ITM Web of Conferences **78**, 03025 (2025) https://doi.org/10.1051/itmconf/20257803025
CSEIT 2025

Machine Learning for Optimizing Database Performance

Jiahui Ren

School of Statistics, Capital University of Economics and Business, Beijing, China

Abstract. Due to the developing requirement of data storage, today's database cannot meet the highly developing world. Besides, the wind of Machine Learning (ML) showed a possible solution to the dilemma. Therefore, many researchers have proposed strategies to improve the performance of databases in terms of security and efficiency. This paper discusses the methods of how to use ML to improve the performance of databases. In the methodology part, an introduction is proposed to interpret the workflow of ML first. Then, security problems and relevant solutions are presented in this paper, including improved performance to deploy the database locally to avoid attacks during data transportation, ignore dangerous configurations, and detect traffic data by building a new system. Besides, efficiency issues are also resolved by ML. A strategy is introduced to filter dimensional queries, and sorting functions, to accelerate scheduling time which rapidly constructs the database, a framework and a method to find the relationships under queries also contributed. Lastly, the paper finds the underlying issues that are still under challenges. For instance, reduce manpower on building or managing a database, unified system, and real-world limitations.

1 Introduction

Database's appearance shows the tremendous requirement for data storage because the advancing world may create countless data. Therefore, information could be stored through this kind of elaborate database. The database is an important scheme to store information in huge demand, as for the prosperity of the international market, there will be more demand for data storage. For instance, the database is a critical part of the Internet of Things (IoT) [1, 2]. By using data, people can transmit, connect, and use information [1, 2].

Machine Learning (ML) is the technology that allows computers to learn the way of handling from data and algorithms and improve its ability to handle. ML has been developing for decades. It has a variety of practical applications. In a Coronavirus disease (COVID-19) investigation, machine learning plays a substantial role [3]. Besides, there is a popular topic about price prediction. When predicting Housing prices, Random Forest got the most accurate result, among Random Forest, Extreme Gradient Boosting (XGBoost), and Light Gradient Boosting Machine (LightGBM), which are methods of ML [4]. ML is also a way

outlook_73731DB439C7F84F@outlook.com

for designers to help reduce costs for many aspects, like building construction, engine design, and optimizing the cost of computation in the cloud [5-7].

The fusion of machine learning and database management has already appeared [8]. It was also mentioned that previous database optimizing functions could not deal with the developing requirements, which created a larger-scale database [8]. Using this strategy to build databases will save designers from security issues and low efficiency. Multiple fields have already been using this function to help construct databases. For example, due to the development of New Energy Vehicles (NEVs), the investigation of batteries is increasing [9]. It was needed to find the hidden correlations between data, so using machine learning to accelerate the analysis process is a rational decision [9]. Besides, the solution to the dilemma of effectively storing data in the database is using traditional schemes like Deep Learning (DL), Reinforcement Learning (RL), Natural Language Processing (NLP), etc. [10]. These functions imply a potential for administrating databases, indexing, optimizing queries using, and assuring the quality of the source of data [10]. Another part machine learning can help with the database is improving its accuracy. New methods were put forward. For instance, SageDB can optimize better solutions by analyzing the distribution of data, workload, and hardware [11].

Today's trend shows that cloud database or Artificial Intelligence (AI) automation is the way to enhance the efficiency of Database Management Systems (DBMS) [8]. Original database building costs manpower to analyze the requirement of the client, draw an Entity-Relationship (ER) diagram, build the relational model, and write codes to store all the information. So, there is a greater demand for automatic database building and management in the future.

The organization of the article part is as follows. In the method part, this article focuses on what was used for database optimization in the past by using ML, focusing on finding the results that have been discovered. Next, there are still many difficulties waiting for resolution, so the content will discuss on the future challenges for DBMS. Lastly, the final part of the article summarizes the whole paper.

2. Method

2.1 Preliminaries of machine learning

The machine learning workflow shown in Fig. 1 is composed of multiple stages. The first thing is to find the requirements of the project. Find what the problem is and what predictions the project wants to make. Then analyse the data and verify if it fits the aim of the prediction. The second thing is data preparation. Gather raw data from databases, files, etc. Then remove duplicates and fix the missing values. Besides, label data to supervise learning. Extracting variables from data is the final step for data-oriented work. The third thing is ML modeling. Choose an algorithm and use processed data to train the model. Usually, 60%-80% of data will be chosen as the training set. This is what the model learns from. 10%-20% of the data will be chosen as the validation set. This is used for testing the model during training. Because there is a problem with overfitting data which huge data fluctuation may cause. Overfitting of a model shows perfect accuracy on the training set. However, once the data have changed, it will not fit the new dataset. The remainder will be the test set, just as its name implies, it is an estimation of the model. The last thing is the deployment of the model, keeping monitoring and controlling. This way, the model can run normally now.

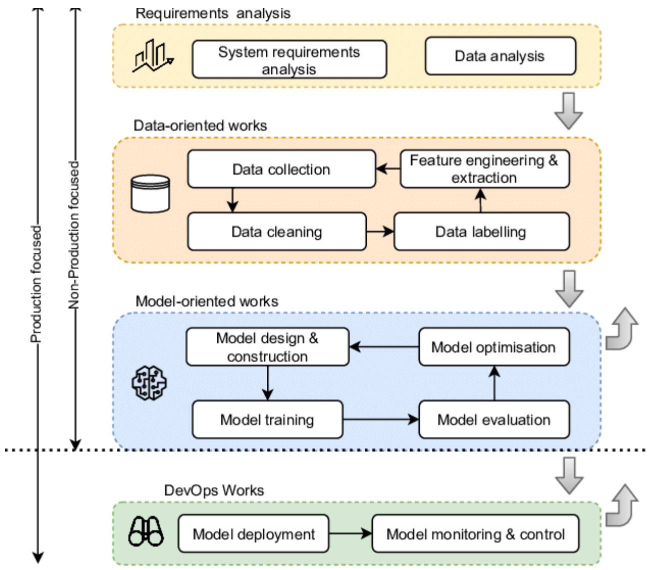

Fig. 1 The workflow of machine learning [12].

2.2 Security of database by using ML

'knobs' are used for controlling storage and Random Access Memory (RAM) will the data costs. Once the knobs has been automatically deployed, there will be no need for researchers to use remote access, which information leakage may happen [13]. Configuration of the knobs used to be designed with database administrators (DBAs), now ML is used for finding a ranked group of the knobs and find which has greater impact on DBMS. The workload characterization will determine the redundant data to make the model simpler. In the knob identification, a method called lasso finds which knob determines the model's performance better, and allocates moderate knobs for DBMS. The automatic tuner will first make a comparison to prior tuning and find the parallel one, then, the algorithm selects test-out knobs to check the efficiency of the configuration of the database. Researchers can use this strategy to identify required DBMS without remote access. This will refuse remote access host DBMS.

An article also focuses on tuning to improve the performance of ML. A strategy that combines Subspace Adaptation and Safety Assessment which includes the black box, and the white box contributed [14]. For the new model, OnlineTune searches for a subspace from the tuning history. The OnlineTune initially sets a base value, if a series of 'failures' or 'successes' happens, it will shrink or extend the region. Based on it, 2 options can be chosen to increase the exploration or align with crucial configuration and improve on exploitation. For the black box, it selects the Gaussian Process (GP) model to access the safety of θ . Then use a previous configuration to restrict the model's performance with a local model. For the white box, it ignores the recommendation as well as the rule. This strategy ignores dangerous configurations to help improve safety.

The Structured Query Language (SQL) injection attack access without authorization [15]. The fusion of models is one of the solutions to detect benign and malicious queries. Others include Naive Bayes, Decision Tree, Random Forest, Support Vector Machine (SVM), and Artificial Neural Network (ANN). The accuracy of all models is higher than 90%, apart from that, the hybrid models' performance is prominent, ANN and SVM's combination get testing accuracy of 98.34% and training accuracy of 98.82%. This function of determination of malicious queries uses the Zero Trust Architectures, which means strictly controlling access based on the principle of ‘never trust, always verify’. This strategy ensures no user is trusted intrinsically. 10-fold cross-validation is also used to guarantee robustness and minimize the risk of overfitting.

Data Tampering means altering databases illegally, which is a challenge for database security [16]. Pathak et al. introduced an Anomaly Detection based on Machine Learning (AD-ML) [17]. This system used unsupervised and supervised ML algorithms to ensure its security by detecting traffic data from the network. They applied 2 algorithms, isolation decision, and decision tree. Isolation decision suits low-dimensional and small-scale data. The isolation decision detected the anomaly. Decision tree Classification and Regression Trees (CART) is used for the classification of the normal and abnormal data which is tempered. The results show that the supervised model did a better job when classifying data, and is easier to deploy.

2.3 Enhance the efficiency of DBMS by using machine learning

In SageDB, a Multi-Dim Indexes and Storage Layout strategy was published [11]. In this strategy, the learned index distinguished itself by filtering dimensional queries, which in most cases, performs better on time use. The research also put effort into sorting functions. Unlike the traditional function, it uses the Cumulative Distribution Function (CDF) puting the records roughly, and then corrects almost perfectly sorted data. Joins can also be accelerated just like sorting, the model ignores the data that will not join. Besides, the scheduling algorithm as a neural network will make scheduling decisions, plus, RL will optimize high-level system objectives. This will speed up scheduling time which helps for constructing a database.

A new method has been proposed to improve query efficiency. First, they put forward the ML-To-SQL framework [18]. Then, among 4 proposals for in-DBMS ML inference, Model Join gets the best score for efficiency improvement. This function is implemented natively as a Central Processing Unit (CPU) and Graphics Processing Unit (GPU) variant and operates independently. While building an operator, they keep data in a columnar layout in CPU caches. Parallelization of linear algebra’s operation additionally benefits from various cores and threads. Besides, considering more parallel units GPU can give than the CPU does, the cu Basic Linear Algebra Subprograms (BLAS) library is used for fitting GPU implementation. To avoid huge memory costs, they have to build a sharing model as a parallel model. As the bias vector was introduced, they reallocated every bias and replicated it in the matrix. That helps enhance efficiency by using Math Kernel Library (MKL)/cuBLAS additionally. From the perspective of optimization. To begin with, a unique node identifier replaced a unique identifying node in the pair to reduce the storage size. Then, they proposed filter predicates on the Layer column of the model table to help replace the filter predicate on the Layer column with a range predicate on the Node column. Besides, a parallel database system that utilizes separating for parallelism——Action Vector has been proposed. At last, vectorized execution and pipelining is also a key which avoid sorting order destroying repartitioning. This eventually builds the ML-To-SQL framework.

There are still various methods that are useful in improving query efficiency in a DBMS [19]. For instance, Deep Learning with Recurrent Neural Networks (RNNs) and Convolutional Neural Networks (CNNs) have the ability to determine queries and relationships within data. Feature Engineering focuses on extracting correlative features from queries, how database schemas influence query performance, and how the fusion of feature selection ML models ensures the efficiency of optimizing decisions. Adaptive query optimization modifies strategy in real-time, which has dynamic and responsive optimization and uses former query execution to modify future strategies. AI can also modify query strategy under a complex condition with the feedback loop. Query Execution Plans (QEPs) estimation on cost and performance is a crucial part of enhancing efficiency. Thanks to ML models and AI, measuring performance for a query or workload is not a big deal.

3 Discussion

Although the application of ML in databases is still developing, some challenges still remain in this field.

First, in most cases, human-involved models (i.e. supervised model) often achieve higher performance than machine-only models (i.e. unsupervised model). There are still improvements to be found in the future to enhance the performance of machine-only models. For instance, in [17], the supervised model achieved higher performance than the unsupervised model. There may be something underlying that machines cannot find but humans can. Even though the unsupervised model achieved a gratifying result, it is a mediocre solution because the more manpower the model requires, the further it strays from automation. What the market needs is something that requires as little manpower as possible. There is still a distance from automatically managing the database. So, future researchers are expected to explore automation in DBMS by enhancing the ability of ML, the better ML improves, the better it can be used to perfect database building.

Second, most of the research is focused on one aspect of using ML to optimize DBMS without integrating them as a unified system, if researchers want to automatically build and maintain an effective database, they need to combine all the researches, but no one has proposed a kind of system to deal with all the prioritizations. Besides, in research [16], the author mentioned that multiple methods were proposed to deal with different aspects of database security, but most of them only got shallow research on that, or they need to be combined. These 2 requirements show the prospect of exploring the practical application of these strategies—fusing models. From the perspective of fusing the model so that everyone could have the ability to build and maintain a database themselves, AI, ML, and the database's organic combination may be the next goal researchers need to explore. From the perspective of model improvement, research guaranteed safety considerations [14], but it took more time to explore. So, whether the combination of the knob and the tuning strategy could work better than using only one model is still under investigation [13, 14]?

Third, some researches not only achieve greater efficiency or better security but also lower the cost of database construction. Although it is great for other researches that only achieves their goal of enhancing the database's performance, which designers do not want to envision, because the ultimate objective they want to achieve is to lower the expenditure and generate more revenue for the project. In research [14], they found that most of the functions were easy to implement and made the database safer. Apart from that, convenience, simplicity, user experience, etc., are under consideration. Therefore, future researchers need to investigate the practical usage of DBMS by using ML. Limit expenditure of computing power and funds on the research. Limitations make people think in multi-dimensional ways. The developers or researchers need to find different solutions to meet the rapidly developing data storage requirements under limitations.

Last, most researchers do not pay much attention to interpretability, in the future, it will be hard to optimize the model easily. Though it is ML's problem, this will finally influence the performance of this research to optimize the database. Some researchers want to get further investigation into it, but they only achieve a few results. For instance, in the passage [20], they only gained interpretability in a particular situation. A study summarized some methods, however, it shows that there is still a lot of room for improvement [21]. Therefore, in the future, designers need to focus on the interpretability of ML.

4 Conclusion

This paper provides a comprehensive review related to database optimization by using ML, due to its strong power. Former methods are proposed to tackle database development problems. ML can be a better choice for designers to ensure the security of the database by automatically analyzing datasets, now users do not need to worry about security problems caused by remote access, because 'knobs' optimization enhances efficiency which makes deploying a local database easier. Besides dangerous configurations problems are solved by OnlineTune, the black box and the white box. ANN and SVM's combination model figures out the SQL injection attack by a strategy that never trust users, improves robustness, and minimizes the risk of overfitting. The high accuracy of this method showes Feasibility. AD-ML is introduced to fix the data tempering problem. From the perspective of efficiency, ML can improve the database's performance by using SageDB to accelerate the construction speed, ML-To-SQL helps improve query efficiency. RNNs and CNN find queries and relationships within data, and use QEPs to estimate the cost and performance of a model. In the future, designers need to focus more on the automation of the database with ML, unified system, real-world limitations, and ML's interpretability.

References

1. Diène, B., Rodrigues, J.J., Diallo, O., Ndoye, E.H.M., Korotaev, V.V.: 'Data management techniques for Internet of Things', Mech. Syst. Signal Process., 2020, 138, 106564
2. Ma, X.: 'The analysis of the internet of things database query and optimization using deep learning network model', PLoS One, 2024, 19 (6), e0306291
3. Kwekha-Rashid, A.S., Abduljabbar, H.N., Alhayani, B.: 'Coronavirus disease (COVID-19) cases analysis using machine-learning applications', Appl. Nanosci., 2023, 13 (3), pp. 2013–2025
4. Truong, Q., Nguyen, M., Dang, H., Mei, B.: 'Housing price prediction via improved machine learning techniques', Procedia Comput. Sci., 2020, 174, pp. 433–442
5. Pham, T.Q.D., Le-Hong, T., Tran, X.V.: 'Efficient estimation and optimization of building costs using machine learning', Int. J. Constr. Manag., 2023, 23 (5), pp. 909–921
6. Kavuri, C., Kokjohn, S.L.: 'Exploring the potential of machine learning in reducing the computational time/expense and improving the reliability of engine optimization studies', Int. J. Engine Res., 2020, 21 (7), pp. 1251–1270
7. Osypanka, P., Nawrocki, P.: 'Resource usage cost optimization in cloud computing using machine learning', IEEE Trans. Cloud Comput., 2020, 10 (3), pp. 2079–2089
8. Li, G., Zhou, X., Cao, L.: 'Machine learning for databases', Proc. 1st Int. Conf. AI-ML Syst., 2021, pp. 1–2
9. Parayil Shaji, S., Tress, W.: 'Data-driven analysis of hysteresis and stability in perovskite solar cells using machine learning', Energy AI, 2025, 20, 100503
10. Fahima, M.M.F., Sreen, A.S., Ruksana, S.F., Weihena, D.T.E., Majid, M.H.M.: 'Machine Learning for Database Management and Query Optimization', Elementaria J. Educ. Res., 2024, 2 (1), pp. 96–108
11. Kraska, T., Alizadeh, M., Beutel, A., Chi, E.H., Ding, J., Kristo, A., et al.: 'SageDB: A learned database system', 2021

12. Idowu, S., Strüber, D., Berger, T.: 'Asset management in machine learning: State-of-research and state-of-practice', ACM Comput. Surv., 2022, 55 (7), pp. 1–35
13. Deepa, M., Dhilipan, J.: 'Security in database management system using machine learning', Int. J. Electron. Secur. Digit. Forensics, 2024, 16 (1), pp. 124–133
14. Zhang, X., Wu, H., Li, Y., Tan, J., Li, F., Cui, B.: 'Towards dynamic and safe configuration tuning for cloud databases', Proc. 2022 Int. Conf. Manage. Data, 2022, pp. 631–645
15. Imtiaz, M.A., Ahmed, M.Z., Khalid, F., Muhammad, A.M., Iqbal, A., Siddique, H.: 'Data Exploration with SQL: A Machine Learning Based End-to-End Prediction and Data Security Framework', Spectrum Eng. Sci., 2025, 3 (2), pp. 272–303
16. Wang, Y., Xi, J., Cheng, T.: 'The overview of database security threats' solutions: traditional and machine learning', J. Inf. Secur., 2021, 12 (1), pp. 34
17. Pathak, A.K., Saguna, S., Mitra, K., Åhlund, C.: 'Anomaly detection using machine learning to discover sensor tampering in IoT systems', Proc. IEEE Int. Conf. Commun. (ICC), 2021, pp. 1–6
18. Kläbe, S., et al.: 'Exploration of Approaches for In-Database ML', Proc. EDBT, 2023, 23, pp. 311–323
19. Panwar, V.: 'AI-Driven Query Optimization: Revolutionizing Database Performance and Efficiency', arXiv preprint, arXiv:2407.13280, 2024
20. Räz, T.: 'ML interpretability: Simple isn't easy', Stud. Hist. Philos. Sci., 2024, 103, pp. 159–167
21. Gao, L., Guan, L.: 'Interpretability of machine learning: Recent advances and future prospects', IEEE MultiMedia, 2023, 30 (4), pp. 105–118