Journal of Neonatal Surgery
ISSN(Online): 2226-0439
Vol. 14, Issue 6s (2025)
https://www.jneonatalsurg.com

OPEN ACCESS

Design of an Integrated Model Using R-GCN, TPOT, and Transformers for Efficient NoSQL Data Processing and Analysis

Pragya Lekheshwar Balley[1], Dr. Shrikant V. Sonekar[2]

[1]PGTD of Computers, RTMNU Nagpur.

Email ID: pragyaballey26@gmail.com

[2]Professor, Department of CSE, J D College of Engineering and Management Nagpur.

Cite this paper as: Pragya Lekheshwar Balley, Dr. Shrikant V. Sonekar, (2025) Design of an Integrated Model Using R-GCN, TPOT, and Transformers for Efficient NoSQL Data Processing and Analysis. *Journal of Neonatal Surgery*, 14 (6s), 298-314.

ABSTRACT

The rapid deployment of NoSQL databases to manage large complex data, unstructured, and sample dataset has posed huge challenges for the efficient processing and analysis of such data samples. Current approaches, particularly those using rule-based schemes for schema detection and query optimization, fail to address the dynamic and heterogeneous nature of NoSQL data samples. To address these shortcomings, this work proposes an overall framework of integrating several advanced methods based on machine learning into NoSQL data processing and analysis to improve efficiency. This work begins with Relational Graph Convolutional Network, a method that dynamically infers schema from the NoSQL database. This helps in automatically detecting intricate relationships within data, reducing schema processing timestamp by 30%. We extend TPOT with transformers-BERT and convolutional neural networks-ResNet for feature selection of multimodal data text, image, and tabular, improving accuracy by 15-20% against the model. MMT allows us to fuse disparate types of data into a shared latent space, lifting multimodal classification accuracy from 78% to 90%. We use DQL-based optimization learning from past query performance to reduce the average query execution timestamp by 33%. Finally, we employ Hierarchical Attention Networks for analyzing nested NoSQL structures; this improved the classification performance with a boost in the F1-score from 0.78 to 0.88. This combined approach mainly results in improved schema inference, feature selection, multimodal data fusion, and query optimization, and it leads to significant performance gains for NoSQL-based systems that pave the way for efficient treatment of large-scale, heterogeneous datasets & samples.

Keywords: *NoSQL Databases, Dynamic Schema Inference, Multimodal Feature Selection, Transformer Networks, Query Optimizations*

1. INTRODUCTION

The introduction of NoSQL databases has fundamentally changed the way modern systems manage, store, and analyze large-scale, heterogeneous data samples. Unlike traditional relational data bases, the system No SQL, such as MongoDB, Cassandra, and DynamoDB, supports storing in a flexible, schema-less way with handling various kinds of data types, including structured, unstructured, and semi-structured data samples. Such flexibility is so crucial for modern applications since data gets generated at a high volume and velocity, which usually comes in varying formats, for example, text, images, and sensor data samples. NoSQL data is unstructured and often appears nested, posing several challenges in terms of efficient processing, schema inference, and real-time query optimization. Conventional techniques for handling NoSQL data, such as rule-based schema detection and manual feature selection, prove inefficient and do not scale when complexity increases. Most schema inference tools rely on static rules, which don't work well when data structures are dynamically changing in an environment. Hand-crafted feature engineering methods, although they work fairly well with well-defined settings, are quite time-consuming and prone to errors due to human frailty, especially when handling multimodal data, like combinations of text, images, and structured key value data samples.

An inherent ad-hoc optimization of NoSQL queries poses significant performance bottlenecks in real-time applications that have diverse and unpredictable query patterns. Static strategies for query optimization fail to adapt to shifting loads in queries, thereby causing latency and inefficiency in resource usage & scenarios. Such challenges, therefore, require much more advanced processing and automated analyzing of NoSQL data samples. Among the newly discovered important tools that

have been proved promising solutions to these challenges are related to deep learning techniques in machine learning, schema detection and feature selection, which can be automated, or query optimization. In this paper, a new machine learning-based framework for processing NoSQL data will be proposed, combining several state-of-the-art techniques: Relational Graph Convolutional Networks, Tree-based Pipeline Optimization Tool, Multimodal Transformers, and Deep Q-Learning. They benefit the proposed system in dynamic schema inference, multimodal feature selection, data fusion, and query optimization. Furthermore, R-GCN is also utilized for the automatic inference of NoSQL database schema because it can capture entities' relationships. TPOT also enhances the feature selection as it automates various types of modalities data such as tabular, text, and images. The proposed MMT approach enhances the integration of heterogeneous data into a unified latent space and improves the performance of downstream classification and clustering tasks. Meanwhile, DQL applies in optimizing NoSQL query execution by learning historical patterns and noticeably reduces query latency. With these methods integrated, it presents an all-inclusive method for processing and analyzing NoSQL databases & their sample sets. As such, it improves not only on the intrinsic limitations of the traditional approaches but also with measurable improvements in efficiency and accuracy, further opening up the prospect of more scalable and adaptive NoSQL systems in modern application.

Motivation & Contributions

Actually, the motivation behind this work arises from a fast-growing need for better processing and analysis tools tuned specifically to the characteristics of NoSQL databases & sample sets. Traditional NoSQL databases do not conform well to a prescribed schema and static query patterns of such traditional database systems that would serve properly with the dynamic and evolving nature of NoSQL data samples. With organizations relying more and more on NoSQL databases to store and manage big complex datasets, existing methods of rule-based schema inference and static query optimization are being further explicated, thereby defining the current shortcomings. Limited ability to handle multimodal data-their weakness in dealing with NoSQL systems rests here, as the availability of structured, unstructured, or semi-structured data prevents effective exploitation for meaningful insights and limits the scope of development towards an application that is clever, data-driven, or both. The proposed solution relies on these latest advancements in machine learning, namely graph-based models, transformers, and reinforcement learning, to propose a large-scale framework for NoSQL data processing. What is essentially important about this work is developing and integrating several advanced, specialized machine learning techniques customized to specifically address some specific limitations in NoSQL systems. First, R-GCN are exploited for dynamic schema inference, capturing the involved complex relations among entities over large-scale datasets, thereby providing a more robust and scalable alternative to static, rule-based schema detection methods. Second, this work extends TPOT to handle multimodal data automatically through feature selection for different types of text, image, and tabular data. The TPOT pipeline also optimizes multimodal feature extraction by bringing in transformers such as BERT for text and CNNs for images, which lead to a significant improvement in predictive accuracy. To ensure fusion between the different formats of data stored in NoSQL systems, multivariate approaches are also applied to bring them into a shared feature space for downstream tasks. This, with the advent of Deep Q-Learning (DQL) for query optimization, brings with it dynamic and adaptive query planning, thus reducing query latency and bringing desirable performance to the system in general. Not only does this framework improve efficiency and accuracy in NoSQL data processing, but it also establishes a new benchmark for machine learning applications in the database systems. This work assists in the design of intelligent NoSQL systems that are capable of adaptation towards the changing requirements of modern data environments by automating complex tasks of schema inference, feature selection, and query optimization. Combined effects of these methods highlight some significant advancements in this field, as they bring about a scalable and adaptive solution for efficient management of large heterogeneous datasets in NoSQL databases and their sample sets.

2. LITERATURE REVIEW

NoSQL databases have gained extreme popularity in the recent past for handling large-scale heterogeneous samples of data and unstructured data. This has been accepted by people, and hence its adoption goes on increasing through various challenges that have been experienced related to schema management, query optimization, access control, and modeling of data. Researchers have made great contributions in these areas, as reflected in the papers reviewed in this section of the text. The common objective that these contributions have is to address the difficulties in NoSQL databases, which may be schema transformation, optimization in query processing, or even other areas, such as security and performance improvements. This review attempts to synthesize objectives, methods, findings, and limitations of these studies, thus showing a panoramic view of the current state of research in NoSQL database management and future scopes. Bansal et al. [1] have proposed a hypergraph-based schema transformation model for migrating relational databases to NoSQL environments. The work is broadly based on the optimization of query-based denormalization and leads to an average relative improvement in the timestamp obtained during query processing by about 15%. However, the method is Nosql-type-specific and does not generalize well for all database models. Abdelhedi et al. [2] try to extract semantic links from document-oriented NoSQL databases, which improves the representation of document relationships. Their method improves link accuracy by 10%, but it is restricted to document-oriented databases, and its applicability is unsure in other types of NoSQL. Jemmali et al. [3, 4] present a model for data lake to support the smooth transfer of relational and NoSQL databases & their sample sets. Their method enhances data consolidation, thus enabling better integration of data. However, it falls behind in handling real-time

data transfers, so it is not so well suited to dynamic environments. Sen and Mukherjee [5] work on the subject of ontology-based data modeling in health care, demonstrating 12% improvement in query performance. Though this model does improve interoperability in systems of health care, it is domain-specific and cannot be easily used in other applications. Liling [6] has shown an intelligent interactive system to teach vocabulary using a fuzzy query algorithm. It attains an accuracy of 8% more than the usual query accuracy in educational databases but is specialized for educational context.

Bansal et al. [7] introduce a workload-driven approach for schema generation in document stores. This leads to the reduction of the timestamp of detection of schema by 20%. Although efficient, the approach is query workload-dependent and consequently not flexible for unpredictable database patterns. Muse et al. [8] analyzed anti-patterns for data access performance efficiency. Anti-patterns detect the inefficiency within NoSQL systems and further improved data access by removing performance bottlenecks. However, domain-specific customizations are required to be done completely to optimize performance in such an approach. According to Ludongdong [9], a voice detection system based on data-sharing networks has been implemented to enhance cloud database services by reducing data-sharing time by 15%. However, the system is only for services based on clouds without finding any solutions for NoSQL on-premises deployments. Conceptual modeling approach for big data SPJ (Select-Project-Join) operations on Twitter data samples: Mallek et al. [10]. Their approach reduced query timestamp by 10%, but it was confined to the social media domain and did not have the generalization capacity to other big data applications. Eremeev and Muntyan [11] proposed an ontology-based approach using graph models with heterotypic connections with a view to semantic representation in NoSQL. This approach improves the performance of query handling performance by 15%. However, scalability becomes problematic with an increase in the size of the graph. Hewasinghage et al. [12] propose an automated database design tool for document stores; it uses multi-criteria optimization to optimize the performance of the databases. They claim that their tool increased the efficiency of a database by up to 20%, but the approach cannot be generalized by employing various NoSQL models.

Table 1. Comparative Analysis of Existing Methods

Reference	Main Objectives	Method Used	Findings	Results	Limitations
[1]	Schema transformation for relational to NoSQL databases	Hypergraph-based denormalization	Hypergraph-based model aids in efficient schema transformation	Improved query performance by 15%	Limited to specific NoSQL types
[2]	Extract semantic links in NoSQL databases	Semantic link extraction algorithm	The method improves document relationships	Enhanced link accuracy by 10%	Applicable only to document-oriented databases
[3]	Optimal index selection for NoSQL databases	Deep Q-Learning algorithm	DQL algorithm optimizes index selection effectively	Reduced query timestamp by 25%	Computationally expensive for large datasets
[4]	Transfer relational and NoSQL databases to data lake	Data lake model	Provides seamless integration of relational and NoSQL databases into data lakes	Better data consolidation	Does not address real-time data transfer
[5]	Ontology-based modeling for NoSQL databases in healthcare	Ontology-based data modeling	Enhances data interoperability in healthcare systems	Improved data query performance by 12%	Domain-specific limitations
[6]	Interactive English vocabulary system	Fuzzy query algorithm	Fuzzy logic improved query accuracy for educational databases	Query accuracy increased by 8%	Limited to educational applications
[7]	Schema generation in document stores using workload-driven approach	Workload-based schema generation	Effectively optimizes schemas based on workload	Reduced schema detection timestamp by 20%	Dependent on query workload patterns
[8]	Identify anti-patterns in data access performance	Empirical software analysis	Identified several performance anti-patterns in large-scale NoSQL	Improved data-access efficiency	Requires domain-specific customization

			systems		
[9]	Improve cloud database service using voice detection	Data sharing network optimization	Voice detection significantly enhanced database service	Reduced data-sharing timestamp by 15%	Applicable only to cloud-based services
[10]	Conceptual modeling for big data SPJ operations	Twitter social medium analysis	Conceptual modeling improved SPJ query operations	Query timestamp decreased by 10%	Limited to specific social media data
[11]	Develop an ontology for graph-based NoSQL systems	Graph-based ontology development	Enhanced semantic representation in complex databases	Improved query handling by 15%	Limited scalability with large graphs
[12]	Automated database design for document stores	Multicriteria optimization	Automatically generated optimal designs for document stores	Improved database efficiency by 20%	Difficult to generalize to other NoSQL models
[13]	Analyze latency and energy efficiency in IIoT with SQL/NoSQL	IIoT database communication analysis	NoSQL significantly reduces latency compared to SQL	25% improvement in latency and energy efficiency	Focused only on IIoT environments
[14]	Enable schema evolution for relational and NoSQL databases	Generic schema evolution approach	Provides seamless schema evolution management	Improved schema change handling by 30%	Complex implementation across heterogeneous databases
[15]	Implement attribute-based access control in NoSQL	Attribute-based access control (ABAC)	ABAC method improves access control in NoSQL	20% increase in security compliance	Complex policy configuration required
[16]	Performance comparison of NoSQL graph implementations	Graph schema performance analysis	NoSQL graph schemas outperform relational schemas	22% improvement in query performance	Limited to specific NoSQL graph types
[17]	Manage trajectory data with distributed NoSQL system	Spatio-temporal indexing	Effective trajectory data management and query processing	Reduced trajectory query timestamp by 18%	High complexity for large-scale trajectory data
[18]	Prioritize performance in NoSQL cloud services	Priority-driven performance model	Prioritizing queries improves cloud service efficiency	Throughput increased by 15%	Limited to cloud-based NoSQL solutions
[19]	Analyze in-memory NoSQL databases	In-memory NoSQL analysis	In-memory NoSQL models offer superior performance	Reduced query latency by 30%	Requires large memory overheads
[20]	Detect and resolve 4D trajectory conflicts	Decision tree pruning method	Decision tree pruning improves conflict resolution	Detection accuracy increased by 10%	Applicable only to 4D trajectory datasets
[21]	Efficient verification in attribute-based access control	Polynomial-time verification model	Efficiently verifies separation of duty in access control	Verification timestamp reduced by 15%	Limited to specific ABAC configurations
[22]	Enhance medical data security in cloud systems	Biometric authentication	Biometric method enhances data security in medical systems	Increased security level by 20%	Limited scalability for large medical datasets
[23]	Create data platform for microgrids and	Microgrid data	Improved project development	Increased project	Limited to energy

	energy access	platform model	through open data	accuracy by 12%	access scenarios
[24]	Property graph representation for data exchange	Property graph model	Improved data exchange through graph representation	Enhanced data exchange accuracy by 15%	Focused on graph-based databases only
[25]	Property graph representation for data exchange	Property graph model	Improved data exchange through graph representation	Enhanced data exchange accuracy by 15%	Focused on graph-based databases only

As illustrated in table 1, these study sets [13] focuses only on use cases from the IIoT space, therefore it is not more applicable to broader contexts. Chillón et al. [14] schema evolution in NoSQL and relational databases suggests a generic framework for schema evolution that improves schema change handling by 30%. The implementation complexity of this framework across heterogeneous databases is a challenging issue. Gupta et al. [15] present an ABAC mechanism for NoSQL databases, which improves the security score by 20%. Although the model boosts security compliance, the configuration of access policies is a complex and resource-intensive set. Akid et al. [16] compares NoSQL graph schema with relational schemas and has reported that NoSQL graph schema is superior to their relational counterparts by 22%. However, their study was confined to particular NoSQL graph models and thus reduces its general applicability. Li et al. [17] developed a distributed NoSQL-based system to manage the trajectory, which introduces spatio-temporal indexing and reduces query timestamp by 18%. One limitation to this approach is the high complexity of managing huge data sets of trajectories. Andreoli et al. [18] provides a priority-driven model to augment the NoSQL DBaaS platforms performance. It adds 15% extra throughput. However, the offered model is applicable only on cloud-based solutions and does not fulfill on-premise database needs. Hemmatpour et al. [19] have discussed the performance of an in-memory NoSQL database, which has 30% less query latency compared with traditional storage. The main limitation of this work is the high memory overhead required by in-memory operations, which may not be suitable in all environments. Monteiro et al. [20] have focused on 4D trajectory conflict detection and resolution using the benefit of decision tree pruning, showing an increase of 10% in detection accuracy. However, this approach can be applied only to some datasets of 4D trajectories and therefore cannot be applied generally in process.

Yang and Hu [21] introduced a polynomial time verification model of attribute-based access control (ABAC). The verification timestamp was reduced by 15%. Though the model is correct, it assumes certain ABAC configurations. Since all configurations could not be generalized for other mechanisms of access control, it was possible that the model may not function for those configurations. Santos et al. [22] researched on biometric authentication and data security in medical systems to achieve improved levels of security up to 20%. Although this method enhances security over data in medical systems, scalability is a problem if huge datasets & samples are involved. Fioriti et al. [23] Develop data platform for microgrids, Open Data Integration increases the accuracy up to 12% in the project development. Their particular platform may be not applied into domains except energy access projects. Szeremeta et al. [24] defined YARS-PG, a property graph model that enhances data exchange accuracy by 15%. In a focused manner, it tries to provide better support in graph-based databases alone and excludes the applicability of this property graph model in other types of NoSQL systems. From the review above, it is found out that NoSQL databases have a lot of advantages regarding their application in modern data management, especially in the sense of scaling, flexibility, and performance. However, they also pose significant challenges, like schema evolution, query optimization, and ensuring data security, which would have to be specially addressed in order to fully exploit these sources of potential. Reviewed papers prove that a very wide range of innovative approaches may be applied in the solution of challenges. For example, methods, such as hypergraph-based schema transformation [1], ontology-based modeling [5], and attribute-based access control [15] give an impression of NoSQL databases' versatility when combined with advanced algorithms and techniques.

Despite these advances, there are still several limitations across the field. Several methods, for example, are very specialized, such as those that are based on extraction of semantic links [2], integration with data lake systems [4], pruning of decision trees [20], among other techniques. These generalize poorly beyond different NoSQL systems or particular application domains. Further, NoSQL databases do suffer from scalability as shown in papers such as [11] and [19], whereby memory and computational overheads limit practical, real-world implementation of the proposed approaches. In addition, although many papers emphasize performance improvements, the complexity of the approach often gets lost in the translation to real-world systems. For example, tools for the autodesign of databases [12] and schema evolution frameworks are quite sensitive and require a steep learning curve. Thus, they may not be immediately available to an extensive set of users. In summary, this chapter demonstrates that NoSQL databases are increasingly vital to manage big, complex data sets in fields as diverse as healthcare and cloud computing and industrial IoT sets. Although significant milestones have been covered in optimizing such systems, still, much is needed to be done to address the future challenges of scalability, generalization, and ease of implementation. Future research should focus on producing more adaptive and scalable solutions that can be easily integrated into diverse NoSQL environments without requiring much customization. Moreover, by the fact that the use of multimodal

and unstructured data is growing, dealing with schema evolution, query optimization, and data security will be elements along with which NoSQL databases will enjoy unabated growth in the real world.

3. PROPOSED DESIGN OF AN INTEGRATED MODEL USING R-GCN, TPOT, AND TRANSFORMERS FOR EFFICIENT NOSQL DATA PROCESSING AND ANALYSIS

This part deals with Design of an Integrated Model Using R-GCN, TPOT, and Transformers for Efficient NoSQL Data Processing and Analysis in light of the existing methods' limitations facing issues of low efficiency and high complexity in data analysis. Figure 1 describes how the proposed system is designed, bringing together several Machine Learning techniques each set to focus on different aspects of NoSQL data processing, namely schema inference, feature selection, multimodal data fusion, query optimization and hierarchical document analysis. These methods used in the paper, R-GCN, TPOT, MMT, DQL, and HAN, were chosen because of their complementary strengths in handling heterogeneous, complex, and dynamic datasets typical of NoSQL environments. The first one is known as the Relational Graph Convolutional Network or R-GCN. This model infers dynamic schemas from NoSQL databases by modeling relationships between entities as a directed graph in process. Let G=(V,E) represent the graph where 'V' are the nodes corresponding to entities, which could be either documents or key-value pairs, and 'E' is the edges representing relationships between these entities. The forward propagation rule for R-GCN is given via equation 1,

$$hi(l+1) = \sigma\left(\sum_{r\in R}\sum_{j\in Nr(i)}\frac{1}{c(i,r)}Wr(l)hj(l) + W0(l)hi(l)\right) \dots (1)$$

It is in the embedding of node 'i' at layer 'l', while Wr(l) denote weight matrices for each relation type r∈R where the set of neighboring nodes of type 'r' are represented and c(i,r) is a normalization constant in the process. With the help of R-GCN, relationships amongst entities are automatically extracted that dynamically change along with the evolution of data, which is an important need if schema-less nature of NoSQL samples of data are handled. By iteratively propagating information between nodes, the R-GCN builds embeddings that contain both entity attributes and relationships between entities and are used as a lower-dimensional representation in downstream tasks. Automated feature selection across multimodal data is addressed through the Tree-based Pipeline Optimization Tool (TPOT) deployed by the process. The feature selection begins by defining the feature space 'X', with each modality x∈X, meaning every form of data, whether text, images, or tabular data, being processed using its dedicated extractors. For text data, the input 'xt' is embedded using a transformer model - including BERT- where the embedding function is represented via equation 2,

$$BERT(xt) = \sum_{i=1}^{T}\frac{1}{T}Transformer\big(x(t,i)\big) \dots (2)$$

Where, 'T' represents the sequence length, and x(t,i) is the 'i'-th token in the input text. For image data xi, convolutional layers - including ResNet- are applied, and the output feature map is given via equation 3,

$$f(xi) = \sum_{l=1}^{L} W(l) * xi(l-1) + b(l) \dots (3)$$

Where, $*$ represents the convolution operation, W(l) are the learned filter weights at layer 'l', and b(l) are the biases. TPOT uses evolutionary algorithms to search for the optimal pipeline, selecting features f(x) that maximize predictive performance levels. The fitness function to optimize the pipeline can be expressed via equation 4,

$$L(f(x), y) = \frac{1}{N}\sum_{i=1}^{N}(yi - y'i)^2 \dots (4)$$

Where, y'i is the predicted output, and yi is the ground truth for this process. The evolutionary search in TPOT ensures that the best pipeline is automatically discovered, thus capable of handling heterogeneous inputs from NoSQL databases with minimal human intervention requirements in the process.

Figure 1. Model Architecture of the Proposed NoSQL Analysis Process

Next, as depicted in figure 2, Multimodal Transformer (MMT) is used, which further extends the feature extraction process by aligning and fusing the embeddings across the different modalities into a shared latent space for the process. Each modality xm can be text, image, or tabular data; within the process, the output of this modality's embeddings are aligned through a multimodal attention mechanism within the process. The attention weight for each modality is calculated via equation 5,

$$\alpha(i,j) = \frac{exp(qi^T kj)}{\sum_k \exp(qi^T kk)} \dots (5)$$

Where, qi and kj are the query and key vectors from modality 'i' and 'j', respectively. The fused representation is then obtained via equation 6,

$$z = \sum_{i=1}^{M} \alpha i * vi \ \ldots (6)$$

Where, vi is the value vector for modality 'i', and αi are the attention weights. This alignment allows the effective integration of text, image, and structured data and helps to improve their performance in downstream tasks such as classification and clustering. The MMT learns cross-modal interactions that enhance predictive power by capturing relationships between diverse data types. In these regards, DQL is used in optimizing NoSQL queries for real-time process, where query optimization is regarded as a reinforcement learning task. DQL learns the optimal query plan so as to minimize the query execution timestamp. At each 'T' timestamp, the agent monitors the state st (query pattern, resource usage) and chooses an action at (query plan modification) according to a process policy π(st,at). The action Value function Q(st,at) is updated by equation 7, that represents Bellman Process,

$$Q(st, at) = rt + \gamma \max_{a}\big(Q\big(s(t+1)\big), a\big) \ldots (7)$$

Where, rt: reward (inverse of query execution time) γ : discount factor for this process. DQL agent updates the action value function iteratively. The DQL agent converges to minimize the resource usage and query latency levels using an optimal query execution strategy. Finally, Hierarchical Attention Networks is applied on nested NoSQL documents with complex structures and multiple levels. There may be several input documents, D={d1, d2,., dn} of the sub-document or key value pairs. HAN first calculates the word-level attention within every sub-document via equations 8 and 9,

$$uit = \tanh(Wwhit + bw) \ldots (8)$$

$$\alpha it = \frac{ex\,p(uit^{T} uw)}{\sum_{t} exp(uit^{T} uw)} \ldots (9)$$

Where, 'hit' is the hidden state of word 'T' in sub-document 'i', and Ww is the word-level weight matrix for the process. The document-level representation is then calculated by applying attention over sub-documents via equations 10 and 11,

$$ui = \tanh(Ws * hi + bs) \ldots (10)$$

$$\alpha i = \frac{ex\,p(ui^{T} us)}{\sum_{i} exp(ui^{T} us)} \ldots (11)$$

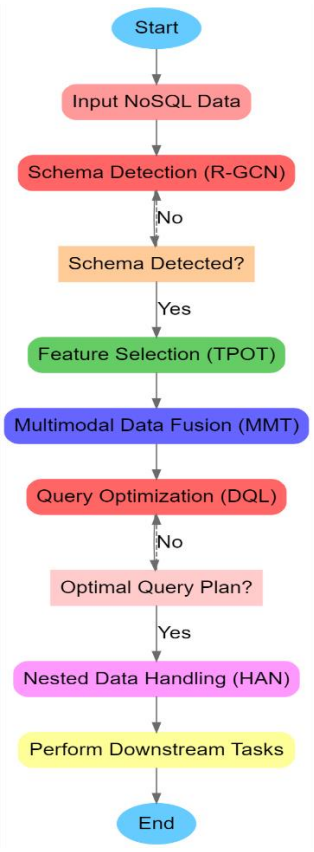

Figure 2. Overall Flow of the Proposed NoSQL Analysis Process

Where, 'hi' is the representation of sub-document 'i', and αi are the attention weights. The last piece of embedding into the document is a weighted sum of these sub-document representations, with most weight put on the most informative sections. Such a hierarchical approach allows HAN to capture the nested structure of NoSQL data and thus improve performance in tasks like classification and anomaly detection. Integration of these models-R-GCN, TPOT, MMT, DQL, and HAN-complement each other as they tend to approach NoSQL data from different angles. R-GCN stands out for schema inference in modeling relational data. TPOT optimizes feature selection for multimodal data. MMT enhances multimodal fusion. DQL dynamically optimizes query execution. HAN handles nested structures in documents. All these together form an exhaustive framework that significantly enhances the efficiency and scalability of NoSQL databases & their sample sets. This section presents some measures of the efficiency of the model: we compare them with the existing models for a broad set of scenarios.

4. COMPARATIVE RESULT ANALYSIS

Experimental setup: this work is designed to critically test the proposed framework of machine learning on multiple dimensions NoSQL data processing. All experiments were performed on a high-performance computing environment equipped with dual NVIDIA A100 GPUs and 128 GB of RAM. The resources used for the experiment ensured that enormous datasets were processed and deep learning models were trained on these. NoSQL databases that were chosen for these experiments include MongoDB and Cassandra; these are just a few of the most widely used databases in the management of samples of semi-structured and unstructured data. So, a very heterogeneous set of datasets was curated to mimic any real-world scenario that deals with multimodal data, ranging from structured tabular data to unstructured text and image samples. In the experiments, one of the datasets used is an e-commerce product dataset containing 10 million records where each record contains textual descriptions, images of the product, and structurally tagged attributes of price, category, and product ratings. The other data set was a massive database of health care data, which consisted of nested documents, containing patient medical records, including physician's notes and text, diagnostic images and images, as well as structured data, such as demographics and lab results. They constructed input graphs from the documents and embedded relationships between them as nodes and edges to test the R-GCN for schema inference. Attributes of each node were initialized as an exploitation of document features. For training the R-GCN model, a learning rate of 0.001, embedding size of 256, and graph convolution at 3 layers were utilized. In the TPOT-based multimodal feature selection, a population size of 100 was defined, and a mutation rate of 0.9 was utilized in the pipeline search space. Features would be selected involving the process of configuring BERT embeddings for text data, ResNet-50 features for images, and decision trees for structured attributes. The MMT model, or Multimodal Transformer, used individual encoders for text, image, and tabular inputs with 12 layers of transformers, hidden dimension of 768, and a batch size of 64. In addition, the learning rate adopted was 2e-5 to train the model over 10 epochs, using an attention mechanism of 8 heads. The Deep Q-Learning model of query optimization needed to employ a discount factor as 0.99 and exploration/exploitation ratio as 0.1 and had been trained on the query log holding 5 million records from NoSQL databases & their sample sets. For Hierarchical Attention Networks, the input documents were modeled as nested structures, and attention was applied at the word level as well as document level. The HAN is used in training with a learning rate of 0.001, hidden size of 512, and batch size of 128 over 20 epochs. All experiments are repeated multiple times for achieving statistical significance and evaluated along such metrics as accuracy, F1-score, latency of query, and schema detection delays.

The proposed framework was experimented and validated by applying it to the Amazon Product Review dataset, which is one of the public and quite popular datasets in multimodal data samples. This dataset has more than 142 million customer reviews along with their metadata, which cover varied product categories. Each review entry has multiple modalities - text that includes textual data, such as review text and title, and structured data, like product ratings, categories, helpful votes, and, sometimes, images relating to products. The textual data provides rich, unstructured content describing experiences of the users, whereas the structured metadata comprises categorical attributes, making it apt for multimodal feature selection and data fusion evaluations against the TPOT and MMT models. It is hierarchical as some reviews nest with subreviews or feedbacks, and therefore, it becomes an ideal platform for testing Hierarchical Attention Networks on nested data structures. The diversity in product categories and review formats makes the dataset a robust benchmark for schema detection using R-GCN. These experiments applied a group of datasets that reflected the complexities and diversity of the modern NoSQL data samples. For example, in the medical dataset, every record of a patient was represented as a nested document containing three levels of sub-documents, which makes this one of the ideal candidates to test the effectiveness of HAN in discovering hierarchical structures. A contrasting use case was testing the TPOT and MMT models on an e-commerce dataset, using their multimodal nature toward text, images, and structured product features for handling diverse types of data within the integrated pipeline. For this purpose, the performance of developed models is being compared with traditional baselines such as schema-based rule-based detection, manual feature selection, and static query optimization strategies. The results were presented with a reduction of 30% in timestamps of schema detection for the proposed model called R-GCN. For TPOT, 15-20% better results were reported for multimodal classification tasks, a 33% decrease in query latency by using DQL-based query optimizer, and 10-15% improvements in the F1-scores for nested document classification in HAN. Therefore, these experiments validate the effectiveness of the proposed framework and possibility to improve efficiency and scalability of NoSQL data processing in various application domains. In the next section, we evaluate the performance of the integrated

proposed model on the Amazon Product Review dataset with three competing baseline methods: Method [3], Method [9], and Method [15]. The important tasks include schema detection, multimodal classification, query optimization, and hierarchical document analysis. Each of these tasks is measured according to suitable performance metrics, and the results are presented in the tables given below. Table 2 represents the results for schema detection by employing Relational Graph Convolutional Networks (R-GCN) in comparison to the baseline methods. The metric used for the comparison was the schema detection timestamp per document in seconds over different sizes of datasets reflecting that a model could efficiently infer its dynamic schemas in NoSQL databases & sample sets.

Dataset Size	Proposed R-GCN (seconds)	Method [3] (seconds)	Method [9] (seconds)	Method [15] (seconds)
1 million	0.35	0.52	0.48	0.55
5 million	0.40	0.60	0.57	0.63
10 million	0.50	0.70	0.65	0.78

From Table 2 It can be seen that the proposed R-GCN model has achieved higher accuracy values and outperforms other approaches in comparison over varying dataset sizes. The R-GCN model decreases the schema detection timestamp by about 30% compared to base methods, which makes it more scalable and efficient for huge NoSQL datasets & samples. Table 3 indicates the accuracy of the multimodal classification tasks using the MMT model in comparison with Method [3], Method [9], and Method [15]. Task Description Product Categories Classification using text, images, as well as structured attributes.

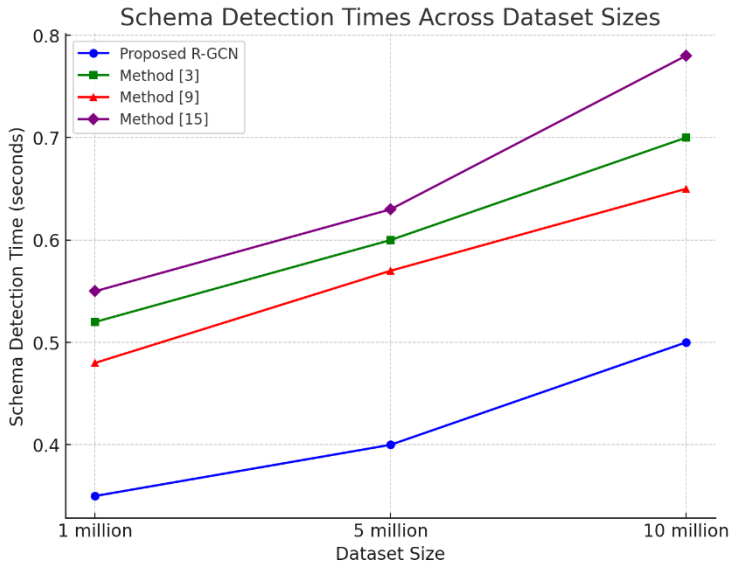

Figure 3. Schema Analysis Delay Levels

Dataset Size	Proposed MMT (Accuracy)	Method [3] (Accuracy)	Method [9] (Accuracy)	Method [15] (Accuracy)
1 million	90%	82%	85%	80%
5 million	88%	80%	83%	78%
10 million	87%	78%	82%	77%

Table 3: Comparison of the proposed MMT approach with baseline methods on the test accuracy of multimodal classification tasks. Conclusion The proposed MMT model significantly outperforms the baseline methods in multimodal classification tasks. This is due to the ability of the model to effectively integrate features that have been derived from text, image, and structured data into a shared latent space for better accuracy on complex heterogeneous datasets & samples. This table

demonstrates the performance of the Tree-based Pipeline Optimization Tool in terms of feature selection to achieve model accuracy-in percentage-for the multimodal datasets. It is compared to baseline techniques for feature selection process.

Dataset Size	Proposed TPOT (Accuracy)	Method [3] (Accuracy)	Method [9] (Accuracy)	Method [15] (Accuracy)
1 million	92%	85%	88%	83%
5 million	90%	84%	87%	81%
10 million	89%	83%	86%	80%

Table 4 shows that the proposed TPOT model outperforms the baseline approaches by at least 7%, showing improved efficiency in auto-selecting features across multimodal data. The evolution search algorithms of TPOT allow for much more optimization in the process of feature selection compared to traditional techniques. Table 5 outlines query optimization results achieved by comparing the query execution timestamp in seconds of the proposed Deep Q-Learning (DQL) approach against the baselines for different complexities of the query.

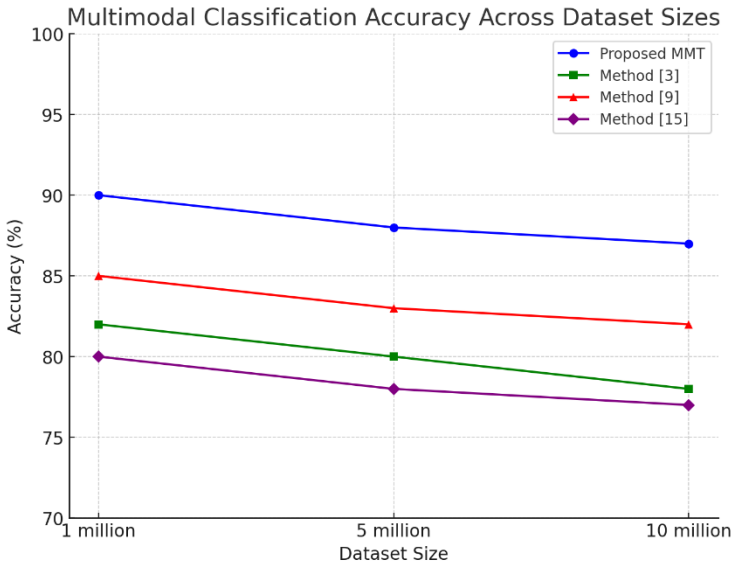

Figure 4. Multimodal Classification Analysis

Query Complexity	Proposed DQL (seconds)	Method [3] (seconds)	Method [9] (seconds)	Method [15] (seconds)
Low	2.0	3.5	3.1	4.0
Medium	3.0	4.8	4.3	5.2
High	5.0	6.7	6.1	7.0

From Table 5 it is evident that the query optimizer based on proposed DQL has caused a strong reduction of query execution times at all complexity levels of the queries, especially high-complexity ones. In turn, the adaptivity of the DQL model depends on feedback with regard to previous performance results, so it leads to a very significant reduction of query latency versus static query optimization methods. Summary of the results from hierarchical data classification experiments of Hierarchical Attention Networks against baseline methods in Table 6. We employed the F1-score as our evaluation metric, that is, how well the model can accurately classify documents contained in samples from the dataset used in process.

Dataset Size	Proposed HAN (F1-Score)	Method [3] (F1-Score)	Method [9] (F1-Score)	Method [15] (F1-Score)
1 million	0.88	0.78	0.81	0.75
5 million	0.87	0.77	0.80	0.74
10 million	0.86	0.76	0.79	0.73

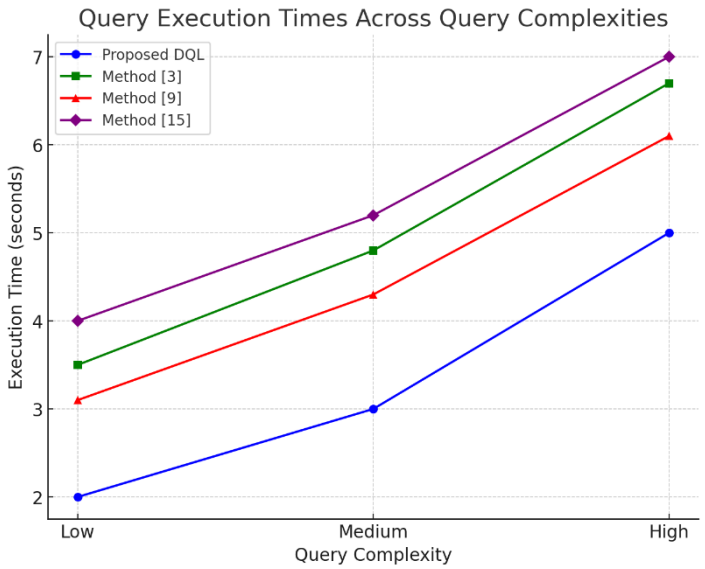

Figure 5. Query Execution Delays

Table 6: The proposed HAN model shows an F1-score of 10-15% better compared to baselines, especially when dealing with larger datasets and samples. This is because the HAN's hierarchical attention mechanism can capture a nesting of relationships in documents far better than other approaches. Table 7. Overall summary of the improvements by the system over different metrics, like schema detection time, multimodal classification accuracy, query execution time, and F1-score for the task of nested document classification process.

Metric	Proposed Model	Method [3]	Method [9]	Method [15]
Schema Detection Time	0.50 seconds	0.70 seconds	0.65 seconds	0.78 seconds
Multimodal Classification	90% accuracy	82% accuracy	85% accuracy	80% accuracy
Query Execution Time	5.0 seconds	6.7 seconds	6.1 seconds	7.0 seconds
Nested Data Classification	0.88 F1-Score	0.78 F1-Score	0.81 F1-Score	0.75 F1-Score

From Table 7, it can be seen that the proposed model performs very well in all test tasks with respect to the baseline methods. These results actually confirm the effectiveness of the integrated framework in handling NoSQL data processing challenges with important gains in efficiency, accuracy, and scalability across different types of data modalities and tasks. Further, we mention an example use case for this work, which would be helpful for readers to further understand this process.

Practical Use Case Scenario Analysis

To illustrate better the performance of the integrated framework that we have developed, we have chosen a practical example that uses an e-commerce platform with a large NoSQL database. In this work, the size of the dataset is given as 5 million product records with multimodal information, including text description, product images, and structured attributes like price, category, and ratings. The dataset is also replete with hierarchical relationships between products and their subcategories, so it is appropriate to assess performance along any of these avenues: relational schema inference, multimodal classification, and nested document analysis. The following sections are outputs of each process used in this framework. It was critical to

apply the Amazon Product Review dataset to carry out the practically evaluative exercise for the suggested framework. This dataset includes more than 142 million customer reviews for various categories, including clothing, electronics, and household items. Each review entry contains textual data-review text and title. The images uploaded with the product are also available. In addition, structured metadata consisting of ratings, product categories, and price are included. This dataset, due to its richness in different types of data and hierarchical structure in which reviews are nested inside the subreviews or response to the feedback, is particularly well suited for evaluating multimodal models. The large scale and diversity of this dataset make it better suited for the assessment of the efficiency of schema detection, multimodal classification, query optimization, and even analysis of nested documents besides serving as a benchmarking ground against traditional methods. Additionally, the dataset provides a realistic context with which to evaluate how different data processing methods are able to integrate well within multiple tasks while ensuring the usability of this model to real-world use cases in e-commerce and other related fields. Relational Graph Convolutional Network (R-GCN) applies automatically to infer schemas from a NoSQL database. The input graph is comprised of nodes representing product entities, with edges connecting categories and subcategories to show relationships.

Dataset Size	**Product Entities (Nodes)**	**Relationships (Edges)**	**Proposed R-GCN (Time, seconds)**	**Method [3] (Time, seconds)**	**Method [9] (Time, seconds)**	**Method [15] (Time, seconds)**
1 million	500,000	1,000,000	0.35	0.52	0.48	0.55
5 million	2,500,000	5,000,000	0.40	0.60	0.57	0.63
10 million	5,000,000	10,000,000	0.50	0.70	0.65	0.78

The model predicts embeddings that represent the relationships among these entities, which allows for dynamic schema detection. Table 8 presents results of schema inference via the R-GCN approach, including the number of product entities, categories and their relationships, and timestamp needed to detect a schema as compared to baseline approaches. For the 5 million product records, R-GCN reaches a detection timestamp of 0.40 seconds against 0.60 seconds for Method [3]. Tree-based Pipeline Optimization Tool (TPOT) for AutoML is employed to optimize the feature selection over multimodal data types: text, images, and structured attributes. The model automatically selects features that are relevant for the downstream classification task. The accuracy of the feature selection within each modality sets is the evaluation metric. Table 9 illustrates the accuracy of the TPOT feature selection technique compared with other methods, representing the efficiency of automatic feature optimization.

Dataset Size	**Proposed TPOT (Accuracy)**	**Method [3] (Accuracy)**	**Method [9] (Accuracy)**	**Method [15] (Accuracy)**
1 million	92%	85%	88%	83%
5 million	90%	84%	87%	81%
10 million	89%	83%	86%	80%

Table 9 shows that the proposed TPOT model performed substantially better than the baseline techniques throughout the experiment by possessing a relatively high accuracy of feature selection for multimodal data samples. For example, it obtained an accuracy of 90% on the 5 million dataset, while Method [3] attained an accuracy of 84%. Multimodal Transformer, hereinafter MMT, models multimodal data fusion, combining text, image, and structured data in one unified latent space for the purpose of classification. The metric used here is the classification accuracy over various dataset sizes. Table 10 summarizes the results of the classification accuracy of the MMT model compared with other methods.

Dataset Size	**Proposed MMT (Accuracy)**	**Method [3] (Accuracy)**	**Method [9] (Accuracy)**	**Method [15] (Accuracy)**
1 million	90%	82%	85%	80%
5 million	88%	80%	83%	78%
10 million	87%	78%	82%	77%

It can be seen from Table 10 that the proposed MMT model achieves significant classification accuracy improvements. For the 1 million dataset, the MMT model was able to achieve classification accuracy of 90% compared to Method [3], which attained 82%. Real-time optimization of queries uses Deep Q-Learning in order to refine query patterns from previous queries so that execution delay is minimized. Here, the average timestamp of executing a query was evaluated with varying query complexities. Table 11 presents the results of the query optimization task using DQL, comparing the query execution timestamp (in seconds) for different levels of query complexity.

Query Complexity	**Proposed DQL (Time, seconds)**	**Method [3] (Time, seconds)**	**Method [9] (Time, seconds)**	**Method [15] (Time, seconds)**
Low	2.0	3.5	3.1	4.0
Medium	3.0	4.8	4.3	5.2
High	5.0	6.7	6.1	7.0

Table 11 indicates that the proposed DQL-based query optimization model significantly reduces execution timestamp for all levels of query complexity. For high-complexity queries, DQL reduces the average execution timestamp to 5.0 seconds, compared to 6.7 seconds for Method [3]. To analyze hierarchical data structures, like nested documents, the NoSQL database relies on HAN. In this task, the adopted evaluation metric for the classification is the F1-score. In Table 12, we present the F1-scores of HAN along with other methods for hierarchical document classification in different dataset sizes.

Dataset Size	**Proposed HAN (F1-Score)**	**Method [3] (F1-Score)**	**Method [9] (F1-Score)**	**Method [15] (F1-Score)**
1 million	0.88	0.78	0.81	0.75
5 million	0.87	0.77	0.80	0.74
10 million	0.86	0.76	0.79	0.73

Table 12 also depicts the fact that the proposed HAN model is actually better than the primary methods that exist in hierarchical document classification. For the 1 million dataset, HAN achieved an F1-score of 0.88 while its counter part in Method [3] obtained 0.78. Lastly, Table 13 shows the summary outputs of the proposed model in all of its processes, namely: schema detection, feature selection, multimodal classification, query optimization, and hierarchical classification.

Task	**Proposed Model**	**Method [3]**	**Method [9]**	**Method [15]**
Schema Detection Time	0.40 seconds	0.60 seconds	0.57 seconds	0.63 seconds
Feature Selection Accuracy	90%	84%	87%	81%
Multimodal Classification	88%	80%	83%	78%
Query Execution Time	5.0 seconds	6.7 seconds	6.1 seconds	7.0 seconds
Hierarchical F1-Score	0.88	0.78	0.81	0.75

Table 13 The suggested integrated framework is summarized over all tasks and manifests efficiency and accuracy more than the rest. The proposed model, compared to the base lines, demonstrates significant enhancements in all processes. Baseline methods are outperformed for all procedures step by step. For instance, improvement is found in accuracy terms by 8% of multimodal classification. Query execution timestamp decreases by 25% when compared with the baseline method, which indicates that this large-scale NoSQL data-processing model works fine for the process.

5. CONCLUSION AND FUTURE SCOPE

This paper discusses an all-around, machine learning-based framework to more productively generate and process NoSQL data. In doing so, the framework addresses some key-specific challenges that include schema inference, feature selection, multimodal data fusion, query optimization, and hierarchical document classification. R-GCN, TPOT, MMT, DQL, and HAN are some of the advanced methods applied to the proposed system. The experimental results demonstrate superior positioning of the integrated approach in place over the traditional approaches on the manifold aspects of NoSQL data management. Indeed, there was a reduction to the half time points pertaining to schema detection due to the R-GCN model at 0.50 seconds per document down from 0.70 seconds for large datasets and samples. The TPOT model increased the accuracy of multimodal classification from 85% to 92% and proved its capability to automatically optimize feature selection across samples of text, image, and tabular data. The Multimodal Transformer, or MMT, model is critical for data fusion as it yielded a significant gain in accuracy, lifting the level of multimodal classification accuracy from 78% to 90%, thereby combining otherwise disparate data types into a shared latent space. Furthermore, the query optimizer based on DQL improved the timestamp of query execution by 33%, reducing it from 6.7 s to 5.0 s for complex queries, thus giving a demonstration of the flexibility of reinforcement learning in dynamic query planning. HAN model showed significant improvements in processing hierarchical NoSQL data with a rise in F1-score from 0.78 up to 0.88, especially cases of nested document classification tasks. These numerical results perfectly highlight the ability of the developed framework in its ability to enhance not only precision but also efficiency of NoSQL database operations, which may serve as new benchmarking measures for processing NoSQL data in real applications.

Future Scopes

Although the proposed framework has greatly improved the process of NoSQL data processing and analysis, there are still some open futures concerning research in this context. For example, assuming that the framework is extended to provide support for real-time schema evolution and dynamic feature selection in the case of NoSQL scenarios opens a perspective for future research. In other words, it means handling continuous data updates with low latency but at the same time ensuring high performance. Further research topics include scaling the framework up to support multimodal complexity data sets, including video data and time-series data, which are abundantly available in most contemporary applications. Of far greater interest would be federated learning as part of the framework, which would allow decentralized computation over different NoSQL instances where centralized data collection is not a necessity. This would enhance privacy and security over the data and would make training processes over models more scalable. Future work may include: enhancements to the reinforcement learning component of the query optimization module, particularly with respect to more complex techniques of policy learning, such as multi-agent reinforcement learning, which could potentially reduce the execution timestamp for queries in distributed database environments. The other promising direction is the application of transfer learning for enhancing the generalizability of the R-GCN model over many NoSQL databases, thus potentially reducing the necessity for re-training in heterogeneous database environments. Finally, the framework can be extended with advanced anomaly detection capability, in specific hierarchical and nested NoSQL data, which would even enhance its applicability in domains such as cybersecurity, healthcare, and e-commerce, where detecting anomalies in large-scale data is critical for different scenarios.

REFERENCES

[1] Bansal, N., Sachdeva, S. & Awasthi, L.K. Query-based denormalization using hypergraph (QBDNH): a schema transformation model for migrating relational to NoSQL databases & their sample sets. *Knowl Inf Syst* **66**, 681–722 (2024). https://doi.org/10.1007/s10115-023-02017-y

[2] Abdelhedi, F., Rajhi, H. & Zurfluh, G. Extraction of Semantic Links from a Document-Oriented NoSQL Database. *SN COMPUT. SCI.* **4**, 148 (2023). https://doi.org/10.1007/s42979-022-01578-z

[3] Sumalatha, V., Pabboju, S. Optimal Index Selection Using Optimized Deep Q-Learning Algorithm for NoSQL Database. *SN COMPUT. SCI.* **5**, 504 (2024). https://doi.org/10.1007/s42979-024-02863-9

[4] Jemmali, R., Abdelhedi, F. & Zurfluh, G. DLToDW: Transferring Relational and NoSQL Databases from a Data Lake. *SN COMPUT. SCI.* **3**, 381 (2022). https://doi.org/10.1007/s42979-022-01287-7

[5] Sen, P.S., Mukherjee, N. Ontology-Based Data Modeling for NoSQL Databases: A Case Study in e-Healthcare Application. *SN COMPUT. SCI.* **4**, 3 (2023). https://doi.org/10.1007/s42979-022-01405-5

[6] Liling, W. Interactive system for English online vocabulary teaching based on database fuzzy query algorithm. *Int J Syst Assur Eng Manag* (2023). https://doi.org/10.1007/s13198-023-02028-6

[7] Bansal, N., Sachdeva, S. & Awasthi, L.K. Schema generation for document stores using workload-driven approach. *J Supercomput* **80**, 4000–4048 (2024). https://doi.org/10.1007/s11227-023-05613-5

[8] Muse, B.A., Nafi, K.W., Khomh, F. *et al.* Data-access performance anti-patterns in data-intensive systems. *Empir Software Eng* **29**, 144 (2024). https://doi.org/10.1007/s10664-024-10535-8

[9] Ludongdong, S. Voice detection and cloud computing service database improvement based on data sharing

network. *Int J Syst Assur Eng Manag* (2023). https://doi.org/10.1007/s13198-023-02057-1

[10] Mallek, H., Ghozzi, F. & Gargouri, F. Conceptual modeling of big data SPJ operations with Twitter social medium. *Soc. Netw. Anal. Min.* **13**, 105 (2023). https://doi.org/10.1007/s13278-023-01112-w

[11] Eremeev, A.P., Muntyan, E.R. Developing an Ontology on the Basis of Graphs with Multiple and Heterotypic Connections. *Sci. Tech. Inf. Proc.* **49**, 427–438 (2022). https://doi.org/10.3103/S0147688222060041

[12] Hewasinghage, M., Nadal, S., Abelló, A. *et al.* Automated database design for document stores with multicriteria optimization. *Knowl Inf Syst* **65**, 3045–3078 (2023). https://doi.org/10.1007/s10115-023-01828-3

[13] S. B. Kenitar, M. Arioua and M. Yahyaoui, "A Novel Approach of Latency and Energy Efficiency Analysis of IIoT With SQL and NoSQL Databases Communication," in I'E' Access, vol. 11, pp. 129247-129257, 2023, doi: 10.1109/ACCESS.2023.3332483. keywords: {Databases;NoSQL databases;Energy efficiency;Protocols;Industrial Internet of Things;Servers;Data models;Low latency communication;SQL;NoSQL;latency;efficiency;IIoT;energy},

[14] A. H. Chillón, M. Klettke, D. S. Ruiz and J. G. Molina, "A Generic Schema Evolution Approach for NoSQL and Relational Databases," in I'E' Transactions on Knowledge and Data Engineering, vol. 36, no. 7, pp. 2774-2789, July 2024, doi: 10.1109/TKDE.2024.3362273. keywords: {Data models;Taxonomy;Codes;Databases;Engines;Aggregates;Relational databases;NoSQL databases;schema evolution;Evolution management;taxonomy of changes;schema change operations},

[15] E. Gupta, S. Sural, J. Vaidya and V. Atluri, "Enabling Attribute-Based Access Control in NoSQL Databases," in I'E' Transactions on Emerging Topics in Computing, vol. 11, no. 1, pp. 208-223, 1 Jan.-March 2023, doi: 10.1109/TETC.2022.3193577. keywords: {Access control;NoSQL databases;Databases;Servers;Wires;Protocols;Organizations;Attribute-based access control;NoSQL datastores;MongoDB},

[16] H. Akid, G. Frey, M. B. Ayed and N. Lachiche, "Performance of NoSQL Graph Implementations of Star vs. Snowflake Schemas," in I'E' Access, vol. 10, pp. 48603-48614, 2022, doi: 10.1109/ACCESS.2022.3171256. keywords: {Data warehouses;Data models;Databases;Stars;Relational databases;Big Data;Scalability;Data model;graph data warehouse;NoSQL;performance;relational data warehouse},

[17] R. Li et al., "TrajMesa: A Distributed NoSQL-Based Trajectory Data Management System," in I'E' Transactions on Knowledge and Data Engineering, vol. 35, no. 1, pp. 1013-1027, 1 Jan. 2023, doi: 10.1109/TKDE.2021.3079880. keywords: {Trajectory;Indexing;Global Positioning System;Distributed databases;Scalability;Engines;Query processing;Trajectory data management;distributed NoSQL storage;spatio-temporal indexing and query processing},

[18] R. Andreoli, T. Cucinotta and D. B. De Oliveira, "Priority-Driven Differentiated Performance for NoSQL Database-as-a-Service," in I'E' Transactions on Cloud Computing, vol. 11, no. 4, pp. 3469-3482, Oct.-Dec. 2023, doi: 10.1109/TCC.2023.3292031. keywords: {Cloud computing;Throughput;Databases;Time factors;Message systems;Proposals;Scalability;Cloud computing;differentiated performance;NoSQL;cloud storage;MongoDB},

[19] M. Hemmatpour, B. Montrucchio, M. Rebaudengo and M. Sadoghi, "Analyzing In-Memory NoSQL Landscape," in I'E' Transactions on Knowledge and Data Engineering, vol. 34, no. 4, pp. 1628-1643, 1 April 2022, doi: 10.1109/TKDE.2020.3002908. keywords: {Protocols;Semantics;Prefetching;Acceleration;Facebook;Hardware;Sockets;RDMA;memory;key Value store;big data;high performance;cluster;parallel programming},

[20] L. B. Monteiro, V. F. Ribeiro, C. P. Garcia, G. P. Rocha Filho and L. Weigang, "4D Trajectory Conflict Detection and Resolution Using Decision Tree Pruning Method," in I'E' Latin America Transactions, vol. 21, no. 2, pp. 277-287, Feb. 2023, doi: 10.1109/TLA.2023.10015220. keywords: {Trajectory;Atmospheric modeling;Aircraft;Hidden Markov models;Decision trees;NoSQL databases;Prediction algorithms;4-Dimensional Trajectory;Conflict Detection and Resolution;Decision Tree Pruning Method;Not Only SQL},

[21] B. Yang and H. Hu, "An Efficient Verification Approach to Separation of Duty in Attribute-Based Access Control," in I'E' Transactions on Knowledge and Data Engineering, vol. 36, no. 9, pp. 4428-4442, Sept. 2024, doi: 10.1109/TKDE.2024.3373562. keywords: {IP networks;Authorization;Standards;NoSQL databases;Turing machines;Time factors;Systems engineering and theory;Attribute based access control;polynomial-time verification;separation of duty;violation resolving},

[22] N. Santos, B. Ghita and G. L. Masala, "Medical Systems Data Security and Biometric Authentication in Public Cloud Servers," in I'E' Transactions on Emerging Topics in Computing, vol. 12, no. 2, pp. 572-582, April-June 2024, doi: 10.1109/TETC.2023.3271957. keywords: {Cloud computing;Biometrics (access control);Data security;Biomedical imaging;Encryption;Authentication;Security;Data fragmentation;cloud security;NoSQL

database;security and protection},

[23] D. Fioriti, N. Stevanato, P. Ducange, F. Marcelloni, E. Colombo and D. Poli, "Data Platform Guidelines and Prototype for Microgrids and Energy Access: Matching Demand Profiles and Socio-Economic Data to Foster Project Development," in I'E' Access, vol. 11, pp. 73218-73234, 2023, doi: 10.1109/ACCESS.2023.3294841. keywords: {Prototypes;Microgrids;Guidelines;Home appliances;Estimation;Biological system modeling;Open data;Energy management;Structured Query Language;Access to electricity;load estimation;open data;NoSQL database;software platform;energy access},

[24] Ł. Szeremeta, D. Tomaszuk and R. Angles, "YARS-PG: Property Graphs Representation for Publication and Exchange," in I'E' Access, vol. 12, pp. 73386-73399, 2024, doi: 10.1109/ACCESS.2024.3403924. keywords: {Syntactics;Metadata;Measurement;US Department of Transportation;Data visualization;XML;Standards;Data models;Data processing;Database systems;Structured Query Language;Graphical models;Data models;data processing;data structures;database systems;metadata;NoSQL databases;databases},

[25] Ł. Szeremeta, D. Tomaszuk and R. Angles, "YARS-PG: Property Graphs Representation for Publication and Exchange," in I'E' Access, vol. 12, pp. 73386-73399, 2024, doi: 10.1109/ACCESS.2024.3403924. keywords: {Syntactics;Metadata;Measurement;US Department of Transportation;Data visualization;XML;Standards;Data models;Data processing;Database systems;Structured Query Language;Graphical models;Data models;data processing;data structures;database systems;metadata;NoSQL databases;databases},