Jurnal Ilmiah
Teknologi Sistem
Informasi

Jitsi

Vol. 7 (2026) No. 2

https://jurnal-itsi.org/index.php/jitsi; E-ISSN: 2722-4600; ISSN: 2722-4619
DOI: 10.62527/jitsi.7.2.601

Machine Learning and Decision Support in The Exploitation of MDX Queries: Case of Banking Products and Services

Patrick Tshimanga Kapuba[#], Donking Kialanda Nsidiovova[#], Jean Teddy Tene Nzinga[#] , Simon Ntumba Badibanga[#], Eugene Mbuyi Mukendi[#]

[#] *Department of Mathematics, Statistics and Computer Science, University of Kinshasa, Democratic Republic of the Congo*
E-mail: pattshimanga2000[at]gmail.com

A B S T R A C T S

The increasing complexity of analytical workloads in banking systems challenges traditional query scheduling mechanisms, particularly in OLAP environments where MDX queries exhibit heterogeneous computational costs and business criticality. This study proposes a semantic-aware scheduling framework based on incremental machine learning to dynamically prioritize MDX queries in real time. The approach models query prioritization as a classification problem, integrating both technical features and business-driven criticality. A Hoeffding Tree algorithm is employed to enable continuous learning from streaming query data without requiring retraining. The model is evaluated using a simulated dataset of 10,000 MDX queries reflecting realistic banking scenarios, including risk monitoring and regulatory reporting. Experimental results show that the proposed approach achieves a classification accuracy of 94.1% and significantly reduces processing latency for high-priority queries, with improvements reaching 42.4% compared to FIFO scheduling. The inference overhead remains negligible, ensuring compatibility with real-time system constraints. These findings demonstrate the effectiveness of integrating incremental learning into query scheduling and highlight the potential of semantic-driven optimization in decision support systems. The study contributes to bridging the gap between learned database systems and business-aware query management.

Manuscript received Jun 22, 2026; revised Jun 25, 2025. accepted Jun 29, 2025 Date of publication Jun 30, 2026. International Journal, JITSI : Jurnal Ilmiah Teknologi Sistem Informasi licensed under a Creative Commons Attribution-Share Alike 4.0 International License



Keywords / Kata Kunci — ***Machine Learning, MDX Queries, Query Scheduling, OLAP Systems, Data Stream Mining***

CORRESPONDING AUTHOR

Patrick Tshimanga Kapuba
Department of Mathematics, Statistics & Computer Science, University of Kinshasa, Democratic Republic of the Congo
Email: pattshimanga2000[at]gmail.com

1. INTRODUCTION

Over the past decades, the digital transformation of financial systems has profoundly altered the mechanisms of data production, processing, and exploitation. In banking institutions, this evolution has resulted in an exponential increase in the volume, velocity, and complexity of data, making decision-making processes increasingly reliant on high-performing analytical systems. In emerging economies, such as the Democratic Republic of Congo (DRC), these challenges are even more critical as infrastructural and regulatory constraints impose heightened demands for reactivity, reliability, and traceability in financial analyses. In this context, decision support systems based on OLAP (Online Analytical Processing) architectures occupy a central place in the multidimensional analysis of banking data, particularly for liquidity monitoring, risk management and regulatory reporting (Abelló, et al., 2021). These systems utilize specialized query languages, such as MDX (Multidimensional Expressions), allowing complex data cubes to be queried along multiple analytical axes (Vaisman, et al., 2022). However, the increasing number and complexity of MDX queries lead to high

computational costs, which can significantly degrade the overall system performance, especially in situations of high query contention. Traditional query scheduling approaches, based on static logics such as First-In-First-Out (FIFO) or heuristics based on estimated cost, present significant limitations in this context. They do not allow for the integration of the semantic dimension related to the business criticality of queries, which can lead to delayed processing of critical queries, particularly those related to regulatory compliance or financial risk detection. This discrepancy between technical optimization and business priority is now a major challenge for banking decision-making systems. Concurrently, recent advancements in machine learning, and more specifically in data stream mining, offer promising perspectives for the dynamic management of analytical systems. Unlike batch learning approaches, incremental algorithms can process continuous data streams in real-time, while adapting to changes in underlying distributions, a phenomenon known as concept drift (Gomes, et al., 2023). Among these approaches, incremental decision trees, such as the Hoeffding Tree algorithm, stand out for their low computational complexity and their ability to provide fast and interpretable decisions, characteristics particularly suited to critical environments (Bifet, et al., 2023) In the field of databases, the integration of machine learning into query optimization processes has given rise to the paradigm of "learned database systems". These works primarily aim to improve the estimation of execution costs or to optimize query plans (Marcus, et al., 2021) (Zhou, et al., 2022). However, they remain largely focused on technical criteria and neglect the explicit consideration of business value associated with analytical queries. In particular, the literature remains limited regarding the use of machine learning models for dynamic query prioritization based on their operational criticality in an OLAP context.

Thus, this research lies at the intersection of two scientific fields: data stream learning and analytical query optimization. It aims to fill a double scientific gap: (i) the lack of integration of the business semantic dimension into OLAP query scheduling mechanisms, and (ii) the lack of empirical applications of incremental learning algorithms in the processing of MDX queries in a banking environment. More precisely, the central problem of this study can be formulated as follows: To what extent can an incremental Machine Learning model improve the real-time prioritization of MDX queries by simultaneously integrating their technical complexity and business criticality into banking decision-making systems? To answer this question, we propose a semantic scheduling framework for MDX queries based on the use of the Hoeffding Tree algorithm. This framework allows dynamic classification of streaming queries according to syntactic and business characteristics, in order to optimize their real-time processing. An experiment is conducted on a simulated stream of 10,000 queries, structured according to scenarios inspired by banking practices in the DRC, allowing evaluation of the model's performance in terms of precision, latency, and operational impact. The main contributions of this research are as follows:

- Conceptual contribution: proposal of a scheduling model semantic, explicitly integrating business criticality into query management analytical;
- Methodological contribution: adaptation of learning techniques incremental to the dynamic classification of MDX queries in an OLAP environment;
- Operational contribution: design of a middleware-type architecture allowing non-intrusive integration of machine learning into existing decision-making systems;
- Empirical contribution: experimental demonstration of the model's effectiveness proposed in terms of latency improvement for critical queries.

The rest of the article is structured as follows. Section 2 presents a critical review of the literature related to data stream learning and analytical query optimization. Section 3 details the methodology and the proposed model. Section 4 describes the experimental protocol. Sections 5 and 6 present the results and their discussion. Finally, Section 7 concludes the study and proposes research perspectives.

2. LITERATURE REVIEW

2.1. Data Stream Learning and Incremental Classification

Data stream learning (data stream mining) has emerged as a central paradigm for real-time data processing in dynamic environments. Unlike traditional batch learning approaches, stream models must be capable of incremental learning, while adapting to evolving data distributions, a phenomenon known as concept drift (Gama, et al., 2014). In this context, incremental decision trees, and particularly the Very Fast Decision Tree (VFDT), also known as Hoeffding Tree, represent a major advance. Introduced by Domingos and Hulten this algorithm relies on Hoeffding's inequality to statistically reliably select partitioning attributes from a limited number of observations (Domingos, et al., 2000). This property ensures low time complexity and an online learning capability suitable for massive data streams.

Subsequent work has consolidated the relevance of these models in non-stationary environments, highlighting their ability to maintain high performance while adapting to distribution changes (Bifet, et al., 2010). In particular, approaches based on adaptive forests and drift detection mechanisms have improved the robustness of these models in evolving contexts. However, despite their effectiveness, these methods have primarily been applied to generic classification tasks (fraud detection, recommendation systems, sensor analysis), and their use in the specific context of analytical query management systems remains limited.

2.2. Query Optimization and Learned Database Systems

Query optimization is a fundamental pillar of database management systems. Historically, optimizers rely on analytical cost models, using data statistics to estimate optimal execution plans (Selinger, et al., 1979).

However, these approaches have significant limitations when faced with the increasing complexity of modern analytical queries. In response to these limitations, a new research trend, known as "learned database systems", proposes integrating Machine Learning techniques into optimization processes (Marcus, et al., 2021) (Zhou, et al., 2022).

From this perspective, several works have explored the use of predictive models to estimate query cost or latency. For example, Krishnan et al. propose a concurrency control mechanism based on performance prediction, while Marcus et al. introduce reinforcement learning-guided optimization approaches (Krishnan, et al., 2019) (Marcus, et al., 2019). Furthermore, supervised learning models have been used to predict the latency of complex queries, thus allowing system performance to be anticipated. These approaches contribute to improving system efficiency, but they remain focused on purely technical criteria, such as execution time or resource consumption. Thus, current literature highlights a growing integration of artificial intelligence into database systems, but without truly considering the business dimension of queries.

2.3. Query Scheduling and Business Value Integration

Query scheduling aims to organize the execution of concurrent queries to optimize overall system performance. Classical approaches rely on simple strategies such as FIFO or Shortest Job First, as well as heuristics based on cost estimations. More recent work has introduced the concept of value-based scheduling, in which queries are prioritized according to their expected contribution to a global objective. This approach represents a significant advance by introducing a decision-making dimension into query management.

However, several limitations remain. Firstly, the notion of value is generally defined abstractly, without explicit consideration of the business context. Secondly, these approaches remain mostly static and do not adapt dynamically to the evolution of query flows. Finally, very few works explicitly combthine e techniques learning machine incremental with analytical query scheduling, particularly in OLAP environments using MDX queries. This absence is all the more problematic as the multidimensional complexity of queries requires adaptive and intelligent mechanisms.

TABEL 1. Critical Literature Review (Source: Author)

Authors	Contribution	Method	Main Limitation
Domingos & Hulten (2000)	VFDT / Hoeffding Tree	Learning incremental	No application to databases
Gama et al. (2014)	Concept drift	Theoretical Survey	Non-operational
Bifet et al. (2010)	Adaptive learning	Stream mining	Generic cases
Krishnan et al. (2019)	Query prediction	Supervised ML	Performance focus
Marcus & Papaemmanouil (2021)	Learned optimizer	RL	High complexity

2.4. Research Workflow

A critical analysis of the literature highlights a structural gap at the intersection of three research fields:

- Firstly, data stream learning works offer models that are powerful for dynamic classification, but remain largely disconnected from issues related to analytical database systems.
- Secondly, query optimization approaches integrating machine learning primarily focus on technical criteria, without integrating the business dimension of queries.
- Thirdly, existing scheduling methods, although sometimes introducing the notion of value, do not offer adaptive mechanisms based on real-time learning.

Thus, there is a clear scientific gap: the absence of an integrated framework combining incremental learning, dynamic query scheduling, and explicit consideration of business criticality in OLAP systems. This research aims to fill this gap by proposing a semantic scheduling model based on incremental machine learning, applied to MDX queries in a banking context.

3. RESEARCH METHODOLOGY

3.1. General Research Framework

This research is part of an applied quantitative approach, aiming to design and evaluate an intelligent MDX query-scheduling model in an OLAP environment. The adopted approach relies on integrating an incremental machine-learning algorithm into a dynamic query classification process, to optimize their real-time prioritization.

The methodological framework is based on three main components:

- the modeling of the query prioritization problem,
- the design of a classification model based on stream learning,
- Experimental evaluation using simulated data representative of the banking context.

3.2. Formalization of the Query Scheduling Problem

Let a stream continuous of queries MDX denoted :

$$\{Q\} = \{q_1, q_2, \dots, q_t\} \quad (1)$$

Each query q_i is characterized by a vector of explanatory variables:

$$X_i = (C_i, S_i, T_i) \quad (2)$$

where:
C_i : syntactic complexity of the query (number of dimensions, MDX operators, cube depth),
S_i: business criticality score (related to the banking product or the nature of the analysis),
T_i: estimated historical execution time.

The objective is to define a function of prioritization :

$$\{P\}(q_i) = f(C_i, S_i, T_i) \quad (3)$$

such that:
- high business criticality queries are processed with priority,
- the overall system latency is minimized,
- resources are allocated optimally.

This problem can be reformulated as a supervised classification problem, where each query is assigned to a priority class:

$$Y_i \in \{P_0, P_1, P_2, P_3\} \quad (4)$$

corresponding respectively to the levels: urgent, critical, high, and standard.

3.3. Incremental Classification Model based on Hoeffding Tree

To process the real-time query stream, we use the Hoeffding Tree algorithm, an incremental decision tree designed for learning on data streams (Domingos & Hulten, 2000). The fundamental principle is based on Hoeffding's inequality, which guarantees that an estimate based on a sample is close to the true value with high probality. The Hoeffding is defined as follows :

$$\epsilon = \sqrt{\left\{\left\{R^2 \backslash ln\left(\frac{1}{\delta}\right)\right\} / \{2n\}\right\}} \quad (5)$$

where:
R : range of values for the splitting criterion
δ : error probability
n : number of observations

At each node of the tree, the algorithm selects the best splitting variable by comparing information gains. If the difference between the two best attributes exceeds ϵ, the split is performed. This approach offers several advantages in the studied context:
- Low complexity: learning in O(F), inference in O(log L),
- Online processing: no need to store data
- Drift adaptation: continuous model update

These properties make the Hoeffding Tree particularly suitable for dynamic classification of MDX queries in an OLAP environment (Bifet, et al., 2010).

3.4. Construction of the business criticality score

Business criticality is a central dimension of the proposed model. It is defined as a weighted function of the business characteristics associated with each query:

$$S_i = \sum_{\{k=1\}}^{\{K\}}{}_{\{K\}} w_k \; f_{k(q_i)} \quad (6)$$

where:
$f_{k(q_i)}$: indicator function related to a business attribute (product type, risk, compliance, etc.)
w_k : eight associated with this attribute

The weights are defined on a linear scale from 1 to 10 (or alternatively as percentages summing to 100%). In the banking context, the weights are defined according to operational priorities, as summarized in Table 1 below. To derive these values, we followed a formalized and iterative process, beginning with Strategic and Regulatory Alignment, followed by Analytical Hierarchy Process (AHP) and Business Impact Analysis (BIA), and concluding with Dynamic Calibration and a feedback loop (iterative refinement).

TABEL 2. Weight Distribution in the Banking Sector (Source: Author)

Business Attribute (fk)	Weight wk (out of 10)	Normalized Weight (%)	Justification
Regulatory compliance (AML/CFT, BCBS 239, Basel III)	10	40 %	Non-negotiable; Non-compliance entails severe financial penalties and immediate reputational damage.
Risk management (credit, market, operational)	8	30 %	Essential for the bank's solvency and long-term survival.
Liquidity monitoring (LCR, NSFR, cash flow analysis)	6	20 %	Critical during periods of market stress; delays or errors distort regulatory reporting.
Performance analysis (P&L, ROI, product-line profitability)	3	10 %	Important for strategic steering and profitability, yet a processing delay does not immediately disrupt daily banking operations.

Applying the above yields the following score formulation:

$$S_i = 10.f_{conform}(q_i) + 8.f_{risque}(q_i) + 6.f_{liquid}(q_i) + 3.f_{perf}(q_i) \quad (7)$$

Where each $f_k(q_i)$ takes the value 0 or 1, depending on whether the query pertains to the corresponding domain. Thus, a purely regulatory query would yield Si=10, whereas a cross-sectional query (e.g., combining liquidity and risk) would yield 6+8=14, thereby justifying a higher priority within the classification system. This modeling approach explicitly integrates the decision-making value of queries into the classification process.

3.5. Proposed system architecture

The proposed system is based on a middleware-type architecture, inserted between the OLAP tool and the query execution engine. The process unfolds as follows:

1. interception of incoming MDX queries
2. extraction of characteristics X_i
3. prediction of priority class via the HT model
4. insertion into a prioritized queue
5. execution of queries according to their priority level

This architecture allows non-intrusive integration into existing systems, while significantly improving query flow management.

3.6. Experimental Protocol

The model evaluation is performed using a simulated stream of 10,000 MDX queries, generated according to distributions representative of banking uses. Each query is characterized by simulated syntactic complexity, analytical depth, estimated execution time, and a business criticality level. The model is evaluated in test-then-train mode, typical of stream environments (Gama, et al., 2014), allowing its performance to be measured under real conditions. The metrics used are:

- Accuracy : overall classification precision
- Cohen's Kappa : robustness against class imbalance
- Inference time : latency introduced by the model
- Operational gain : improvement in critical query processing

4. EXPERIMENTAL DESIGN

4.1. Experiment Objective

The objective of the experiment is to evaluate the performance of the proposed semantic scheduling model based on incremental learning, in comparison with classical query scheduling strategies. More precisely, the experiment aims to measure to what extent the integration of business criticality and technical complexity into a dynamic classification model improves the prioritization of critical queries, reduces the average latency of high-decision-value queries, and maintains an acceptable overall system performance.

4.2. Experimental Environment

The experiment is conducted in a simulated environment reproducing the conditions of a banking OLAP system. The model is implemented in Python, relying on a library dedicated to data stream learning, allowing incremental processing of observations. The experimental protocol is based on a stream processing logic, where queries are analyzed sequentially, without global data storage, in accordance with the principles of data stream mining (Gama, et al., 2014).

4.3. Data Generation and Structuring

The dataset consists of a stream of 10,000 simulated MDX queries, constructed to reflect the characteristics of banking systems. Each query is described by a vector of explanatory variables:

$$X_i = (C_i, D_i, H_i, S_i) \quad (8)$$

where:

C_i: syntactic complexity (number of MDX operators, filters, functions)

D_i: analytical depth (number of dimensions used)

H_i :simulated historical execution time

S_i : business criticality score

Queries are divided into four priority classes:

TABEL 3. MDX queries class distribution based on priority

Class	Level	Description	Proportion
P0	Urgent	Compliance, AML-CFT	10 %
P1	Critical	Risk, liquidity	15 %
P2	High	Financial analysis	35 %
P3	Standard	Secondary reporting	40 %

This distribution reflects a realistic context where critical queries are minority but strategic.

4.4. Evaluation Method (Test-then-Train)

Model evaluation is performed according to the test-then-train protocol, widely used in data stream environments (Gama et al., 2014). At each iteration:

1. The model predicts the priority class of query q i ,
2. The prediction is compared to the actual class,
3. The model is updated with the new observation.

This protocol allows simulating a real context in which data arrives continuously and where the model must adapt in real time.

4.5. Comparison Methods (baseline models)

To evaluate the added value of the proposed model, it is compared to three baseline strategies:

- FIFO (First-In-First-Out) : chronological scheduling without prioritization
- Shortest Job First (SJF) : prioritization based on estimated execution time
- Cost-based scheduling : scheduling based on an analytical cost estimation

These methods represent the classical approaches used in query management systems.

4.6. Evaluation Metrics

Model performance is evaluated using several complementary indicators:

4.6.1. Predictive Performance

Accuracy :

Accuracy = {Numbero fpredictions correct}}/{Total number of observations}}

Cohen's Kappa : measures classification quality taking into account chance.

4.6.2. Operational Performance

Average Response Time :

$$T_{\{avg\}} = \{1\}/\{N\} \sum_{\{i=1\}}^{\{N\}T} T_{\{i\}} \quad (9)$$

- Processing time for critical queries (P0,P1)
- Latency Reduction (%) :

$$Gain = \{T_{\{baseline\}} - T_{\{model\}}\}/\{T_{\{baseline\}}\} \times 100 \quad (10)$$

4.6.3. System Performance

- Model inference time (overhead)
- Processing throughput (queries/second)

4.7. Research Hypotheses

The experimentation aims to test the following hypotheses:

- H1 : The model based on Hoeffding Tree significantly improves the accuracy of MDX query classification.
- H2 : The integration of business criticality allows a significant reduction in the processing time of critical queries compared to classical approaches.
- H3 : The proposed model introduces negligible latency compatible with the constraints of real-time OLAP systems.

4.8. Validity and Limitations of the Experimentation

Although the adopted approach allows for reproducing a realistic environment, certain limitations must be highlighted. Firstly, the data used is simulated, which may introduce a bias compared to real data from banking

systems. Secondly, business criticality parameters are based on sectoral assumptions that could vary across institutions. Finally, the absence of a standard benchmark (such as TPC-DS) limits direct comparability with other works. These limitations open up avenues for improvement in future research, particularly through the integration of real data and the evaluation of the model in production environments.

5. RESULTS

5.1. Performance of the Incremental Classification Model

The evaluation of the model based on the Hoeffding Tree algorithm highlights a rapid convergence of predictive performance in a stream environment. After processing 10,000 simulated queries, the model achieves an overall accuracy of 94.1%, accompanied by a Cohen's Kappa coefficient of 0.85, indicating strong classification robustness beyond chance. These results confirm the ability of incremental models to maintain a high level of performance in non-stationary contexts, in line with the conclusions of Gama et al., who emphasize the effectiveness of adaptive approaches in the face of concept drift. Furthermore, the observed rapid convergence suggests that the model is capable of learning effectively from a limited volume of observations, which is a decisive advantage in real-time systems (Gama, et al., 2014). A finer analysis by class reveals that performance is particularly high for critical queries. The recall rate for class P0 (urgent) reaches 96%, indicating that almost all business-critical queries are correctly identified and prioritized. This capability is essential in the banking context, where processing times for compliance or risk management related queries must be minimized.

5.2. Impact on System Operational Performance

The integration of the semantic scheduling model has a significant impact on the system's operational performance, particularly on the processing times of critical queries.

TABEL 4. Comparison of Average Processing Times

Class	FIFO (s)	Proposed Model (s)	Improvement
P0 (Urgent)	12,5	7,2	-42,4 %
P1 (Critical)	15,8	9,8	-38,0 %
P2 (High)	22,1	18,5	-16,3 %
P3 (Standard)	35,4	41,2	+16,4 %

Source: Author

The results show a substantial reduction in processing time for critical requests (P0 and P1), with a gain of up to 42.4%. This improvement is achieved at the cost of a slight degradation in performance for standard requests (P3), which reflects a deliberate trade-off in favor of high-decision-value requests.

This result is consistent with the work of Zhou et al., who demonstrate that value-based scheduling optimizes overall performance by prioritizing the most important requests. However, unlike these approaches, our model explicitly integrates a contextualized business dimension, which enhances the relevance of scheduling decisions (Zhou, et al., 2022).

5.3. Analysis of Latency and Algorithmic Overhead

A central element in the model's evaluation concerns the latency introduced by the classification process. The results show that the average inference time of the model is less than 1.5 milliseconds per request, which is negligible compared to the overall execution times of MDX queries. This low algorithmic overhead confirms the suitability of the Hoeffding Tree for real-time environments, due to its logarithmic complexity during the inference phase. This result is consistent with the theoretical properties of incremental decision trees, which guarantee high scalability even in the presence of massive data streams (Domingos & Hulten, 2000). Thus, the integration of the model into a scheduling middleware does not alter the overall system performance, while providing significant gains in terms of prioritization.

5.4. Comparison with Classical Approaches

Comparison with reference methods (FIFO, Shortest Job First, and cost-based scheduling) highlights the limitations of traditional approaches. Strategies based solely on technical criteria, such as SJF or cost models, optimize average processing time but fail to guarantee the prioritization of critical requests. In particular, these methods can lead to delaying the processing of high business value requests if they exhibit high complexity. Conversely, the proposed model allows for the simultaneous integration of technical and business criteria, thus offering an optimal compromise between overall performance and decision-making relevance. This capability is a major advantage in banking environments, where the criticality of analyses cannot be reduced to a simple cost estimate.

6. DISCUSION

The obtained results highlight the interest of an approach based on incremental machine learning for the dynamic management of analytical queries in an OLAP environment. The model's high predictive performance, combined with low inference latency, confirms the relevance of using the Hoeffding Tree algorithm in a stream

processing context. This observation is consistent with work on data stream learning, which emphasizes the ability of incremental models to maintain stable performance in non-stationary environments characterized by continuous data streams (Gama, et al., 2014). Beyond technical performance, the results highlight a transformation in query scheduling logic. The significant improvement in processing times for critical queries reflects the effectiveness of a prioritization mechanism explicitly integrating the business dimension. Unlike classical approaches, which primarily rely on technical criteria such as execution time or estimated complexity, the proposed model introduces a hierarchy based on the decision-making value of queries. This orientation partially aligns with work on value-based scheduling, which aims to optimize performance based on task importance (Zhou, et al., 2022), while extending it by integrating a contextual dimension specific to the banking sector. The observed degradation in performance for low-priority queries is an expected outcome within a differentiated optimization mechanism. This phenomenon reflects a trade-off between overall efficiency and targeted prioritization, where system resources are reallocated to benefit highly critical requests. In a banking context, where certain analyses are directly linked to regulatory requirements or risk management, this trade-off appears consistent with the operational objectives of decision-making systems.

Furthermore, the stability of the model's inference time confirms the compatibility of the proposed approach with real-time system constraints. This property is particularly important in OLAP environments, where the introduction of additional mechanisms must not degrade overall performance. The obtained results thus corroborate the theoretical properties of incremental decision trees, which guarantee controlled complexity and online processing capability (Domingos, et al., 2000). The integration of machine learning into the scheduling process also falls within a broader dynamic of database system transformation, marked by the emergence of so-called "intelligent" or "learning" systems. Recent work in this field highlights the potential of artificial intelligence to improve query optimization and resource allocation mechanisms (Marcus, et al., 2021) (Zhou, et al., 2022). However, these approaches remain predominantly focused on technical criteria. The main contribution of this research lies in the explicit introduction of the business dimension into this process, allowing system performance to be aligned with decision-making priorities.

Finally, the results obtained open up interesting perspectives for the evolution of decision-making systems towards autonomous architectures. The use of an incremental learning model for query management constitutes a first step towards the development of systems capable of dynamically adapting their behavior according to data flows and business needs. This perspective is in line with research on self-adaptive databases, which aim to design systems capable of optimizing their operation autonomously. Despite these contributions, certain limitations must be highlighted. The use of simulated data constitutes a first approximation that could be enriched by the exploitation of real data from banking systems. Furthermore, the definition of criticality scores relies on sectoral assumptions that could vary according to institutional contexts. Finally, the study focuses on a specific algorithm, and a comparison with other incremental learning models or hybrid approaches could allow for refinement of the results.

7. CONCLUSIONS

This research aimed to explore the contribution of incremental machine learning in optimizing MDX query scheduling in an OLAP environment, with particular attention paid to integrating the business dimension into the decision-making process. The results obtained confirm that an approach based on dynamic query classification significantly improves the management of analytical flows, especially for high-criticality queries. The introduction of a model based on the Hoeffding Tree algorithm demonstrated that it is possible to reconcile high predictive performance with real-time constraints. The model's ability to process queries incrementally, without requiring global data storage or a retraining phase, is a major asset in environments characterized by continuous and evolving flows. Furthermore, the explicit integration of business criticality into the prioritization process allows for overcoming the limitations of traditional approaches, by aligning system performance with the operational needs of banking institutions.

Scientifically, this research helps fill a gap identified at the intersection of data stream learning, query optimization, and business value consideration. It proposes a conceptual and operational framework for integrating incremental learning models into decision-making systems, while highlighting the importance of a semantic approach to query scheduling. This contribution is in line with recent work on intelligent database systems, which aim to integrate artificial intelligence at the heart of data management mechanisms (Marcus, et al., 2021) (Zhou, et al., 2022). Operationally, the results suggest that integrating intelligent scheduling middleware can be an effective solution for improving OLAP system performance without requiring a complete overhaul of existing infrastructures. This approach is particularly relevant for banking contexts in emerging economies, where technical and organizational constraints often limit investment capacity in complex architectures.

However, certain limitations must be considered. The use of simulated data is a first approximation that would benefit from being supplemented by real data from banking systems. Furthermore, the definition of criticality scores relies on sectoral assumptions that could vary depending on institutional contexts. Finally, the study focused on a specific algorithm, and exploring other incremental or hybrid learning models could enrich the

results. From this perspective, several research avenues can be considered. Firstly, integrating deep reinforcement learning techniques could dynamically optimize resource allocation based on system state and query flows, moving beyond purely classificatory approaches. Secondly, extending the model to multi-user and distributed environments would be an important step for its application to large-scale systems. Thirdly, utilizing real data from banking systems would allow for empirical validation of the model's robustness under operational conditions.

Finally, this research paves the way for the development of autonomous database systems, capable of adapting their behaviour in real-time according to business needs and technical constraints. Integrating continuous learning mechanisms into decision-making architectures could thus constitute a step towards "self-driving" systems, where query optimization and resource allocation are managed intelligently and adaptively.

REFERENSI

[1] Abelló A., Romero O. et Vassiliadis P. Evolution of OLAP: From data cubes to data lakes. [Revue] // Foundations and Trends in Databases. - 2021. - pp. 11(1–2), 1–189..

[2] Bifet A. [et al.] Machine learning for data streams: With practical examples in MOA [Revue] // MIT Press. - [s.l.] : MIT Press, 2023.

[3] Bifet A. [et al.] MOA: Massive Online Analysis [Revue] // Journal of Machine Learning Research. - [s.l.] : Journal of Machine Learning Reseach (JMLR), 2010. - Vol. 11. - pp. 1601-1604.

[4] Domingos P. et Hulten G. Mining high-speed data streams. In Proceedings of the 6th ACM SIGKDD [Conférence] // ACM SIKDD. - [s.l.] : International Conference on Knowledge Discovery and Data Mining, 2000. - pp. 71-80.

[5] Gama J. [et al.] A survey on concept drift adaptation. [Revue] // ACM Computing Surveys. - [s.l.] : ACM Computing Surveys, 2014. - 4 : Vol. 46. - p. 44.

[6] Gomes H. M. [et al.] Machine learning for streaming data: State of the art, challenges, and opportunities [Revue] // ACM Computing Surveys. - [s.l.] : AMC Computing Survey, 2023. - 13 : Vol. 55. - pp. 1–36.

[7] Krishnan K. [et al.] Prediction-based concurrency control [Conférence] // ACM SIGMOD. - [s.l.] : In Proceedings of the ACM SIGMOD Conference, 2019.

[8] Marcus R. [et al.] Neo: A Learned Query Optimizer [Revue] // Proceedings of the VLDB Endowment (PVLDB). - 2019. - 11 : Vol. 12. - pp. 1705-1718.

[9] Marcus R. et Papaemmanouil O. Bao: Making learned query optimization practical. [Conférence] // In Proceedings of the 2021 International Conference on Management of Data. - [s.l.] : ACM SIGMOD, 2021. - pp. 1275–1288.

[10] Selinger P. Griffiths [et al.] Access path section in a Relational Database Management System [Conférence]. - New York : ACM SIGMOD International Conférence on Manaement of Data, 1979.

[11] Vaisman A. et Zimányi E. Data warehouse systems: Design and implementation (2nd ed.). [Ouvrage]. - [s.l.] : Springer, 2022. - p. 2022.

[12] Vaisman A., & Zimányi, E. Data warehouse systems: Design and implementation (2nd ed.) [Revue]. - [s.l.] : Springer, 2022.

[13] Vaisman A., Zimányi, E. Data warehouse systems: Design and implementation [Revue]. - [s.l.] : Springer, 2022. - (2nd ed.).

[14] Zhou X [et al.] Database meets AI: A survey [Revue] // IEEE Transactions on Knowledge and Data Engineering. - [s.l.] : IEEE Transactions on Knowledge and Data Engineering (TKDE), 2022. - 3 : Vol. 34. - pp. 1096–1116