ISSN NUMBER (ONLINE) – 2368-1209 & (PRINT) – 2691-5944

AI DBMS in modern-day applications

Barsha Rani Shaw, Biprojit Halder, Soham Sen, Sumi Basak, and Sumanta Bhattacharya

***Abstract*--The future of computing is anticipated to be shaped by the fusion of DBMS (Database Management Systems) and AI (Artificial Intelligence). This union is of utmost importance to advance DBMS technology and enable next-generation computing Although DBMS and AI systems are well-established technology, research, and development into the combination of AI and databases is still in its early phases. The availability of vast volumes of shared data for knowledge processing, effective data, knowledge management, and intelligent data processing is the driving force behind this convergence. The Intelligent Database Interface (IDI) architecture was developed to protect the significant investments made in current databases. Numerous well-liked techniques and developments for fusing AI with databases have been examined in-depth through extensive study and published articles.**

***Index Terms*--AI, DBMS, modern technology, the future of AI, AI for computing, IDI.**

I. Introduction

Over the past 50 years, both database technology and artificial intelligence (AI) have been the subject of much research. Due to its user-friendliness, declarative query style, and capacity for incorporating complex query optimization methods, databases are frequently used in a variety of applications. Recent years have seen substantial progress in AI, driven by three main elements: an abundance of data, creative algorithms, and strong computer resources. There is a chance for win-win results when AI and datasets are combined [1], [2]. AI's capacity to improve system intelligence is one of its benefits (AI4DB). When dealing with the high-performance demands of large-scale database instances and diverse applications with a large user base, particularly in cloud environments, traditional empirical methods for database optimization, such as cost estimation, join order selection, knob tweaking, index, and view selection, etc., frequently fall short.

Barsha Rani Shaw is a pre-final year student - Bachelor of Computer Application, Global College of Science and Technology (GCST) - Krishnanagar, India.

Biprojit Halder is a pre-final year student - Bachelor of Computer Application, Global College of Science and Technology (GCST) - Krishnanagar, India.

Soham Sen is a pre-final year student - Bachelor of Computer Application, Global College of Science and Technology (GCST) - Krishnanagar, India (e- mail: sohamsen710@gmail.com).

Sumi Basak is a pre-final year student - Bachelor of Computer Application, Global College of Science and Technology (GCST) - Krishnanagar, India.

Sumanta Bhattacharya is Asst. Professor & Head - Computer Application, Global College of Science and Technology (GCST) - Krishnanagar, India (e- mail: sumanta.gcst@gmail.com).

For instance, deep reinforcement learning can boost the accuracy of cost forecasts, while deep learning helps optimize database controls. The opposite is also true; database technology may improve AI algorithms (DB4AI). Developers must create intricate code and go through rigorous model training to implement AI in real-world applications. Utilizing However, this problem may be solved by utilizing learning-based strategies. database technology makes it feasible to speed up AI algorithms, lessen the difficulty of using AI models, and allow AI functionality within the database itself. Consequently, both the DB4AI and AI4DB domains have received a lot of attentionin recent studies.

II. Artificial Intelligence in DBMS

Artificial intelligence (AI) in Database Management Systems (DBMS) refers to the use of machine learning algorithms and other AI techniques to improve the usability and effectiveness of database systems. To produce optimized execution plans for upcoming queries, machine learning algorithms are employed to assess past data on query execution. This is one of the primary uses of AI in DBMS. This has the potential to significantly boost the database efficiency and query performance.

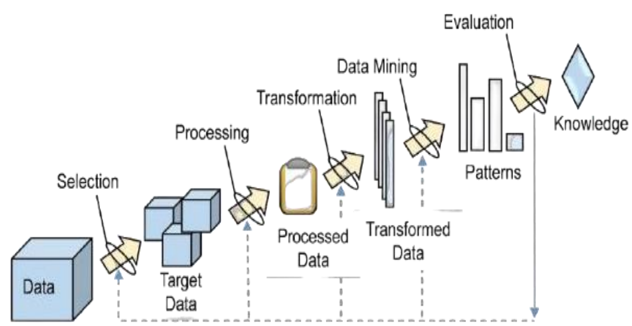

Fig: 1 Extraction of data to get knowledge for AI

III. Database Artificial Intelligence

The tuning and upkeep of the database are often handled by a database administrator (DBA), which is a traditional component of empirical specifications, techniques, and human interaction in database design [3], [4]. However, AI technology can overcome these constraints by examining a wider design space and utilizing heuristics that are more effective than human talents in solving challenging issues. The current technology may be categorized in the following ways for databases.

https://doi.org/10.15864/ajac.21012

A. Optimization using AI

Database Configuration A database administrator (DBA) must be involved in tone-tune and adapting the database settings to various circumstances due to the abundance of changeable knobs in a database. However, especially in the cloud context, it is hard for DBAs to expand their efforts to handle millions of databases. The database community has recently accepted the use of learning-based strategies to address this difficulty [5], [6], and [7]. These methods allow for the automatic adjustment of knobs and the recommendation of the best configurations of settings that produce better results than manual DBA involvement. The database system can autonomously optimize its setup by utilizing learning-based methodologies, which eliminates the scalability issues caused by human-driven knob tweaking.

B. Database Optimization

The database optimizer takes cost and cardinality estimation into account while choosing the optimum plans. Traditional methods sometimes fail to catch correlations across numerous columns or tables, which can result in inaccurate cost estimates. Recent ideas have suggested deep learning-based ways to overcome this restriction [8], [9]. One such method is Cardinality Deep, which successfully correlates data and enhances the precision of cost forecasts. The database optimizer can better capture complicated relationships by utilizing deep learning techniques, which will lead to more accurate cost estimates and eventually improved query execution plans.

C. Database Design

Traditional databases are frequently created by knowledgeable database builders, but their capacity to experiment with different design options is constrained. Several learning-based self-design solutions have lately surfaced to get around this constraint. One such method suggests lowering index capacity and increasing indexing effectiveness by using educational indicators [10]. Another strategy emphasizes understanding data structure building [11], considering that various reading systems and devices can need distinct data formats that are appropriate for their environments. The notion of "The Art of Data Structure Alchemy" [12] recommends the development of various data inference engines and the building of storage structures with recommendation functions notwithstanding the difficulty of designing successful applications and optimizing overall data structures. These learning-based techniques can help databases improve their design process, adapt to various needs, and improve data storage architectures.

D. Database Monitoring

Database monitoring is essential for gathering read/write metrics and tracking a database's performance while it is in use. When irregularities like sluggish performance or database assaults are discovered, such as latency, CPU/memory utilization, and abnormalities, administrators can be informed. Traditional monitoring methods, on the other hand, place a great deal of reliance on database administrators to carry out these monitoring chores, which may result in inefficiencies and insufficient reporting of database issues and activities. There have been suggestions made to improve database monitoring by utilizing techniques based on [40], [62], and machine learning to solve these issues. These methods seek to increase the efficiency and thoroughness of the monitoring process by utilizing machine learning algorithms to ascertain the best time and strategy for monitoring the database. Through the use of machine learning techniques, database monitoring may be automated and improved. Administrators can get more precise information about the database's performance and potential problems.

E. Database Security

When it comes to automatically discovering possible security aws, traditional database security techniques like auditing and data masking based on established criteria sometimes fall short. AI-based algorithms are used for a variety of security activities, including the ending of sensitive material,the detection of anomalies, the enforcement of access controls,the defense against SQL injection attacks, and more [13–16].

(I) Sensitive data discovery uses machine learning techniques to automatically identify sensitive data. (II) Finding security risks and possible vulnerabilities in databases is the focus of vulnerability discovery. (III) Mechanisms for access control areused to stop data leakage and unauthorized access. (IV) To identify and counteract SQL injection attacks, deep learning techniques are used in conjunction with user behavior research. These security procedures may be automated and improved with the use of AI algorithms, allowing the system to proactively identify sensitive data, spot abnormalities, enforce access rules, and guard against security breaches like SQL injection attacks. By utilizing AI-based techniques, databases' overall security is enhanced, and sensitive data is protected.

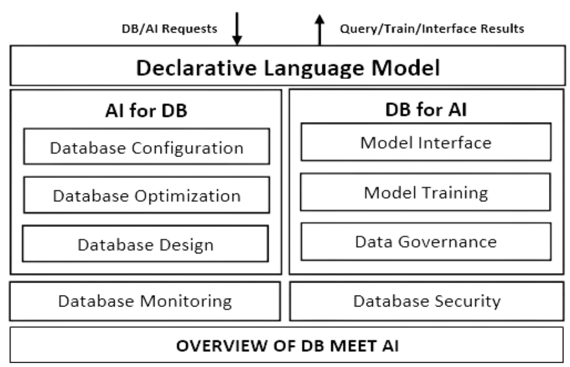

Fig: 2 Overview of Database meet Artificial Intelligence

IV. Database for Artificial Intelligence

Although AI systems are used in many different fields,their adoption is not as ubiquitous as that of DBMS. However, despite difficulties like limited repeatability and complicated

ISSN NUMBER (ONLINE) – 2368-1209 & (PRINT) – 2691-5944

user interfaces, AI has the potential to solve many real-world issues. Database technology can significantly contribute to addressing these challenges since it makes it easier to use AI. The obstacles to implementing AI may be eliminated by incorporating AI capabilities into DBMS. The integration of AI with database technology can improve the usability an accessibility of AI systems, increasing their usefulness and scope of application.

A. Standard Query Framework

Generally regarded as a reasonable language, SQL is widely accepted and used in database systems. In contrast to other cutting-edge machine learning languages, SQL does have limits when it comes to implementing more complex processing methods, such as iterative operations. Fortunately, AI models may be supported by modifying and expanding SQL [17]. It is possible to work on creating frameworks and tools that make it simple to integrate AI models into SQL. The language may become more flexible and potent in handling sophisticated analytics and machine learning workloads by extending SQL statements to accommodate AI models [18]. With the help of these modifications and tools, developers and data professionals will be able to seamlessly and easily make use of the advantages of both AI and SQL.

B. Data Governance

Machine learning systems must guarantee data integrity. Data integration, data labeling, data lineage, data cleaning, and data discovery are just a few of the activities that are covered by data governance, which is important in preserving the quality of data. (I) Data discovery is essential for efficiently locating pertinent data from enormous volumes of data from many sources [19]. (II) To resolve faulty or inconsistent data, which can negatively impact the performance of training models, data cleansing is required. Techniques for data cleaning and integration can be used to x inaccurate data and merge information from many sources [20]. Large amounts of training data for machine learning algorithms may be annotated by using crowdsourcing and subject matter experts. To make the most use of human resources for efficient data labeling, existing expertise, and knowledge bases can be used [21] [22]. (III) Data lineage is crucial for ensuring that machine learning models are operating correctly. The relationship between input and output data may be recorded using database technologies like joins and graph modeling, allowing the verification of data's anterior and posterior linkages [23], [24]. Data governance practices and integrity are essential for machine learning systems. To guarantee accurate and dependable training and performance of machine learning models, it encompasses activities including data discovery, data purification, data labeling, and establishing data lineage.

C. Model Training

Good machine learning models must carefully take several aspects into account when being trained. Model training is time-consuming and difficult a job that requires maintaining models, controlling features, using hardware acceleration, and choosing the best model. (I) Picking the best characteristics from a large pool of available possibilities is a crucial stage that includes evaluation. Although this procedure can be time-consuming, techniques like batching and materialization [25] have been suggested to successfully handle this difficulty. (II) Model selection seeks to choose the best model and attribute values from a wide range of available possibilities. Several parallelprocessing strategies have been proposed to speed up thisprocedure. Task parallelism, bulk synchronous parallelism, parameter web services, and model halt parallelism are a few examples [26, 27, 28, and 29]. Using these strategies can decrease the time and complexity involved in feature selection and model selection This makes it possible to train models more quickly and effectively, which results in the creation of machine learning models of the highest caliber.

D. Model Inference

In-database optimization methods are essential for drawing inferences about the effectiveness of trained models during model inference. These methods involve operator selection, operator support, and execution speedup. (I) Operator Support: Machine learning and AI models frequently include several operators, including scalars and tensors, each of which has a different optimization task. Various in-database strategies have been suggested to facilitate the effective execution of AI operators. To properly manage various types of operators inside the database system, for instance, scalar operations [30], tensor operations [31], and tensor divisions have been created. (II) Operator Selection: Performance might vary greatly depending on which physical model is used to translate the same machine learning model. Estimating the number of resources needed is crucial, as is carefullycontrolling operator use and scheduling. Database operators may be used toestimate resource requirements and choose the best operators [31].Model inference may be improved in terms of performance and resource use byusing in-database optimization approaches, such as supporting several typesof operators and choosing the most effective operators. Improved inferencecapabilities result from these approaches' efficient execution of trainedmodels within the database environment.

V. Conclusion

Modern-day applications Modern applications have been significantly impacted by the combination of database management systems (DBMS) and artificial intelligence (AI). The development of intelligent, effective, and precise systems capable of real-time data analysis, individualized suggestions, and automated decision-making processes is made possible by the integration of AI capabilities into DBMS. A wide number of applications, such as fraud detection, individualized marketing, and predictive maintenance, have been made possible because of this connectivity. AI and DBMS are positioned to play a crucial role in determining the future of contemporary applications as they develop and progressively integrate.

VI. References

[1] W. Wang, M. Zhang, G. Chen, H. V. Jagadish, and B. C. O. et al. The database meets deep learning: Challenges and opportunities. SIGMOD Rec., 2016.

[2] S. Idreos and T. Kraska. From auto-tuning one size is all toself-designed and learned DATA-INTENSIVE SYSTEMS. In SIGMOD, 2019

[3] J. Chen, Y. Chen, and G. L. et al. Data management at Huawei: Recent accomplishments and future challenges. In ICDE, 2019.

[4] G. Li, X. Zhou, and S. Li. Xuanyuan: An ai-native database. IEEE Data Eng. Bull., 42(2):70–81, 2019.

[5] G. Li, X. Zhou, and S. L. et al. Qtune: A query-aware database tuning system with deep reinforcement learning. VLDB, 2019.

[6] D. V. Aken, A. Pavlo, G. J. Gordon, and B. Zhang. Automatic database management system tuning through large-scale machine learning. In SIGMOD 2017, pages 1009–1024, 2017.

[7] J. Zhang, Y. Liu, K. Zhou, G. Li, Z. Xiao, B. Cheng, J. Xing, Y. Wang, T. Cheng, L. Liu, M. Ran, and Z. Li. An end-to-end automatic cloud database tuning system using deep reinforcement learning. In SIGMOD 2019, pages 415–432, 2019.

[8] A. Kipf, T. Kipf, B. Radke, v. Leis, P. A. Boncz, and A. Kemper. Learned cardinalities: estimating correlated joins with deep learning. In CIDR 2019, 2019.

[9] J. Ortiz, M. Balazinska, J. Gehrke, and S. S. Keerthi. Learning state representations for query optimization with deep reinforcement learning. In deem@sigmod, pages 4:1–4:4, 2018.

[10] T. Kraska, A. Beutel, and E. H. C. Et al. The case for learned index structures. In SIGMOD, pages 489–504, 2018.

[11] S. Idreos, N. Dayan, W. Qin, M. Akmanalp, S. Hilgard, A. Ross, J. Lennon, V. Jain, H. Gupta, D. Li, and Z. Zhu. Design continuums and the path toward self-designing key-value stores that know and learn. In CIDR, 2019.

[12] S. Idreos, N. Dayan, W. Qin, M. Akmanalp, S. Hilgard, A. Ross, J. Lennon, V. Jain, H. Gupta, D. Li, and Z. Zhu. Design continuums and the path toward self-designing key-value stores that know and learn. In CIDR, 2019.

[13] Z. Lin, X. Li, and X. Kuang. Machine learning in vulnerability databases. In ISCID 2017, pages 108–113, 2017.

[14] M. L. Goyal and G. V. Singh. Access control in distributed heterogeneous database management systems. Computers & Security, 10(7):661–669, 1991.

[15] P. Tang, W. Qiu, Z. Huang, H. Lian, and G. Liu. SQL injection behavior mining based deep learning. In ADMA 2018, pages 445–454, 2018.

[16] H. Zhang, B. Zhao, H. Yuan, J. Zhao, X. Yan, and F. Li. SQL injection detection based on deep belief network. In CSAE 2019, pages 20:1–20:6, 2019.

[17] T. Schindler and C. Skornia. Secure parallel processing of big data using order-preserving encryption on Google big query. CoRR, abs/1608.07981, 2016.

[18] S. Fernandes and J. Bernardino. What is a big query? In IDEAS, pages 202– 203, 2015.

[19] R. C. Fernandez, Z. Abedjan, F. Koko, G. Yuan, S. Madden, and M.Stonebraker. Aurum: A data discovery system. In ICDE 2018, pages 1001–1012,2018.

[20] G. M. Weiss and H. Hirsh. Learning to predict rare events in eventsequences. In KDD, pages 359–363, 1998

[21] G. Li, J. Wang, Y. Zheng, and M. J. Franklin. Crowdsourced datamanagement: A survey. IEEE Trans. Know l. Data Eng., 28(9):2296– 2319, 2016.

[22] M. Mintz, S. Bills, R. Snow, and D. Jurafsky. Distant supervision for relationextraction without labeled data. In ACL2009, pages 1003–1011, 2009.

[23] J. Cheney, L. Chiticariu, and W. C. Tan. Provenance in databases: Why, how,and where. Foundations and Trends in Databases, 1(4):379–474, 2009.

[24] Y. Cui, J. Widom, and J. L. Wiener. Tracing the lineage of view data in awarehousing environment. ACM Trans. DatabaseSyst., 25(2):179–227, 2000.

[25] C. Zhang, A. Kumar, and C. Re. Materialization optimizations ´ for featureselection workloads. In SIGMOD, 2014.

[26] P. Moritz, R. Nishihara, S. Wang, A. Tumanov, R. Liaw, E. Liang, M. Elibol, Z.Yang, W. Paul, M. I. Jordan, and I. Stoica. Ray: A distributed framework foremerging AI applications. In OSDI 2018, pages 561–577, 2018.

[27] T. Kraska, A. Talwalkar, J. C. Duchi, R. Griffith, M. J. Franklin, and M. I.Jordan. Mlbase: A distributed machine-learning system. In CIDR, 2013.

[28] M. Li, D. G. Andersen, J. W. Park, A. J. Smola, A. Ahmed, V. Josifovski, J.Long, E. J. Shekita, and B. Su. Scaling distributed machine learning with theparameter server. In OSDI2014, pages 583–598, 2014.

[29] S. Nakandala, Y. Zhang, and A. Kumar. Cerebra: Efficient and reproducible model selection on deep learning systems. InDEEM@SIGMOD 2019, pages 6:1–6:4, 2019.

[30] J. M. Hellerstein, C. Re, and F. S. et al. The Madlib analytics library or MADskills, the SQL. PVLDB, 5(12):1700–1711, 2012.

[31] M. Boehm, M. Dusenberry, D. Eriksson, A. V. Evmievski, F. M. Manshadi, N.Pansare, B. Reinwald, F. Reiss, P. Sen, A.

ISSN NUMBER (ONLINE) – 2368-1209 & (PRINT) – 2691-5944

[32] Surve, and S. Tatikonda. Systemml: Declarative machine learning on spark. PVLDB, 9(13):1425–1436, 2016.

VII. BIOGRAPHIES

Barsha Rani Shaw was born on 24 May 2004 in Odisha, India. She completes her Secondary education from Prasad Nagar High School (H.S.) in 2019 and her Higher Secondary education completed from Halisahar Annapurna Balika Vidyalaya in 2021. She is currently pursuing her undergraduate degree in Bachelor of Computer Application at the Global College of Science & Technology.

Biprojit Halder was born on 07 Jul 2003 in Krishnagar, Nadia, India. He completed his Secondary and Higher Secondary from Saktinagar High School (H.S) in 2019 and 2021 respectively. He is currently pursuing his undergraduate degree in Bachelor in Computer Applications at the Global College of Science and Technology

Soham Sen was born on 19 Oct 2002 in Bagula, Nadia, India. He completed his Secondary and Higher Secondary from Bagula High School (H.S.) in 2019 and 2021 respectively. He is currently pursuing his undergraduate degree in Bachelor in Computer Applications at the Global College of Science and Technology.

Sumi Basak was born on 16Jul 2003 in Nabadwip, Nadia, India. She completed his Secondary from Bangavani Institution for Girls in 2019 and Higher Secondary education completed from Nabadwip Balika Vidyalaya in 2021. Now she is pushing for an undergraduate degree in Bachelor's in Computer Applications at the Global College of Science and Technology.

Sumanta Bhattacharya is working as Assistant Professor and Head in Department of Computer Application in Global College of Science & Technology - Krishnagar under the Maulana Abul KalamAzad University of Technology, West Bengal, India. He has more than 17 years of Academic Experience, in including Deputation in Bhutan for 3 Years (under Indian Technical and Economic Cooperation Programme – Government of India) in Gyalpozhing College of Information Technology – Royal University of Bhutan. He has several Research Publications. He reviewed a Book on Software Engineering. He has attended several International Conferences, National Seminars/Workshops, Summer Schools, Refresher / Orientation course in India as well as abroad so far. He is a associated to many Professional Societies in various capacities as well as Lifetime Member, namely Computer Society of India (CSI), Kolkata, India, Forum of Scientists, Engineers and Technologists (FOSET), Kolkata, India International Association of Computer Science and Information Technology (IACSIT), Singapore, Indian Society for Technical Education (ISTE), New Delhi, India, Global Member of the Internet Society (ISoc) – USA;Chapter Member of the Internet Society (ISoc) – Kolkata Chapter, Kolkata, India, Life Member of the Indian Science Congress Association (ISCA), Kolkata, India, The Society of Digital Information and Wireless Communications (SDIWC), Computer Science Teachers Association (CSTA), New York – USA, International Association of Engineers (IAENG) - Hong Kong.

https://doi.org/10.15864/ajac.21012