zhang. A meta reinforcement learning approach for predictive autoscaling in the cloud, 2022.

[39] Xinyi Zhang, Hong Wu, Yang Li, Jian Tan, Feifei Li, and Bin Cui. Towards dynamic and safe configuration tuning for cloud databases. In *Proceedings of the 2022 International Conference on Management of Data*, pages 631–645, 2022.

[40] Changgang Zheng, Mingyuan Zang, Xinpeng Hong, Riyad Bensoussane, Shay Vargaftik, Yaniv Ben-Itzhak, and Noa Zilberman. Automating in-network machine learning, 2022.

[41] Bowen Wu, Dimitrios Koutsoukos, and Gustavo Alonso. Efficiently processing joins and grouped aggregations on gpus, 2025.

[42] Jialin Ding. *Instance-Optimized Database Indexes and Storage Layouts*. PhD thesis, Massachusetts Institute of Technology, 2022.

[43] Jason Mohoney, Anil Pacaci, Shihabur Rahman Chowdhury, Ali Mousavi, Ihab F. Ilyas, Umar Farooq Minhas, Jeffrey Pound, and Theodoros Rekatsinas. High-throughput vector similarity search in knowledge graphs, 2023.

[44] Zihao Zhao, Zhihong Shen, Mingjie Tang, Chuan Hu, Huajin Wang, and Yuanchun Zhou. Pandadb: Understanding unstructured data in graph database, 2022.

[45] Guoliang Li, Xuanhe Zhou, Ji Sun, Xiang Yu, Yue Han, Lianyuan Jin, Wenbo Li, Tianqing Wang, and Shifu Li. opengauss: An autonomous database system. *Proceedings of the VLDB Endowment*, 14(12):3028–3042, 2021.

[46] Andreas Pfadler, Rong Zhu, Wei Chen, Botong Huang, Tianjing Zeng, Bolin Ding, and Jingren Zhou. Baihe: Sysml framework for ai-driven databases, 2021.

[47] Chuzhe Tang, Zhiyuan Dong, Minjie Wang, Zhaoguo Wang, and Haibo Chen. Learned indexes for dynamic workloads, 2019.

[48] Abdullah Al-Mamun, Hao Wu, Qiyang He, Jianguo Wang, and Walid G. Aref. A survey of learned indexes for the multi-dimensional space, 2024.

[49] Luis Croquevielle, Guang Yang, Liang Liang, Ali Hadian, and Thomas Heinis. Querying in constant expected time with learned indexes, 2024.

[50] Supawit Chockchowwat, Wenjie Liu, and Yongjoo Park. Airindex: Versatile index tuning through data and storage, 2023.

[51] Alireza Heidari, Amirhossein Ahmadi, and Wei Zhang. Uplif: An updatable self-tuning learned index framework, 2024.

[52] Sachith Pai, Michael Mathioudakis, and Yanhao Wang. Wazi: A learned and workload-aware z-index, 2024.

[53] Minsu Kim, Jinwoo Hwang, Guseul Heo, Seiyeon Cho, Divya Mahajan, and Jongse Park. Accelerating string-key learned index structures via memoization-based incremental training, 2024.

[54] Hussam Abu-Libdeh, Deniz Altınbüken, Alex Beutel, Ed H. Chi, Lyric Doshi, Tim Kraska, Xiaozhou, Li, Andy Ly, and Christopher Olston. Learned indexes for a google-scale disk-based database, 2020.

[55] Supawit Chockchowwat, Wenjie Liu, and Yongjoo Park. Automatically finding optimal index structure, 2022.

[56] Ali Hadian and Thomas Heinis. Shift-table: A low-latency learned index for range queries using model correction, 2021.

[57] Hai Lan, Zhifeng Bao, J. Shane Culpepper, and Renata Borovica-Gajic. Updatable learned indexes meet disk-resident dbms – from evaluations to design choices, 2023.

[58] Yaliang Li, Daoyuan Chen, Bolin Ding, Kai Zeng, and Jingren Zhou. A pluggable learned index method via sampling and gap insertion, 2021.

[59] Peter Akioyamen, Zixuan Yi, and Ryan Marcus. The unreasonable effectiveness of llms for query optimization, 2024.

[60] Jens Dörpinghaus and Andreas Stefan. Optimization of retrieval algorithms on large scale knowledge graphs, 2020.

[61] Zhiwei Fan, Rathijit Sen, Paraschos Koutris, and Aws Albarghouthi. A comparative exploration of ml techniques for tuning query degree of parallelism, 2020.

[62] Yuri Kim, Yewon Choi, Yujung Gil, Sanghee Lee, Heesik Shin, and Jaehyok Chong. Bite : Accelerating learned query optimization in a mixed-workload environment, 2023.

[63] Debajyoti Mukhopadhyay, Dhaval Chandarana, Rutvi Dave, Sharyu Page, and Shikha Gupta. Query optimization over web services using a mixed approach, 2012.

[64] Peizhi Wu, Ryan Marcus, and Zachary G. Ives. Adding domain knowledge to query-driven learned databases, 2023.

[65] Xiaoying Wang, Changbo Qu, Weiyuan Wu, Jiannan Wang, and Qingqing Zhou. Are we ready for learned cardinality estimation?, 2021.

[66] Rojeh Hayek and Oded Shmueli. Improved cardinality estimation by learning queries containment rates, 2019.

[67] Jonas Heitz and Kurt Stockinger. Join query optimization with deep reinforcement learning algorithms, 2019.

[68] Yifan Wang, Haodi Ma, and Daisy Zhe Wang. No more optimization rules: Llm-enabled policy-based multi-modal query optimizer, 2024.

[69] Riccardo Mancini, Srinivas Karthik, Bikash Chandra, Vasilis Mageirakos, and Anastasia Ailamaki. Efficient massively parallel join optimization for large queries, 2022.

[70] K. F. D. Rietveld and H. A. G. Wijshoff. Redefining the query optimization process, 2022.

[71] Immanuel Trummer and Christoph Koch. Parallelizing query optimization on shared-nothing architectures, 2015.

[72] Ziniu Wu, Parimarjan Negi, Mohammad Alizadeh, Tim Kraska, and Samuel Madden. Factorjoin: A new cardinality estimation framework for join queries, 2022.

[73] Yuvaraj Chesetti and Prashant Pandey. Evaluating learned indexes for external-memory joins, 2024.

[74] Ibrahim Sabek and Tim Kraska. The case for learned in-memory joins, 2022.

[75] Shuai Ma and Jinpeng Huai. Approximate computation for big data analytics, 2019.

[76] Amin Kamali, Verena Kantere, Calisto Zuzarte, and Vincent Corvinelli. Roq: Robust query optimization based on a risk-aware learned cost model, 2024.

[77] Immanuel Trummer and Christoph Koch. Probably approximately optimal query optimization, 2015.

[78] Dawei Tao, Enqi Liu, Sidath Randeni Kadupitige, Michael Cahill, Alan Fekete, and Uwe Röhm. First past the post: Evaluating query optimization in mongodb, 2024.

[79] Julien Pierre Edmond Ghali, Kosuke Shima, Koichi Moriyama, Atsuko Mutoh, and Nobuhiro Inuzuka. Enhancing retrieval processes for language generation with augmented queries, 2024.

[80] Chi Zhang, Olga Papaemmanouil, Josiah P. Hanna, and Aditya Akella. Multi-agent databases via independent learning, 2022.

[81] Yaoshu Wang, Chuan Xiao, Jianbin Qin, Rui Mao, Onizuka Makoto, Wei Wang, Rui Zhang, and Yoshiharu Ishikawa. Consistent and flexible selectivity estimation for high-dimensional data, 2021.

[82] Yingze Li, Hongzhi Wang, and Xianglong Liu. One stone, two birds: A lightweight multidimensional learned index with cardinality support, 2023.

[83] Jennifer Ortiz, Magdalena Balazinska, Johannes Gehrke, and S. Sathiya Keerthi. Learning state representations for query optimization with deep reinforcement learning, 2018.

[84] Ryan Marcus, Parimarjan Negi, Hongzi Mao, Nesime Tatbul, Mohammad Alizadeh, and Tim Kraska. Bao: Learning to steer query optimizers, 2020.

[85] Hongwei Jin, George Papadimitriou, Krishnan Raghavan, Pawel Zuk, Prasanna Balaprakash, Cong Wang, Anirban Mandal, and Ewa Deelman. Large language models for anomaly detection in computational workflows: from supervised fine-tuning to in-context learning, 2024.

[86] David Charte, Francisco Charte, María J. del Jesus, and Francisco Herrera. A showcase of the use of autoencoders in feature learning applications, 2020.

[87] Tianyu Cui, Shiyu Ma, Ziang Chen, Tong Xiao, Shimin Tao, Yilun Liu, Shenglin Zhang, Duoming Lin, Changchang Liu, Yuzhe Cai, Weibin Meng, Yongqian Sun, and Dan Pei. Logeval: A comprehensive benchmark suite for large language models in log analysis, 2024.

[88] Marcel Maltry and Jens Dittrich. A critical analysis of recursive model indexes, 2021.

[89] Daniel Lindner, Daniel Ritter, and Felix Naumann. Enabling data dependency-based query optimization, 2024.

[90] Xuanhe Zhou, Cheng Chen, Kunyi Li, Bingsheng He, Mian Lu, Qiaosheng Liu, Wei Huang, Guoliang Li, Zhao Zheng, and Yuqiang Chen. Febench: A benchmark for real-time relational data feature extraction. *Proceedings of the VLDB Endowment*, 16(12):3597–3609, 2023.

[91] Lin Ma. *Self-Driving Database Management Systems: Forecasting, Modeling, and Planning*. PhD thesis, Carnegie Mellon University, 2021.

[92] Florian Schmidt, Mathias Niepert, and Felipe Huici. Representation learning for resource usage prediction, 2018.

[93] Deepika Saxena and Ashutosh Kumar Singh. workload forecasting and resource management models based on machine learning for cloud computing environments, 2021.

[94] Deepika Saxena, Jitendra Kumar, Ashutosh Kumar Singh, and Stefan Schmid. Performance analysis of machine learning centered workload prediction models for cloud, 2023.

[95] Anders Høst-Madsen, Elyas Sabeti, and Chad Walton. Data discovery and anomaly detection using atypicality: Theory, 2017.

[96] Lin Ma, Dana Van Aken, Ahmed Hefny, Gustavo Mezerhane, Andrew Pavlo, and Geoffrey J Gordon. Query-based workload forecasting for self-driving database management systems. In *Proceedings of the 2018 International Conference on Management of Data*, pages 631–645, 2018.

[97] Jiaqi Wang, Tianyi Li, Anni Wang, Xiaoze Liu, Lu Chen, Jie Chen, Jianye Liu, Junyang Wu, Feifei Li, and Yunjun Gao. Real-time workload pattern analysis for large-scale cloud databases. *arXiv preprint arXiv:2307.02626*, 2023.

[98] Thorsten Wittkopp, Philipp Wiesner, Dominik Scheinert, and Odej Kao. A taxonomy of anomalies in log data, 2021.

[99] Alexander Bakumenko, Kateřina Hlaváčková-Schindler, Claudia Plant, and Nina C. Hubig. Advancing anomaly detection: Non-semantic financial data encoding with llms, 2024.

[100] Asaf Shabtai. Anomaly detection using the knowledge-based temporal abstraction method, 2016.

[101] Jasmin Bogatinovski, Gjorgji Madjarov, Sasho Nedelkoski, Jorge Cardoso, and Odej Kao. Leveraging log instructions in log-based anomaly detection, 2022.

[102] Neil Caithness and David Wallom. Anomaly detection for industrial big data, 2018.

[103] Xiaoze Liu, Zheng Yin, Chao Zhao, Congcong Ge, Lu Chen, Yunjun Gao, Dimeng Li, Ziting Wang, Gaozhong Liang, Jian Tan, et al. Pinsql: Pinpoint root cause sqls to resolve performance issues in cloud databases. In *2022 IEEE 38th International Conference on Data Engineering (ICDE)*, pages 2549–2561. IEEE, 2022.

[104] Thibault Sellam and Martin Kersten. 80 new packages to mine database query logs, 2017.

[105] Sepanta Zeighami and Cyrus Shahabi. On distribution dependent sub-logarithmic query time of learned indexing, 2023.

[106] Georg Gottlob, Giorgio Orsi, and Andreas Pieris. Ontological queries: Rewriting and optimization (extended version), 2011.

[107] Moritz Kulessa, Alejandro Molina, Carsten Binnig, Benjamin Hilprecht, and Kristian Kersting. Model-based approximate query processing, 2018.

[108] Jialin Ding, Umar Farooq Minhas, Jia Yu, Chi Wang, Jaeyoung Do, Yinan Li, Hantian Zhang, Badrish Chandramouli, Johannes Gehrke, Donald Kossmann, David Lomet, and Tim Kraska. Alex: An updatable adaptive learned index, 2020.

[109] Xinyang Zhao, Xuanhe Zhou, and Guoliang Li. Automatic database knob tuning: a survey. *IEEE Transactions on Knowledge and Data Engineering*, 35(12):12470–12490, 2023.

[110] Rojeh Hayek and Oded Shmueli. Nn-based transformation of any sql cardinality estimator for handling distinct, and, or and not, 2020.

[111] Remco Dijkman, Juntao Gao, Paul Grefen, and Arthur ter Hofstede. Relational algebra for in-database process mining, 2017.

Disclaimer:

SurveyX is an AI-powered system designed to automate the generation of surveys. While it aims to produce high-quality, coherent, and comprehensive surveys with accurate citations, the final output is derived from the AI's synthesis of pre-processed materials, which may contain limitations or inaccuracies. As such, the generated content should not be used for academic publication or formal submissions and must be independently reviewed and verified. The developers of SurveyX do not assume responsibility for any errors or consequences arising from the use of the generated surveys.