Computer Systems
Science and Engineering

Tech Science Press

DOI: 10.32604/csse.2024.051851

REVIEW

A Systematic Review of Automated Classification for Simple and Complex Query SQL on NoSQL Database

Nurhadi, Rabiah Abdul Kadir*, Ely Salwana Mat Surin and Mahidur R. Sarker*

Institute of Visual Informatics, The National University of Malaysia, Selangor, 43600, Malaysia

*Corresponding Authors: Rabiah Abdul Kadir. Email: rabiahivi@ukm.edu.my; Mahidur R. Sarker. Email: mahidursarker@ukm.edu.my

Received: 17 March 2024 Accepted: 08 August 2024 Published: 22 November 2024

ABSTRACT

A data lake (DL), abbreviated as DL, denotes a vast reservoir or repository of data. It accumulates substantial volumes of data and employs advanced analytics to correlate data from diverse origins containing various forms of semi-structured, structured, and unstructured information. These systems use a flat architecture and run different types of data analytics. NoSQL databases are nontabular and store data in a different manner than the relational table. NoSQL databases come in various forms, including key-value pairs, documents, wide columns, and graphs, each based on its data model. They offer simpler scalability and generally outperform traditional relational databases. While NoSQL databases can store diverse data types, they lack full support for atomicity, consistency, isolation, and durability features found in relational databases. Consequently, employing machine learning approaches becomes necessary to categorize complex structured query language (SQL) queries. Results indicate that the most frequently used automatic classification technique in processing SQL queries on NoSQL databases is machine learning-based classification. Overall, this study provides an overview of the automatic classification techniques used in processing SQL queries on NoSQL databases. Understanding these techniques can aid in the development of effective and efficient NoSQL database applications.

KEYWORDS

NoSQL database data lake; machine learning; ACID; complex query; smart city

1 Introduction

The data lake (DL) procedure commences with data ingestion and concludes with data transformation for analysis, with each element of the data being delineated within the DL ecosystem [1]. This system enables the storage of vast quantities of data, which can be in diverse types and formats. A smart city comprises various elements employing DL storage technology for storing vast data volumes, incorporating a NoSQL database that serves as a dataset comprising multiple data groupings [2]. A DL is a storage infrastructure crafted to accommodate diverse data formats, such as structured, semi-structured, unstructured, and binary data while aligning with the four key aspects of big data, namely Volume, Velocity, Variety, and Veracity (4V) [3,4]. NoSQL databases have the capacity to accommodate diverse data types; however, they lack complete support for the atomation, integrity, isolation, and durability (ACID) features, such as trigger functions in multi-transaction management

[1,5] because they use a non-relational database management system (RDBMS) [6,7]. The NoSQL database encompasses four database types: document-based, key-value store, graph store, and wide column store [8]. According to [2], NoSQL technology is now a common part of a large number of information systems and software applications. Meanwhile, according to [3], the technology's primary focus is on performance, enabling the efficient transmission of large volumes of unstructured and structured data. Some features that distinguish NoSQL are high-performance writing, large scalability, and free writing schemes. Nevertheless, to attain horizontal scalability, NoSQL databases lack the conventional ACID properties commonly offered by relational databases [4,5], and NoSQL databases are less supportive of the ACID features. Certain databases loosen the data consistency and freshness of the data to support scalability and durability [6,9]. Suitable tools are required to develop a new translation algorithm aimed at enhancing complex function queries, specifically those related to ACID properties. Despite the lack of support for ACID transactions and the elimination of the need for fixed schemas, NoSQL databases use a schema-free approach where schemas are considered flexible concepts, allowing instances related to the same concept to be stored using multiple local schemas [7,8]. This issue has been a longstanding challenge in the processing and analysis of online transactions. Classification is essential when applying intricate SQL queries in a NoSQL database to aid in the translation and categorization from SQL to NoSQL due to the varying data formats—structured, semi-structured, and unstructured [10].

ACID features and the trigger function are two important features in managing data in an RDBMS or a NoSQL database. Atomicity and consistency are used to ensure that every transaction in the database will be correctly and consistently carried out. Isolation ensures that each transaction is not affected by other ongoing transactions, while durability guarantees that any data changes made to the database will survive [9]. The trigger function, which is part of the ACID function, is used to activate certain actions when an event occurs in the database, such as when the data is changed, deleted, or added. In this study, the use of the trigger function will be tested in conjunction with complex queries involving other ACID functions to see how far these two features can be utilized together in data management in NoSQL databases [6]. On this basis, machine learning must be applied in the classification of simple and complex queries on the RDBMS and NoSQL databases to improve the efficiency and accuracy in processing these queries. In an RDBMS, machine learning methods can be employed to distinguish between straightforward and intricate SQL inquiries. For example, simple queries, such as SELECT, INSERT, UPDATE, and DELETE, can be grouped into one category. Meanwhile, complex queries, such as subqueries, joins, triggers, union operation queries, and aggregations, can be grouped into different categories. This category can be used to select the best algorithm to process the query, thereby increasing efficiency in processing it [10]. Meanwhile, in NoSQL databases, machine learning can implement complex query functions, such as those in RDBMS, by classifying these queries based on their type and structure.

The objective of this research, utilizing the Preferred Reporting Items for Systematic Reviews and Meta-Analysis (PRISMA) methodology, is to provide a clear and precise systematic review of a specific subject, following a predetermined methodological framework [11]. The PRISMA method is designed to help obtain and evaluate relevant research systematically and objectively and clarify how the research is carried out. This method helps ensure that research conducted in a systematic review is of high quality and meets the predetermined criteria [12]. The use of the PRISMA method in this research can help ensure the accuracy, objectivity, and transparency of systematic reviews [13], thereby allowing them to have a significant contribution to the development of knowledge in the field of classification of simple and complex queries on the SQL on NoSQL Databases with a machine learning approach.

This paper is presented in five major sections. The second section presents a rough methodology and how research is carried out. The third section presents the DL and the ecosystem within it, namely, the RDBMS and NoSQL databases. The fourth section discusses how to classify machine learning in the RDBMS and NoSQL databases. The fifth section highlights the techniques used for classification in complicated SQL queries in resolving the problem of trigger and ACID functions. The sixth section concludes the paper. This study provides a comprehensive description of the automatic classification of SQL queries in NoSQL databases. This research fills in existing knowledge gaps and provides insight into the latest developments in the field of NoSQL databases which include: 1. Gap: The existing literature lacks comprehensive exploration and investigation into the automation of classifying SQL queries within NoSQL databases. There is a noticeable deficiency in research focusing on this automatic classification technique.

This study aims to address this disparity by exploring the intricacies of automated query classification within the domain of NoSQL databases. 2. Objective: The primary goal of this study is two-fold. Firstly, it seeks to identify and clarify the current techniques employed for automating the classification of both simple and complex SQL queries within NoSQL databases. Secondly, it seeks to provide insightful suggestions for further research endeavors in this domain. By achieving these objectives, this research endeavors to enhance understanding and implementation in the automation of SQL query classification within NoSQL environments. 3. Suggestion: There is a pressing need to explore alternatives utilizing machine learning and artificial intelligence methodologies to improve and optimize the classification process further. Additionally, expanding the scope of research to encompass various types of NoSQL databases and encompassing more complex SQL queries can yield invaluable insights and broaden the applicability of findings. By embracing a diverse range of methodologies and scenarios, researchers can acquire a thorough comprehension of the difficulties and opportunities inherent in automating query classification within the dynamic landscape of NoSQL databases.

The motivation for this research is that in the ever-growing digital era, data management and analysis are becoming increasingly important for organizations in various fields. With DL and the use of NoSQL databases, organizations can store and manage various types of data on a large scale and in various formats. However, the use of NoSQL databases also raises new challenges related to processing complex SQL queries. This problem arises because NoSQL databases do not always fully support features such as ACID which are usually found in relational databases. Therefore, an automated approach is needed to classify complex SQL queries in NoSQL databases, and the use of machine learning is a promising solution to this problem.

Meanwhile, the contribution of this research is to provide main contributions in two ways. First, we present a systematic review of automatic classification techniques for SQL queries in NoSQL databases, with a focus on the use of machine learning in this context. This systematic review will provide an in-depth understanding of the techniques that have been used in the literature, as well as identify emerging trends and patterns. Second, we propose a new method to improve the efficiency and accuracy of SQL query processing in NoSQL databases by utilizing machine learning approaches. It is hoped that this method can provide practical guidance for developers and researchers in selecting appropriate classification techniques for their needs. One of the defining aspects of this research is its originality in addressing the gap in the current literature concerning the automatic classification of complex SQL queries in NoSQL databases using machine learning techniques. To the best of our knowledge, no prior studies have comprehensively explored the integration of machine learning for the classification of both simple and complex SQL queries specifically within the context of NoSQL databases, including their unique ACID and trigger functionalities. This research stands out by not only proposing a novel methodology for efficient query classification but also by demonstrating its

practical applicability and potential benefits for data management in the evolving landscape of NoSQL database systems.

The main motivation of this research is to address the challenges faced in SQL query classification in NoSQL databases. With the increasing complexity of data and the need for efficient and accurate processing, this research aims to provide in-depth insights and practical solutions in the application of query classification automation techniques. The main contributions of this research are two things. First, this research presents a systematic review of automatic classification techniques for SQL queries in NoSQL databases, with a focus on the use of machine learning in this context. This systematic review will provide an in-depth understanding of the techniques that have been used in the literature, as well as identify emerging trends and patterns. Second, this research proposes a new method to improve the efficiency and accuracy of SQL query processing in NoSQL databases by leveraging machine learning approaches.

2 Methodology

2.1 Research-Questions

RQs are important for conducting research and effectively analyzing data. The questions listed below are reviewed and answered in detail in Sections 3–6.

- RQ1. How to handle the implementation of trigger function transactions involving complex statements in the management of DL in smart city?
- RQ2. Where should machine learning be applied to implement the trigger functions in the NoSQL database?

2.2 Search Strategy

Four electronic databases, namely, ACM Digital Library, Google Scholar, Springer, and IEEE Xplore, to identify pertinent papers for this systematic review, of the ACID properties were utilized. A concise examination of the study was performed to ascertain suitable search terms. These databases possess access to comprehensive information, yet they cannot extract value from it due to their semi-structured or unstructured nature, which was produced in the terms 'Trigger function NoSQL Database in Data Lake Smart City' and 'Implementation Trigger function of SQL Complex Classification using a Learning Machine'. Another term set was recognized to guide our investigation toward research employing comparable terms in categorizing various machine learning methods.

2.3 Inclusion Criteria

Studies that used machine learning methods for classifying complex transactions with trigger functions in the ACID in the NoSQL Database and published either in journals or as conference proceedings are deemed eligible for inclusion. Papers written in English are also considered. A year filter is applied to choose the main study published from 2016 to the present. The study discussed the classification using machine learning to leverage the NoSQL Database in DL in smart city. Papers that describe the implementation of learning machines for the classification of complex query transactions are included in the research. This systematic literature review excludes guidelines, case reports, review articles, and abstracts from conference papers.

2.4 Criteria for Assessing Quality

The eligibility of the studies was determined by excluding those that obtained a score lower than a predefined threshold 'quality threshold'. This approach helps in distinguishing studies based on their overall contribution. Analysis and Content: Was the content technically sound and substantiated by evidence and theories offering advantages compared to the presented approaches?; Novelty: What degree of originality does the proposed concept exhibit, or is it simply an enhancement of an existing iteration?; Results: Was the outcome effectively presented and contrasted with the benchmark dataset? Each study received a score out of 10, allocated as follows: 2 for novelty, 6 for content and analysis, and the remaining 2 for outcomes from two datasets (Google Scholar, Springer, IEEE Xplore, and ACM Digital Library). The implementation of quality assessment criteria led to a reduced number of papers. Studies considered of poor or below-average quality would be excluded from the final selection.

The method used in this study is PRISMA: a framework designed to assist researchers in reporting the results of systematic reviews and meta-analyses systematically and transparently. The PRISMA flow diagram depicts the flow of study selection from the initial search database to the number of studies selected to be included in the final review. This diagram consists of several boxes and arrows that describe the stages of study selection as shown in Fig. 1 below.

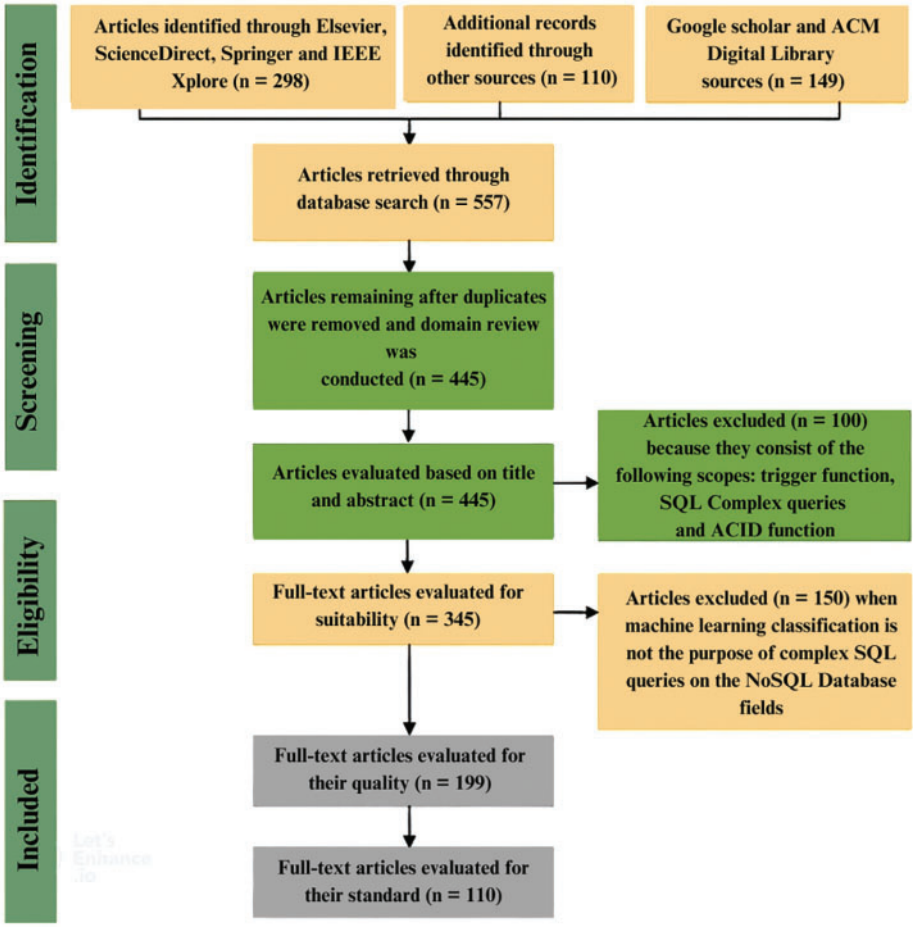

Figure 1: PRISMA guidelines of this systematic review

The PRISMA workflow described above can be explained in the following stages:

- *Identification*: Study searches were conducted through various databases or other literature sources, such as Springer, Google Scholar, IEEE Xplore, and ACM Digital Library. Henceforth, the number of initial studies found is recorded.
- *Screening*: This stage involves reviewing the title and abstract of all the studies that were identified from the previous stage. The aim is to determine whether the study is relevant to the research topic set and meets the criteria for inclusion and exclusion. Studies that are irrelevant or do not meet the criteria will be excluded from the selection. The scope of the selection process will be restricted to the following: trigger functions, Complex SQL queries, and ACID functions.
- *Eligibility*: In this case, the determination of eligibility is according to the predetermined inclusion and exclusion criteria, referring to the criteria used to determine whether a study is suitable for inclusion in a systematic review or meta-analysis. Specifically, a study will be excluded if the objective of a complex SQL query on a NoSQL database field is not related to machine learning classification.
- *Included*: This term refers to studies that have met the inclusion criteria and have been selected for inclusion in the data synthesis or analysis in a systematic review or meta-analysis. The studies incorporated into a systematic review or meta-analysis are expected to provide a significant and quality contribution toward solving the research problem under investigation.

2.5 Data Extraction

To accomplish this objective, a mapping exercise was carried out. Information regarding the author, publication year, publisher, methodology, dataset, description, and results of the approach proposed in the selected studies was extracted. The purpose of this phase is to gather data from the chosen studies and ascertain which ones address the identified research questions.

2.6 Combining Data

The procedure of gathering and synthesizing data from articles entails acquiring pertinent information to address the research question and amalgamating all the gathered information. In terms of data, we extract the techniques or algorithms used for the classification of simple and complex queries, the datasets used, the machine learning methods, the details and scope of research, and the best results and solutions achieved.

2.7 Results

After applying inclusion and exclusion criteria, all studies that meet the specified criteria are selected for further analysis. The number of research that meets the quality assessment criteria is recorded. The data extracted during the process of extraction and synthesis is displayed in a tabular format, depicted in Fig. 1 (flow based on PRISMA). The final results of the review are summarised to answer RQ1 and RQ2.

3 DL Ecosystem in a Smart City

During the data synthesis, we divide the analysis into three main topics. The first topic focuses on finding the classification requirements for simple and complex SQL queries and complex NoSQL queries in DL in Smart City Management. The second topic highlights the role of NoSQL as part of the DL in the Smart City Management Framework. The third topic discusses the methods or frameworks used for machine learning-based classification in the NoSQL database.

The following details each element in this study: 1. Research Gap: This research was conducted to fill in the existing knowledge gaps in the literature regarding classification automation in SQL queries on NoSQL databases. 2. Objectives: To identify existing techniques for automating the classification of simple and complex SQL queries in NoSQL databases, and to assess the advantages and disadvantages of the techniques employed. 3. Approaches: This study used a systematic review approach by identifying specific research questions and clear inclusion and exclusion criteria, conducting a systematic and comprehensive literature search through relevant databases. Suggestion: can provide some suggestions as the direction of further research, namely; Develop more sophisticated and efficient techniques for automating the classification of SQL queries on NoSQL databases, exploring machine learning and additional artificial intelligence techniques approaches in optimizing the classification process.

3.1 Data Lake

DL relies on big data technologies like Hadoop, Spark, and Yarn, enabling it to be scalable, distributed, reliable, and fault-tolerant. It functions as a data repository for vast quantities of diverse data in their native context [14,15]. The data undergo processing and transformation as needed. Additionally, this system is linked with advanced analytics, automation, orchestration, and machine intelligence tools and languages. DL comprises distinct yet interconnected components for storage, standardization, structuring, and processing [16]. The need for an effective dataset classification to facilitate data analysis and retrieval of information has increased due to the emergence of DL [17,18]. DL is an integrated storage for all company data in its original format, regardless of whether it is structured, semi-structured, or unstructured [19,20]. Each facet of the data, starting from managing the data ingestion channel to the transformation layer for analytical purposes, will be addressed within the data lake ecosystem [21,22]. Thus, this system can store vast amounts of data in different structures and formats. A data lake may encompass raw, unstructured, or structured data, much of which may hold undisclosed value for the organization [23].

DL is a methodology that leverages public data and low-cost technology to enable capturing, refining, archiving, and exploring raw data in the enterprise, and it uses a flat architecture and runs different types of data analytics [18,24,25]. This system is designed to store various forms of data, which may include unstructured or semi-structured data that have great value for the company. In essence, DL is a repository for all types of data in an enterprise, regardless of their type, format, or structure. DL is based on the 4VS data features: volume, velocity, variety, and veracity. Given the considerable amount of DL, a proficient dataset classification is essential to facilitate data analysis and information absorption [22,26–28]. The objective is to utilize a data metacharacteristics to characterize the dataset and identify resemblances. Nonetheless, data stored in the data lake might become inaccessible over time if the semantics are unavailable. Additionally, manual cleaning and consolidating of data can be challenging due to the data format and the volume of collection. The data lake relies on big data technologies like Hadoop, Spark, and Thread to establish a scalable, distributed, reliable, and fault-tolerant system. DL serves as a data storage unit that holds vast and varied datasets in their native form. The data undergo on-demand processing and modifications and are integrated with analytics, automation, advanced orchestrations, and machine intelligence tools and languages [29,30].

DL enables the consolidation of different types of data into one location. The data may come from a variety of data sources, can be logically relevant, and can be stored in structured, semi-structured, or non-structured raw format [31,32]. DL storage is categorized based on goals, such as: (1). Reservoir Data: Only datasets in the Hadoop File System (HDFS) are cleared and subjected to profile creation rules; (2). Exploratory Lake: A compilation of data applications ingested into HDFS with minimal cleaning, transformation, or merging from diverse data sources as required; (3). Data Analytics DL:

digests data in HDFS and feeds them to their analytic models for additional analysis, like predictive analysis [23]. Essentially, a data lake serves as a repository that preserves company data in its original state, encompassing structured data, unstructured, and semi-structured data, regardless of their type, format, or structure. Finally, DL understands the data properties that are delegated to data users during the data retrieval.

3.2 Smart City

The six domains that affect the understanding of the basic components of a smart city include a smart economy, living conditions, environment, populace, governance, and mobility [33,34]. The notion of a smart city is among the prominent themes of the fourth industrial revolution. This concept refers to cities that leverage information communication technology for their operation [35,36]. The need for smart cities becomes more pressing as the world's population rapidly increased [37]. To generate fresh concepts for forthcoming smart city technologies, this research examined the current literature on the topic [36]. The population in urban areas is projected to increase by 75% by 2050, resulting in a heightened demand for smart and sustainable environments that provide citizens with a high quality of life. This situation leads to the evolution of smart cities [38]. Smart city is a city innovation that aims to enhance the quality of life of its inhabitants by using technology to advance social and economic aspects.

Different definitions have been formulated for smart cities: a smart city leverages society and technology to create smart economics, smart mobility, smart environment, smart management administration, and overall smart life. The general idea of a smart city covers the following main components [39]: (1). Intelligent economy; (2). Intelligent environment; (3). Intelligent administration; (4). Intelligent communication; (5). Intelligent transportation [40–42]. The smart city framework includes management and organization, technology, management administrators, policy contexts, community and community, economics, infrastructure, and the natural environment [43,44]. The important technological factors in the use and maintenance of smart cities and technology are the driving forces that establish and maintain smart cities to deliver promised services. However, studying the safety of a smart city to manage administration and socioeconomic factors is important in identifying the concerns and safety needs of stakeholders [45–47]. Smart cities are conceived as a unified system where human and social interactions are facilitated by technology-driven solutions. The objective is to attain sustainable, resilient, efficient, and high-quality development through partnerships involving multiple stakeholders, including municipalities. Smart cities offer opportunities to connect people and places and use innovative technology to help plan and manage better cities. The core of smart cities lies in the process of collecting, managing, analyzing, and visualizing large amounts of data that are generated in the city environment every minute as a result of socioeconomic and other activities [48]. Smart city data can be collected directly from a variety of sensors, smartphones, and citizens and integrated with urban data repositories to implement analytical algorithms and produce the information needed to make decisions for better smart city management administration [49–52].

3.3 Relational Database Management System

The RDBMS facilitates the connection among tables within a database [53,54]. Each table has a key called the primary key to be connected to the next table that has a foreign key (e.g., MySQL, MS.SQL Server, and ORACLE). The system will prevent data redundancy using the primary key contained in a table, and it can be used to develop a complex database. RDBMS provides a normalization process. RDBMS is specifically designed to handle large-sized data and has a considerable number of users. Such a system applies security and functions, such as ACID, to increase

the integrity of a database [55,56]. RDBMS comprises a set of programs and functionalities enabling application developers to generate, modify, administer, and engage with relational databases [1]. In addition, RDBMS stores data in the form of tables, with the most commercial RDBMS using the SQL to access the database. Hence, this system offers a dependable approach to storing and fetching substantial amounts of data, combining system efficiency with straightforward implementation.

3.4 NoSQL Database

The NoSQL database can store and process significant amounts of data rapidly, without relying on a relational model [57–59]. This database does not adopt the relational data model [60]. NoSQL databases offer high flexibility and employ a dynamic schema to accommodate both structured and unstructured data [61]. The NoSQL database offers improved support for scalable architecture by utilizing open-source software, commodity servers, and cloud computing, rather than relying on large monolithic server and storage infrastructure used in relational databases [10,62]. The dataset was transferred into every NoSQL database (Redis, Mongodb, Redis, Cassandra, Neo4j). ACID properties (Atomicity, Consistency, Isolation, Durability) are important principles of transaction management in databases, traditionally implemented in relational databases. However, in the context of NoSQL databases, the implementation of ACID properties is often not done explicitly or not strictly adhered to. For example, some types of NoSQL databases, such as document-based or columnar databases, may not prioritize as strict data consistency as relational databases. Consequently, this can have significant implications for query classification, especially in the context of using machine learning techniques. For example, query classification that relies on strict data consistency assumptions may not always hold in a NoSQL environment, which can affect the accuracy and reliability of the classification model. Therefore, a deeper understanding of how ACID properties are applied or ignored in NoSQL databases is key to developing effective and reliable query classification approaches in these diverse environments. By considering the unique characteristics of different types of NoSQL databases and different approaches to ACID properties, we can develop query classification strategies that are more adaptive and responsive to the changing environment [63–65]. The characteristics and comprehension of each NoSQL database depend on its respective type, as outlined below [66–68]:

- Document-Base (Mongodb)
  Each data entry is maintained as a document, and each document within a dataset doesn't necessarily conform to the same structure as another (known as a table in SQL terminology). The type offers the advantage of not requiring a predetermined scheme.
- Key Value Store (Redis)
  Redis, short for Remote Dictionary Server, is a tool that furnishes in-memory data structures suitable for serving as both a database and cache for streaming machines.
- Graph store (Neo4j)
  This database is structured to depict relationships, resembling graphs consisting of nodes and edges. Such databases are frequently utilized for social media platforms, public transportation systems, mapping services, network topology, and similar applications. All relationships are stored in a single table.
- Wide column store (Cassandra)
  This database stores data using models similar to columns, employing the concept of keyspaces. A keyspace resembles a schema in SQL. It contains column families, which are akin to tables in SQL, comprising rows and columns.

The following is a comparison of the features between RDBMS (MySQL) and NoSQL:

According to Table 1, RDBMS has a structured and well-organized data structure and schema using tables and relationships between tables. Meanwhile, NoSQL has an unstructured and semi-structured data structure schema and does not have a fixed schema. Accordingly, RDBMS is more suitable for structured data and complex transactions, whereas NoSQL is more suitable for managing unstructured data, handling large amounts of data, and providing high horizontal scalability.

Table 1: Comparing the characteristics of RDBMS and NoSQL databases

No.	Feature	RDBMS	NoSQL			
		MySQL	Mongodb	Redis	Neo4j	Cassandra
1	Database category [66,69]	Relational DBMS [66,69]	Document store [66,69]	Key-value store [66,69]	Graph database [66,69]	Wide column store [66,69]
2	Database [60]	Database [60]	Database [60]	Database [60]	Graphs [60]	Keyspace [60]
3	Table [70]	Relation [70]	Collection [70]	Hash set, list set, sorted set and string [70]	Label [70]	Column family [70]
4	Value [66]	Rows [66]	Documents [66]	Key-value pair [66]	Node and edges [66]	Rows [66]
5	License [69,71]	Open source [69,71]	Open source [69,71]	Open source [69,71]	Open source [69,71]	Open source [69,71]
6	Language [69]	C and C++ [69]	C++ [69]	C [69]	Java, Scala [69]	Java [69]
7	Description [69,72]	Widely used open source RDBMS [69,72]	It is one of the well-known databases for storing documents [69,72]	It functions as an in-memory data structure repository and serves as a significant key-value store [69,72]	Open source graph database [69,72]	It is among the widely used databases with a wide column storage structure, inspired by the BigTable concept [69,72]
8	Schema [66]	Structured [66]	Semi-structured, Structured, and unstructured data [66]	Semi-structured, Structured, and unstructured data [66]	Semi-structured, Structured, and unstructured data [66]	Semi-structured, Structured, and unstructured data [66]

In transaction processing databases, complex SQL queries, RDBMS, and NoSQL are segregated into sections known as Online Transactional Processing (OLTP) and Online Analytical Processing (OLAP).

In Table 2, when viewed from its purpose, OLTP is used to handle simple business transactions with high volume and fast responsive requests, typically for day-to-day operations, such as buying, selling, and ordering. Meanwhile, OLAP is used to analyze business data, extract information, and gain deeper insights into the data. OLAP is used to view historical data, identify trends and patterns, and generate reports.

Table 2: Comparison of OLTP and OLAP in the NoSQL database

No.	Characteristics	Online Transactional Processing (OLTP)	Online Analytical Processing (OLAP)
1	Description [73,74]	Transaction processing [73,74]	Information and analytical processing [73,74]
2	Orientation [75,74]	Transactions [75,74]	Analysis [75,74]
3	Objective [76]	Controlling, executing and managing real-time business operations [76]	Analyse data to identify hidden patterns and derive insights [76]
4	Function [77]	Day to day operations [77]	Decision support and long-term informational requirements [77]
5	Data [78]	Up to date [78]	Consistency maintained over time [78]
	Access data [75]	Read/write [75]	A number of scans [75]
6	Database design [57]	Application-oriented [57]	Subject-oriented [57]
7	Size [79]	Gigabytes [79]	Terabytes [79]
8	Use [80]	Transactional data with evenly distributed usage, updated and written in real-time [80]	Reporting based on batch loading and read-only usage peaks, correlated with warehouse load times [80]
9	Data updates [81]	Short, fast, and regular updates [81]	Data periodically refreshed [81]

NoSQL databases have several differences from each of the existing NoSQL database categories in terms of functionality, complexity, flexibility, scalability, and performance. Table 3 illustrates a comparison between the NoSQL databases in terms of categories and descriptions.

Table 3: Comparison of the type of categories of the NoSQL database

No.	Categories of the NoSQL database	Functionality	Complexity	Flexibility	Scalability	Performance
1	Object store [82,83]	Object-oriented programming [82,83]	Low [82,83]	High [82,83]	Variable (high) [82,83]	High [82,83]
2	Column stores [82–85]	Minimum [82–85]	Low [82–85]	Moderate [82–85]	High [82–85]	High [82–85]

(Continued)

Table 3 (continued)

No.	Categories of the NoSQL database	Functionality	Complexity	Flexibility	Scalability	Performance
3	Graph store [82–85]	Graph theory [82–85]	High [82–85]	High [82–85]	Variable (high) [82–85]	Variable (high) [82–85]
4	Key-value stores [82–85]	Variable (none) [82–85]	None [82,84,83,85]	High [82–85]	High [82–85]	High [82–85]
5	Document stores [82–85]	Variable (low) [82–85]	Low [82–85]	High [82–85]	Variable (high) [82–85]	High [82–85]

Table 3 shows several comparisons based on the NoSQL categories owned by every NoSQL database, each with its advantages and disadvantages. When considering the basic concepts of RDBMS and NoSQL databases, a relationship exists between the two, which can complement each other in managing DL in smart city. RDBMS is used to process structured and table-based data, whilst NoSQL is utilized to process unstructured data and is based on documents, graphics, or key-value. In DL management, RDBMS can be used to store data that has been extracted, transformed, and loaded (ETL) and is ready for processing. Meanwhile, NoSQL databases can be used to store raw data that have not been ETL and allow for more flexible and scalable data processing.

Additionally, it is important to consider the challenges that arise in managing and analyzing complex SQL query data in diverse NoSQL environments. Different types of NoSQL databases have different characteristics, including flexible data schemas, eventual consistency, and geographically distributed distribution. This leads to the need for an approach that can adapt to the diverse nature of NoSQL databases. In this context, the scalability and adaptability of machine learning approaches are key to ensuring effectiveness and efficiency in data management and analysis. Machine learning techniques capable of operating on a large scale and adapting to variations in data structure and characteristics would be the ideal solution to overcome these challenges. Therefore, continued research in the development and implementation of machine learning approaches that can effectively adapt to diverse NoSQL environments will be an important step in improving the performance and relevance of modern database management systems. This research is in the broader context of developments in big data and machine learning technology, particularly in the increasingly popular application of NoSQL databases to handle large and varied data volumes. Previous studies have explored various techniques for managing and analyzing data in the Data Lake ecosystem, as well as highlighting the challenges of handling complex SQL queries in NoSQL databases that do not fully support ACID properties. Reference [20] shows the importance of automatic classification of SQL queries to improve data processing efficiency in the context of smart cities. However, there is still a lack of understanding of the optimal method for automatic classification using machine learning. Therefore, this research aims to fill this gap by systematically reviewing existing techniques and proposing new approaches that can improve the efficiency and accuracy of SQL query processing in NoSQL databases. This

contribution is critical to progress in the implementation of smart city technology and other data-intensive applications.

3.5 Methods Employed in Machine Learning Classification within NoSQL Databases

Machine learning pertains to the creation of learning algorithms within the realm of artificial intelligence (AI) capable of automatically improving their performance over time without explicit instruction from a user. This type of machine learning provides different frameworks for solving different problems and exploring patterns in data [86]. The selection of the right type depends on the characteristics of the data, the objectives to be achieved, and the resources displayed in Fig. 2.

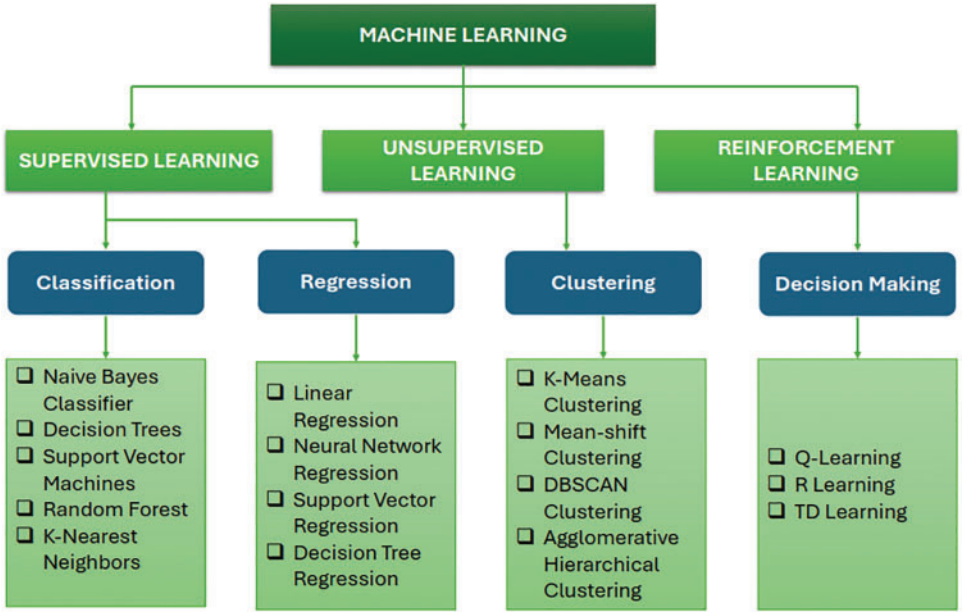

Figure 2: Types of machine learning. Reprinted from Reference [86]

Machine learning discusses the question of how to develop a computer that automatically increases through experience. In the domains of computer science, statistics, AI, and data science, this tool is experiencing rapid growth and adoption [86]. The machine learning algorithm is classified into the following types based on the output required for each algorithm:

- *Monitored Learning:* Learning under supervision, this type produces algorithms that correspond the input to the desired output. For example, the right decisions are made in the input to the algorithm model during the learning process. This type of learning is fast and accurate.
- *Uncovered Learning:* The form of learning that is more difficult as it involves a computer learning how to do something without being given explicit instructions. Clustering is the most common method of learning.
- *Enforcement Learning:* Strengthening Learning, each action affects the environment. The feedback received from the environment serves as a guide for the learning algorithms. Although machine learning is widely used and has great potential, its limitations must be understood. Currently, machine learning cannot replicate the full functionality of the human brain. Thus, one must exercise caution when applying machine learning algorithms in real-world settings and have a clear understanding of its capabilities before deployment.

Employing supervised machine learning with historical and current monitoring data, the existing middleware adjusts configurations to map application workloads onto distributed system configurations, adapting to changing workloads. The overview is depicted in Fig. 3.

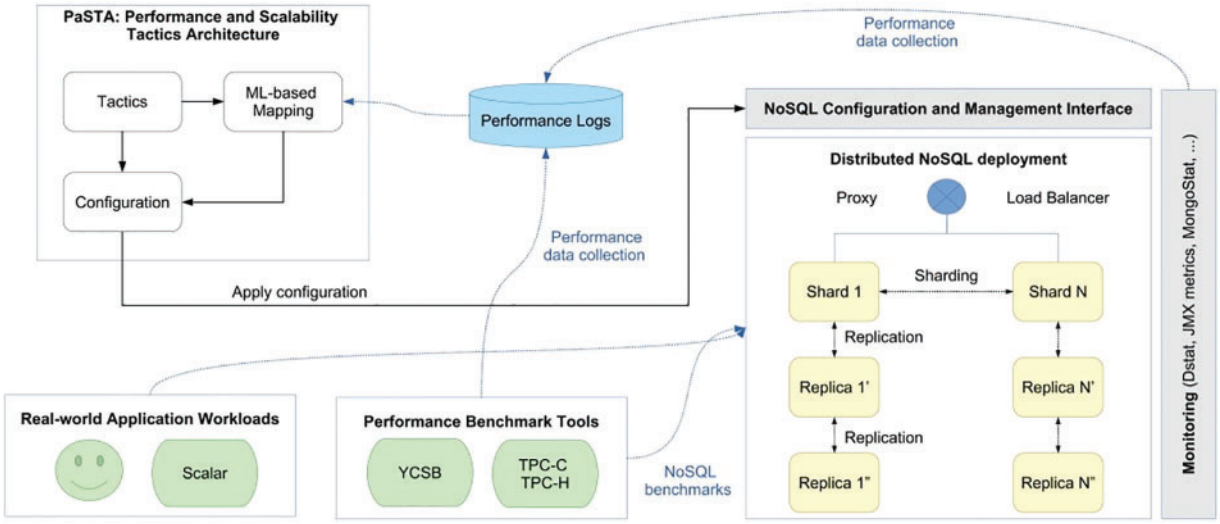

Figure 3: Performance and scalability on a NoSQL database system using machine learning. Reprinted from Reference [87]

Machine learning is applied to understand the performance patterns and scalability of NoSQL systems during runtime. Data on performance and scalability collected from active NoSQL systems serve as input for machine learning algorithms. The machine learning algorithm will analyze the data and identify patterns and correlations that exist between system configuration, performance parameters, and scalability. One of the main challenges is data heterogeneity, where NoSQL databases often store data in various formats and structures, ranging from structured data to unstructured data. This requires an approach that can adapt to the unique characteristics of each type of data. In addition, scalability issues are also a major concern, because NoSQL databases are often used to store data on a very large scale. Machine learning must be able to cope with increasing data scale without sacrificing performance or accuracy. Additionally, handling semi-structured and unstructured data is also an important challenge, as NoSQL databases can store data in unstructured formats such as JSON documents or graphics. In facing these challenges, machine learning approaches must be able to extract useful information from unstructured data and apply it in the classification process of complex SQL queries in NoSQL database environments. By exploring the performance, accuracy, and efficiency of different classification methods in a NoSQL setting, comparative studies can provide an in-depth understanding of the strengths and weaknesses of each approach. This can assist practitioners in selecting the method that best suits their analytical goals while improving the efficient use of computing resources.

Architectural classification is used to regulate the current related work based on system architecture followed by the approach. Fig. 4 shows the proposed classification. As previously mentioned, our classification grouping approaches are classified into two categories: layer and storage machines. In summary, all approaches are considered to map the relational scheme and SQL operations to the data and operations models of each NoSQL database [60]. The primary objective is to integrate

the strengths of both worlds: the robustness of the SQL standard and the extensive data handling capabilities of NoSQL databases.

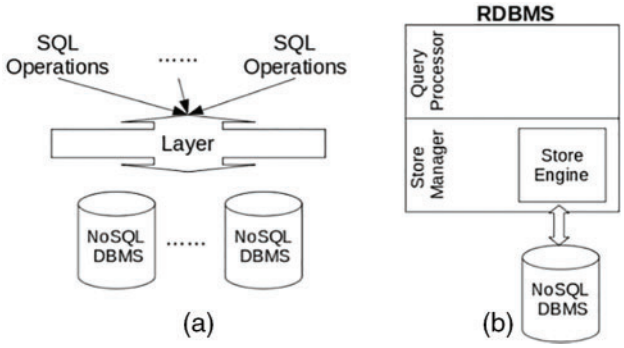

Figure 4: Summary of the architectural categorization for the related studies on simple and complex queries. Layer over one or more NoSQL databases (a). Layer of an RDBMS to provide the storage of relational data in a NoSQL database (b). Reprinted from Reference [60]

Based on Fig. 4 above, Fig. 4a depicts an architecture where an intermediate layer facilitates SQL operations on multiple NoSQL databases. Meanwhile, Fig. 4b shows an RDBMS architecture that integrates a NoSQL DBMS in its storage engine, enabling direct interaction between the query processor and the NoSQL database. Several mathematical formulas are used to provide accurate systematic mapping, especially when checking the availability of cluster or classification data for a simple and SQL query complex. This formula is presented as follows [88]:

$$C_{m_{c,n}} = \sum_{i=0}^{n} \frac{c!}{i! \times (c-i)!} \times C^{(c-i)} \times (1-A)\, i. \tag{1}$$

If the system requirement is C consistency, and the level is M medium. For example, if a system has a number of C cluster SQL queries, then it is considered available under this formula when at least the C–N component is available, which means no more than n clusters can fail. Several studies that explore various simple and complex SQL queries and methods for managing NoSQL databases show how NoSQL databases retrieve data in different ways to address the trigger function problems and ACID functions with a machine-learning approach. Some researchers also inquire about questions or ways to facilitate queries of existing approaches, including the classification and clustering methods. A summary of previous studies is illustrated in Table 4.

Table 4: Synopsis of machine learning techniques and their applications within the NoSQL

No.	Applications	Support the ACID				NoSQL database type	Methods and references
		Atomicity	Consistency	Isolation	Durability		
1	Smart grid system and big data frameworks [22,89,90–92]	√ [22,89–92]	X [22,89–92]	X [22,89–92]	√ [22,89–92]	Key-value store and wide column store [22,89–92]	Machine learning and clustering [22,89–92]

(Continued)

Table 4 (continued)

No.	Applications	Support the ACID				NoSQL database type	Methods and references
		Atomicity	Consistency	Isolation	Durability		
2	SPARQL converting complex queries [89,93–95]	√ [89,93–95]	X [89,93–95]	√ [89,93–95]	√ [89,93–95]	Document base [89,93–95]	Ontology-based access data (OBDA) Querying [89,93–95]
3	NewSQL databases [60,96,97]	√ [60,96,97]	X [60,96,97]	√ [60,96,97]	√ [60,96,97]	Document base, key-value store and graph store [60,96,97]	Machine learning and classification [60,96,97]
4	NoSQL transformation [81,95,98,99]	√ [81,98,99]	X [81,98,99]	√ [81,95,98,99]	√ [81,95,98,99]	Document base, key-value and wide column store [81,95,98,99]	Query mapping approaches For synchronisation [81,95,98,99]
5	Transformation of nested queries [75,100]	√ [75,100]	X [75,100]	√ [75,100]	√ [75,100]	Document base and key-value store [75,100]	Nest-G algorithm [75,100]

Graph database technology is used to manage data and metadata in the concept of DL architecture of the semantic DL catalog knowledge using graphs. A graph database is a database format that arranges data as graphs or networks, comprising nodes and edges. This database allows users to access data in a more flexible and integrated manner and perform a more complex data analysis [101]. The DL architecture of the semantic DL catalog knowledge using a graph concept can create a highly flexible, accessible, and integrated data environment by combining the concepts of DL, semantic DL, data catalog, and graph database technology, making it easier for users to quickly and efficiently find and analyze data. The DL architecture of semantic DL catalog knowledge utilizing graph concepts creates a highly adaptable and interconnected data environment. Integrating DL concepts with semantic DL, data cataloging, and graph database technology, facilitates quick and efficient data discovery and analysis for users.

By leveraging these components together, the DL architecture incorporating graph database technology enhances data accessibility, flexibility, and integration. Users can navigate through the interconnected data landscape more intuitively and perform complex analyses with ease. Overall, the integration of graph database technology enhances the capabilities of the semantic DL catalog knowledge concept, making it a powerful tool for managing and analyzing data in modern data environments.

Meanwhile, the catalog semantic DL is a key component of the DL architecture because it handles and controls the collection and access to information that enables the search and classification of semantic data, increasing discoverability [102]. Data can be found faster, better, and automatically by

machines, enabling further data analytics use cases. Data usage is much better overall, as shown in Fig. 5.

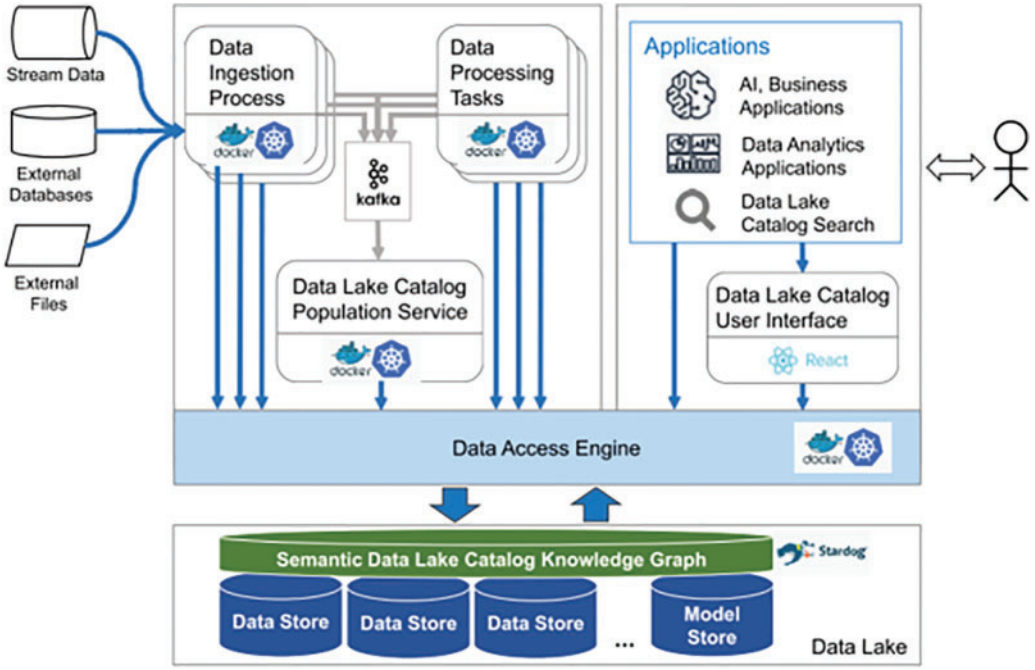

Figure 5: DL architecture of semantic DL catalog knowledge using a graph. Reprinted from Reference [103]

DL architecture of semantic DL catalog knowledge using a graph is a data structure that integrates the principles of Data Lake (DL), Semantic Data Lake (SDL), and data catalog to create a highly flexible, accessible, and integrated data environment [103]. This architecture also uses graph database technology to manage data and metadata. A DL is a concept for collecting raw data from various sources and storing it in one place without the need to perform data transformation or processing first [103]. This concept enables organizations to access and quickly and efficiently analyze data. The semantic DL is a concept that enriches raw data with additional information, such as metadata, ontology, and semantics. This system makes the data stored in the DL more understandable and accessible to users. Meanwhile, a data catalog is a system used to manage metadata from a data environment.

Metadata can be information about the data origin, data structure, or other pieces of information related to the data. Data catalogs allow users to quickly and easily find the data that they need. Graph database technology is used to manage data and metadata in the DL architecture of semantic DL catalog knowledge using a graph concept [104]. A graph database is a type of database that organizes data in the form of graphs or networks that consist of nodes and edges. Fig. 6 demonstrates the proposed Complex Querying for Relational and NoSQL Databases (CQNS) approach enables the execution of complex queries across heterogeneous data stores. This framework consists of three layers, namely, the suitable layer of voters, processing layers and queue execution layers.

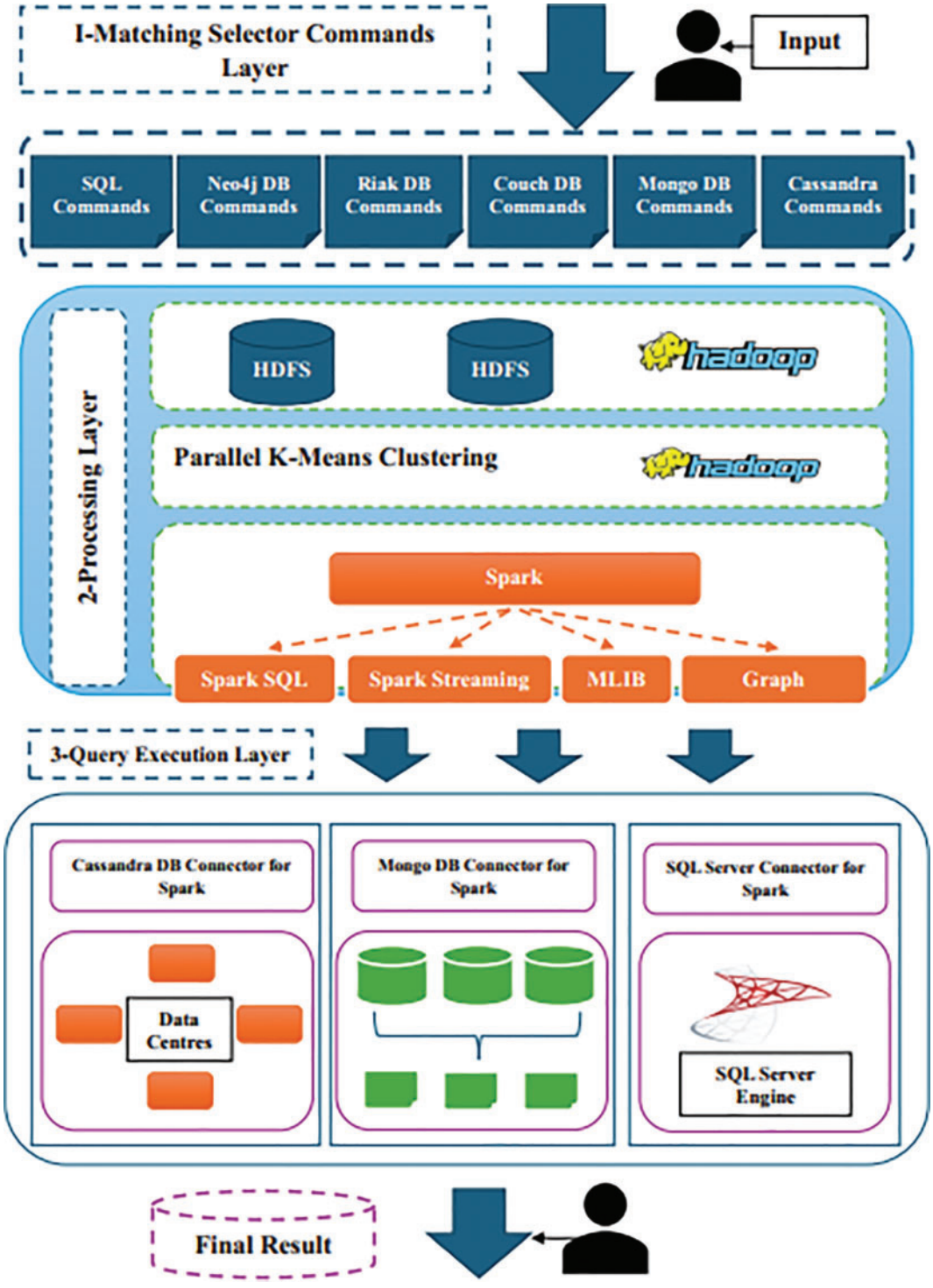

Figure 6: Framework of complex query for relational and NoSQL databases. Reprinted from Reference [104]

The layer illustrated in Fig. 6 processes an SQL or NoSQL database request, matching it with user queries against a stored library containing various statements for each type of database (SQL or NoSQL) from the database machine. The system then compares the sentence with the stored library to determine the database machine required to operate. In Fig. 7, the SQL library statement contains a set of CRUD statements for each document stored in the SQL database, such as Mongodb and Cassandra, as a test of the NoSQL database library [95]. This framework model contains certain database operating functions, and users must add particular data storage implementations if they need to integrate other additional NoSQL databases.

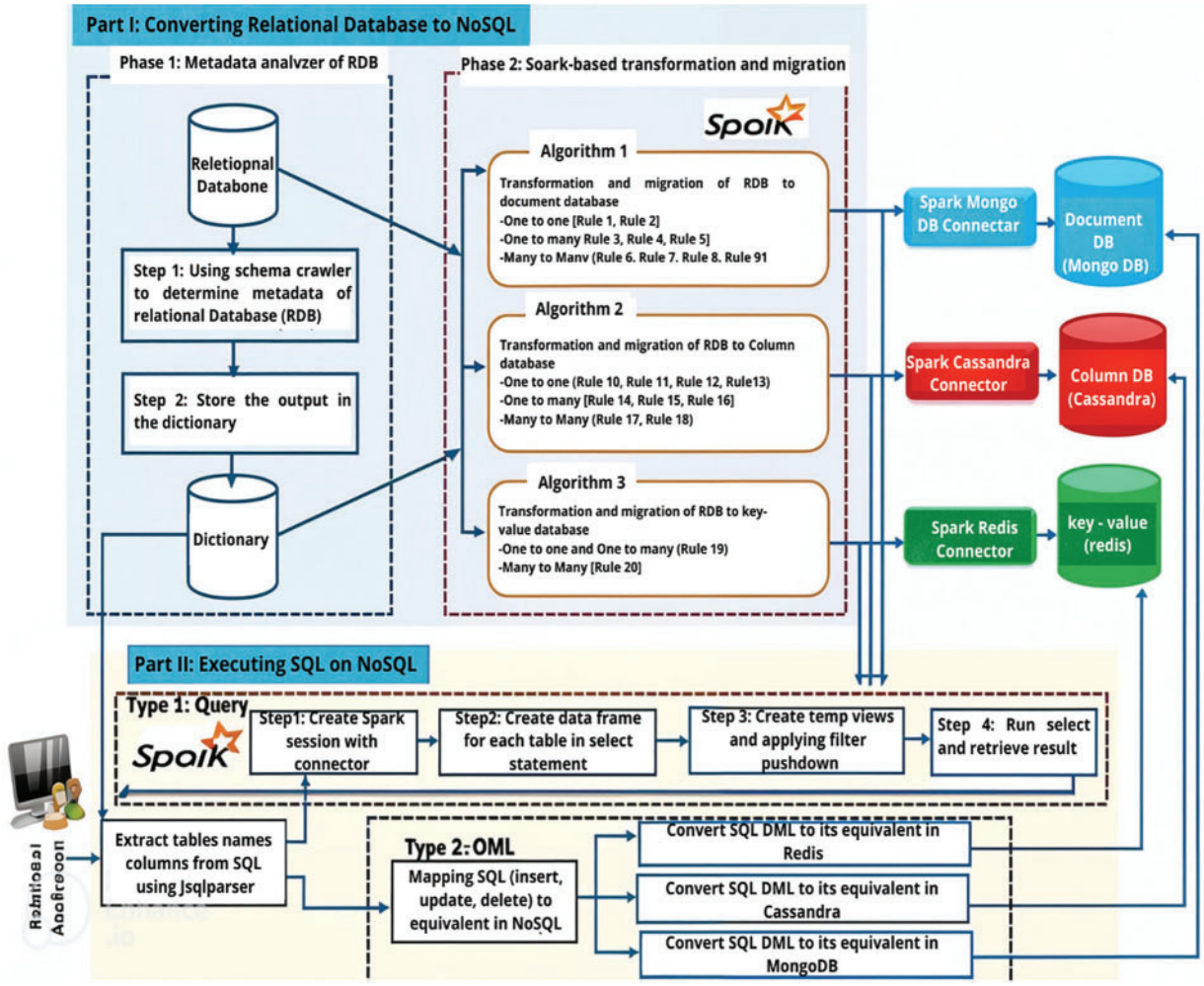

Figure 7: Architecture of the Spark-based layer. Reprinted from Reference [95]

The next research is to transform SQL to NoSQL using the query mapping approaches for the synchronization method (Fig. 7). Database conversion from relational to NoSQL can be more quickly and efficiently carried out using the query mapping clustering machine learning method and approach for relational database conversion to NoSQL using a Spark-based layer architecture because Apache Spark can process data in a distributed and efficient manner. Query mapping belongs to NoSQL databases (document, column, and key-value databases).

The subsequent research, called multi-source integration, shows a model for overcoming the trigger function problem and how SQL queries can be implemented into several databases (RDBMS and NoSQL), namely, the transformation of nested queries approach with the RDBMS database integration model and NoSQL data storage (Fig. 8).

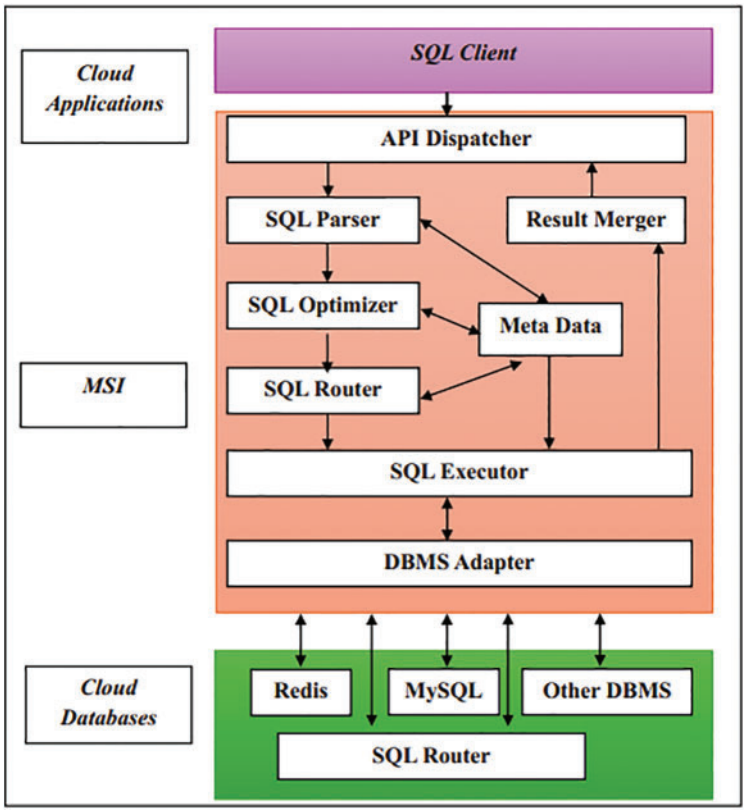

Figure 8: Architecture transformation of nested queries. Reprinted with permission from Reference [75]. Copyright © 2018 Wiley

The DBMS adapter component primarily comprises semantic checking, parameter generation, API calls, run-time supporter, and error handling, with the concept of transformation of nested queries. The inputs of the DBMS adapter component are the language objects generated by the SQL parser component, which has probably been optimized by the SQL optimizer component, and its outputs are the calls of the encapsulated APIs for the selected NoSQL DBMS. The semantic checking module is responsible for checking the language objects produced by the upper layer. The parameter generation module extracts the relevant parameters from each SQL statement and converts them into specific parameter objects, which are then utilized in the encapsulated APIs for the NoSQL database [75].

The problem of trigger and ACID functions in NoSQL databases using the conversion of nested query models can be addressed by creating one main query that contains all the required subqueries; by doing so, the nested queries are converted into a single main query [100]. Locking techniques can be used to ensure the data integrity of transactions performed on NoSQL databases. Meanwhile, compensating techniques are used to correct errors that occur in transactions in case an error emerges during a transaction. Furthermore, distributed technology ensures accurate data replication and

overcome data availability problems in NoSQL databases. Thus, through a careful synthesis of multiple perspectives in the relevant literature, this research can be effectively placed in a broader context, strengthen its theoretical foundation, and make a meaningful contribution to our understanding of the implementation of trigger functions and classification of complex queries using machine learning in NoSQL databases for Data Lake in Smart Cities. By expanding the analysis of the limitations of this study, including an evaluation of the potential generalizability of findings, implementation barriers, and suggestions for further research, this study can make a more significant contribution to filling existing knowledge gaps and provide clearer guidance for practitioners and researchers in the field SQL query classification in the context of NoSQL databases for Data Lake environments in smart cities.

4 Discussion

4.1 Support Vector Machine

Following are some of SVM's main contributions to this classification: (1). Query Classification: SVM can be used to classify queries in NoSQL databases into two or more relevant categories. (2). Improved Query Performance: SVM can help improve query performance in NoSQL databases by predicting the type and complexity of queries before executing them. (3). Pattern Recognition: SVM can recognize patterns and complex relationships in past queries and use them to perform never-before encountered query classifications. (4). Handling a Variety of Data Formats: NoSQL databases can store data in diverse formats, such as structured, semi-structured, and unstructured. SVM can overcome these variations of data formats by studying the patterns and characteristics of each data format [105]. (5). In evaluating the performance of SVM in classifying queries in NoSQL databases, it is important to consider that the results obtained can be influenced by several factors, including dataset characteristics, feature selection, and the complexity of the query being handled. A deeper analysis of these factors could provide more comprehensive insight into the reliability and effectiveness of SVMs in this context [106]. (6). The practical implications of this research are significant, given the continued increase in the use of NoSQL databases in various applications. With an improved comprehension of the capabilities and constraints of SVM in classifying queries, practitioners and researchers can develop more effective strategies for managing and analyzing data in NoSQL database environments.

4.2 Naïve Bayes Classification

Following are some of NBC's contributions to this classification: (1). Query Classification: NBC can be used to classify queries in NoSQL databases into relevant categories. By utilizing the probability method and naive assumptions that each feature (query characteristic) is mutually independent, NBC can provide an estimate of the optimal class probability based on the observed features. (2). Diverse Data Handling Capabilities: NBC can handle diverse data in NoSQL databases. NBC can be used to classify queries with different data types, whether they are structured, semi-structured, or unstructured. (3). Scalability: NBC has good scalability and is efficient in classifying queries in NoSQL databases. (4). Ease of Implementation and Interpretation: NBC is a machine learning method that is relatively simple and easy to implement [107]. (5). In facing the challenges of implementing trigger functions as part of ACID functions in NoSQL databases, the need for innovative and adaptive approaches becomes increasingly important. The integration of machine learning methods such as SVM, and NBC can be a promising step to overcome this obstacle, by providing a more effective and efficient solution in transaction management and query classification in NoSQL databases [108,109].

This study shows that DL storage management uses classification and grouping methods with machine learning, such as SVM, NBC, and K-means, to address complex SQL statement transaction problems in NoSQL databases. However, the implementation of the trigger function as part of the ACID function may still face challenges in achieving optimal performance. The reason is that the ACID and trigger functions are still fully supported and run on a relational database (RDBMS) with a structured data type. Meanwhile, NoSQL databases can store all data formats and types, whether structured, semi-structured, or unstructured. Consequently, special methods and adapters may be required to address these issues. Machine learning techniques, such as SVM, K-means, and NBC, have the potential to offer solutions and significantly contribute to resolving these problems in NoSQL databases, especially the ACID and trigger functions. NoSQL can be optimally applied for managing smart city DL. This study can also assist in DL in Smart City Data Management by providing effective and complex query support using classification machine learning methods.

Although machine learning has been widely used in the context of data analysis, its use specifically for SQL query classification in NoSQL databases has unique challenges and requirements. NoSQL databases have a different structure and often do not support features such as ACID properties that are common in relational databases. Therefore, developing and applying machine learning techniques for SQL query classification in NoSQL databases requires an approach tailored to the characteristics and needs of that environment. Classification of SQL queries in NoSQL databases poses additional challenges, such as managing semi-structured and unstructured data, as well as processing complex queries with features such as triggers and ACID functions. This requires the development of classification models that to overcome this complexity and provide accurate and reliable results. The application of machine learning for SQL query classification in NoSQL databases aims to improve the efficiency and accuracy of data processing in this environment. By automating the classification process, users can save time and resources in data management and analysis, and enable more effective use of NoSQL databases. Therefore, although machine learning concepts have been widely applied in data analysis, research specifically focusing on its use for the classification of complex SQL queries in NoSQL databases is still a developing field and has the potential to significantly contribute to information technology and data management. Therefore, this research can provide a new and valuable contribution to understanding and development in this domain.

5 Issues and Challenges

The problems and challenges mentioned relate to the distinctions between relational databases (RDBMS) and NoSQL databases in terms of ACID (Atomicity, Consistency, Isolation, Durability) and trigger functionality, as well as data structure flexibility. The following is a summary of the problems and challenges mentioned: ACID and trigger functions, Data format and type flexibility, and Special methods and adapters. Overall, the challenges lie in adapting the ACID and trigger functions, which are typically associated with structured data in RDBMS, to the flexible and diverse data formats supported by NoSQL databases. Finding suitable methods and adapters is crucial to overcoming these challenges and ensuring efficient and effective data management in NoSQL environments. Essentially, the paradigm differences between RDBMS and NoSQL databases create significant obstacles in adapting ACID and trigger functions, which are commonly associated with data structured in RDBMS, to NoSQL environments that support more flexible and diverse data formats.

These changes prompt fundamental questions about how NoSQL databases can maintain data integrity, transaction consistency, and isolation in the context of schema-independent data structures.

Some issues and challenges for future research include; (1). NoSQL Limitations in Supporting ACID Properties: NoSQL databases often cannot fully support ACID properties which are the main feature in managing data transactions in RDBMS [10,67]. This has led to research into solutions that allow NoSQL databases to support these features without sacrificing scalability and performance. (2). Application of Machine Learning for Query Classification: Studies indicate that machine learning holds significant promise for enhancing the efficiency of categorizing intricate SQL queries within NoSQL database varieties [8,9]. However, the challenge here is in identifying and applying the most effective machine learning algorithms for this complex SQL classification task. (3). Complex Query Classification: The main challenge is in classifying complex queries, such as subqueries, joins, triggers, and aggregation operations, which are often difficult to automate [7]. This requires the development of more sophisticated classification techniques and a deeper understanding of query and data structures in NoSQL databases [83]. (4). A new model of complex transaction algorithm with a trigger function in Atomicity, Consistency, Isolation, and Durability (ACID) is needed for managing various transactions in the Data Lake which can be used for databases in smart bookies using a machine learning approach [4,110]. In this research, there are also several limitations including; (1). Limitations on systematic reviews, and the limited number of studies that met the inclusion criteria resulted in restrictions on the variety of classification techniques in this review. (2). Limiting the scope of the search, studies conducted relating to less common NoSQL databases may not be included in the analysis. (3). Challenges in classifying complex SQL queries, and limitations in understanding the structure and behavior of complex queries can hinder the ability of classification models to differentiate appropriately between different types of queries. The unique challenges faced in the integration of machine learning with NoSQL databases give rise to new, innovative methodologies to increase the effectiveness and reliability of data analysis processes. One of the main challenges is the complexity of the diverse data structures in NoSQL databases, which include structured, semi-structured, and unstructured data. Solutions to these challenges may involve developing classification algorithms that can handle the flexibility of data structures and take into account the specific context of each type of data. Additionally, scalability is an important challenge in managing large and dynamically increasing data volumes. New approaches such as the use of cloud-based technologies or distributed computing can help overcome these problems by improving system performance and scalability. The issue of data consistency is also an important focus because NoSQL databases often offer eventual consistency rather than strong consistency as in relational databases. New methodologies that combine machine learning techniques with fast and efficient data recovery techniques can help in ensuring the desired data consistency.

This study has several limitations that need to be noted. First, limitations to this systematic review are due to the limited number of studies that met the inclusion criteria, which resulted in a limited variety of classification techniques that could be reviewed. Second, restrictions on the scope of the search mean that studies conducted related to less common NoSQL databases may not be included in this analysis. Third, challenges in classifying complex SQL queries as well as limitations in understanding the structure and behavior of complex queries can hinder the ability of classification models to differentiate appropriately between different types of queries. The unique challenges faced in integrating machine learning with NoSQL databases provide opportunities for the development of innovative new methodologies to increase the effectiveness and reliability of data analysis processes. One of the main challenges is the complexity of the diverse data structures in NoSQL databases, which include structured, semi-structured and unstructured data. Solutions to these challenges may involve developing classification algorithms that can handle the flexibility of data structures and take into account the specific context of each type of data. In addition, scalability is an important challenge in

managing large and dynamically increasing data volumes. New approaches such as the use of cloud-based technologies or distributed computing can help overcome these problems by improving system performance and scalability. Data consistency issues are also an important focus because NoSQL databases often offer eventual consistency rather than strong consistency as in relational databases. New methodologies that combine machine learning techniques with fast and efficient data recovery techniques can help ensure desired data consistency.

6 Conclusion

1. As a conclusion, there are several suggestions from the author that can be considered for future research as follows: Systematic Review: This systematic review has identified that the use of automatic classification techniques, especially those based on machine learning, has become the dominant approach in processing SQL queries in NoSQL databases.
2. Research and Development: Further research and development efforts should focus on exploring innovative approaches and techniques for implementing ACID and triggering functionality in NoSQL databases.
3. Standardization: Establishing industry-wide standards and guidelines for implementing ACID and triggering functionality in NoSQL databases can provide a framework for consistency and interoperability.
4. Developing the Smartdb Adapter for translating SQL queries to NoSQL databases can become a useful tool for bridging the gap between the structured nature of SQL and the flexible data models of NoSQL databases.
5. Smart Cities experience exponential growth in data volumes and user demands. Therefore, NoSQL databases must be optimized for scalability to accommodate increasing workloads and evolving requirements. Techniques such as sharding, replication, and horizontal scaling should be explored to ensure seamless scalability and high availability of data storage and processing resources.
6. Monitoring the performance of NoSQL databases in real time is critical for identifying bottlenecks, optimizing resource utilization, and ensuring optimal system efficiency. Implementing robust monitoring tools and performance analytics frameworks enables proactive problem detection and resolution, thereby enhancing the overall reliability and performance of smart city data management systems.

For future research, several directions can be explored to overcome existing limitations and improve understanding of processing complex SQL queries in NoSQL databases:

1. Future research could include more types of NoSQL databases and variations of complex SQL queries to gain a more comprehensive understanding of effective automatic classification techniques.
2. More research is needed to explore methods to improve the quality and representativeness of training data in the context of SQL query classification in NoSQL databases, including the use of advanced data processing techniques.
3. Future research could focus on developing more sophisticated and adaptive classification models to overcome challenges in complex query classification, including the integration of features such as triggers and ACID functions.
4. Further research is needed to test and evaluate the practical implementation of automatic classification techniques in real operational environments, as well as identify possible obstacles and challenges.

In addition, testing and evaluating practical implementation of automated classification techniques in real operational environments, as well as identifying barriers and challenges that may arise, are also important steps to direct future research in this area. Thus, future research is expected to make a significant contribution in developing SQL query classification techniques in NoSQL databases to support increasingly complex requirements in the smart city management context. Looking forward, several avenues for future research can build upon the findings of this study. One promising direction involves expanding the scope of research to include a broader variety of NoSQL databases and complex SQL query types. This could provide a more comprehensive understanding of the effectiveness of automatic classification techniques across different NoSQL environments. Additionally, future studies could focus on improving the quality and representativeness of training data used in SQL query classification by employing advanced data processing techniques. Another important area for exploration is the development of more sophisticated and adaptive classification models that can better handle the challenges associated with complex query classification, including those that involve triggers and ACID functionalities. Practical implementation and testing of these automatic classification techniques in real-world operational environments will also be crucial for identifying potential obstacles and refining these methodologies. By addressing these areas, future research can significantly enhance the efficiency and accuracy of SQL query processing in NoSQL databases, contributing to more robust and scalable data management solutions.

Acknowledgement: We would like to express gratitude to the Universiti Kebangsaan Malaysia (UKM) for providing the opportunity for this research.

Funding Statement: This research article was supported by the Student Scheme provided by Universiti Kebangsaan Malaysia with the Code TAP-20558.

Author Contributions: Nurhadi: Involved in the implementation of the SQL simple and complex in RDBMS and NoSQL Databases model development and experiment, analysis, and writing the draft of the manuscript. Rabiah Abdul Kadir: Responsible for testing the of complex SQL in RDBMS and NoSQL Databases Query model, examining the implementation, reviewing, and writing the manuscript. Ely Salwana Mat Surin: Review the manuscript. Mahidur R. Sarker: Review the manuscript. All authors reviewed the results and approved the final version of the manuscript.

Availability of Date and Materials: Data will be made available on request.

Ethics Approval: Not applicable.

Conflicts of Interest: The authors declare that they have no conflicts of interest to report regarding the present study.

References

[1] J. Endres, R. Bernsteiner, and C. Ploder, "A framework for the evaluation of NoSQL databases for big data use cases," in *Handbook of Research on Engineering Innovations and Technology Management in Organizations*, vol. 20, no. 4, pp. 66–90, 2020.

[2] A. Maté, J. Peral, J. Trujillo, C. Blanco, D. García-Saiz and E. Fernández-Medina, "Improving security in NoSQL document databases through model-driven modernization," *Knowl. Inf. Syst.*, vol. 63, no. 8, pp. 2209–2230, 2021. doi: 10.1007/s10115-021-01589-x.

[3] M. Hemmatpour, B. Montrucchio, M. Rebaudengo, and M. Sadoghi, "Analyzing in-memory NoSQL landscape," *IEEE Trans. Knowl. Data Eng.*, vol. 34, no. 4, pp. 1628–1643, 2022. doi: 10.1109/TKDE.2020.3002908.

[4] C. Zhang and J. Xu, "A unified SQL middleware for NoSQL databases," in *ACM Int. Conf. Proc. Ser.*, New York, NY, USA, 2018, pp. 14–19.
[5] E. Baralis, A. D. Valle, P. Garza, C. Rossi, and F. Scullino, "SQL versus NoSQL databases for geospatial applications," in *Proc. 2017 IEEE Int. Conf. Big Data, Big Data 2017*, Boston, MA, USA, 2017, vol. 2018, pp. 3388–3397.
[6] R. Byali, M. Jyothi, and M. C. Shekadar, "Evaluation of NoSQL database MongoDB with respect to JSON format data representation," *Int. J. Res. Publ. Rev.*, vol. 3, no. 9, pp. 867–871, 2022. doi: 10.55248/gengpi.2022.3.9.24.
[7] S. Rizzi, *OLAP and NoSQL: Happily Ever After*. in Lecture Notes in Computer Science. vol. 13389. Turin, Italy, Springer International Publishing, 2022.
[8] A. Krechowicz, S. Deniziak, and G. Lukawski, "Highly scalable distributed architecture for NoSQL datastore supporting strong consistency," *IEEE Access*, vol. 9, pp. 69027–69043, 2021. doi: 10.1109/ACCESS.2021.3077680.
[9] A. E. Lotfy, A. I. Saleh, H. A. El-Ghareeb, and H. A. Ali, "A middle layer solution to support ACID properties for NoSQL databases," *J. King Saud Univ. Comput. Inf. Sci.*, vol. 28, no. 1, pp. 133–145, 2016. doi: 10.1016/j.jksuci.2015.05.003.
[10] E. M. Kuszera, L. M. Peres, and M. Didonet Del Fabro, "Exploring data structure alternatives in the RDB to NoSQL document store conversion process," *Inf. Syst.*, vol. 105, 2022, Art. no. 101941. doi: 10.1016/j.is.2021.101941.
[11] Q. HamaMurad and N. M. Jusoh, "A literature review of smart city: Concept and framework," *J. Adv. Geospatial.*, vol. 1, no. 1, pp. 92–111, 2022.
[12] A. Arowosegbe and T. Oyelade, "Application of natural language processing (NLP) in detecting and preventing suicide ideation: A systematic review," *Int. J. Environ. Res. Public Health*, vol. 20, no. 2, 2023, Art. no. 1514. doi: 10.3390/ijerph20021514.
[13] M. Burlacu, R. G. Boboc, and E. V. Butilă, "Smart cities and transportation: Reviewing the scientific character of the theories," *Sustainability*, vol. 14, no. 13, 2022, Art. no. 8109. doi: 10.3390/su14138109.
[14] I. Suriarachchi and B. Plale, "Crossing analytics systems: A case for integrated provenance in data lakes," in *Proc. 2016 IEEE 12th Int. Conf. e-Sci.*, MD, USA, 2017, pp. 349–354.
[15] R. Hai, C. Quix, and C. Zhou, Query rewriting for heterogeneous data lakes. in *Lecture Notes in Computer Science*, vol. 11019. Budapest, Hungary, Springer International Publishing, 2018.
[16] F. Ahmad, A. Sarkar, and N. C. Debnath, "QoS lake: Challenges, design and technologies," in *Proc.-2017 Int. Conf. Recent Adv. Signal Process. Telecommun. Comput. (SigTelCom 2016)*, USA, 2017, pp. 65–70.
[17] A. Alserafi, T. Calders, A. Abelló, and O. Romero, DS-prox: Dataset proximity mining for governing the data lake. in *Lecture Notes in Computer Science*. vol. 10609. Munich, Germany, pp. 284–299, 2017.
[18] H. Dibowski and S. Schmid, "Using knowledge graphs to manage a data lake," in *Lecture Notes in Informatics*, Berlin, Germany, 2020, vol. P-307, pp. 41–50.
[19] M. Cherradi and A. El Haddadi, "Grover's algorithm for data lake optimization queries," *Int. J. Adv. Comput. Sci. Appl.*, vol. 13, no. 8, pp. 568–576, 2022. doi: 10.14569/issn.2156-5570.
[20] O. Azeroual, J. Schöpfel, D. Ivanovic, and A. Nikiforova, "Combining data lake and data wrangling for ensuring data quality in CRIS," *Proc. Comput. Sci.*, vol. 211, pp. 3–16, 2022. doi: 10.1016/j.procs.2022.10.171.
[21] M. N. Mami, D. Graux, S. Scerri, H. Jabeen, S. Auer and J. Lehmann, "Uniform access to multiform data lakes using semantic technologies," in *ACM Int. Conf. Proc. Ser.*, 2019. doi: 10.1145/3366030.
[22] A. A. Munshi and Y. A. R. I. Mohamed, "Data lake lambda architecture for smart grids big data analytics," *IEEE Access*, vol. 6, pp. 40463–40471, 2018. doi: 10.1109/ACCESS.2018.2858256.
[23] V. Belov, A. N. Kosenkov, and E. Nikulchev, "Experimental characteristics study of data storage formats for data marts development within data lakes," *Appl. Sci.*, vol. 11, no. 18, 2021, Art. no. 8651. doi: 10.3390/app11188651.

[24] C. Giebler, C. Gröger, E. Hoos, R. Eichler, H. Schwarz and B. Mitschang, "The data lake architecture framework: A foundation for building a comprehensive data lake architecture," in *Lecture Notes in Informatics*, 2021, pp. 351–370.
[25] M. Cherradi and A. El Haddadi, "DLDB-service: An extensible data lake system," in *Lecture Notes on Data Engineering and Communications Technologies*, vol. 147, pp. 211–220, 2023. doi: 10.1007/978-3-031-15191-0.
[26] M. Klettke, H. Awolin, U. Storl, D. Muller, and S. Scherzinger, "Uncovering the evolution history of data lakes," in *Proc.-2017 IEEE Int. Conf. Big Data, Big Data 2017*, Boston, MA, USA, 2017, pp. 2462–2471.
[27] H. Che and Y. Duan, "On the logical design of a prototypical data lake system for biological resources," *Front. Bioeng. Biotechnol.*, vol. 8, pp. 1–15, 2020. doi: 10.3389/fbioe.2020.553904.
[28] A. Guyot, A. Gillet, E. Leclercq, and N. Cullot, *A Formal Framework for Data Lakes Based on Category Theory*. Budapest, Hungary, Association for Computing Machinery, 2022, vol. 1, no. 1.
[29] J. Kachaoui and A. Belangour, "Enhanced data lake clustering design based on K-means algorithm," *Int. J. Adv. Comput. Sci. Appl.*, vol. 11, no. 4, pp. 547–554, 2020. doi: 10.14569/issn.2156-5570.
[30] A. A. Sukhobokov, Y. E. Gapanyuk, A. S. Zenger, and A. K. Tsvetkova, "The concept of an intelligent data lake management system: Machine consciousness and a universal data model," *Proc. Comput. Sci.*, vol. 213, pp. 407–414, 2022. doi: 10.1016/j.procs.2022.11.085.
[31] I. D. Nogueira, M. Romdhane, and J. Darmont, "Modeling data lake metadata with a data vault," *ACM Int. Conf. Proc. Ser.*, pp. 253–261, 2018. doi: 10.1145/3216122.
[32] J. Colleoni Couto, O. Teixeira Borges, and D. Dubugras Ruiz, "Data integration in a Hadoop-based data lake: A bioinformatics case," *Int. J. Data Min. Knowl. Manag. Process.*, vol. 12, no. 4, pp. 1–24, 2022. doi: 10.5121/ijdkp.2022.12401.
[33] S. B. Lim, J. A. Malek, M. F. Y. M. Yussoff, and T. Yigitcanlar, "Understanding and acceptance of smart city policies: Practitioners' perspectives on the Malaysian smart city framework," *Sustainability*, vol. 13, no. 17, 2021, Art. no. 9559. doi: 10.3390/su13179559.
[34] T. Aditya, S. Ningrum, H. Nurasa, and I. Irawati, "Community needs for the digital divide on the smart city policy," *Heliyon*, vol. 9, no. 8. 2023, Art no. e18932. doi: 10.1016/j.heliyon.2023.e18932.
[35] M. M. Rathore, A. Paul, A. Ahmad, and G. Jeon, "IoT-based big data: From smart city towards next generation super city planning," *Int. J. Semant. Web Inf. Syst.*, vol. 13, no. 1, pp. 28–47, 2017. doi: 10.4018/IJSWIS.
[36] V. Diaconita, A. R. Bologa, and R. Bologa, "Hadoop oriented smart cities architecture," *Sensors*, vol. 18, no. 4, 2018, Art. no. 1181. doi: 10.3390/s18041181.
[37] S. Korea, "Behavioral model deployment for the transportation projects within a smart city ecosystem: Cases of Germany and South Korea," *Process. Artic.*, vol. 11, 2023, Art. no. 48.
[38] C. S. Lai, L. L. Lai, and Q. H. Lai, *Smart Grids and Big Data Analytics for Smart Cities*. Cham, Switzerland, Springer Nature Switzerland AG, 2021.
[39] D. D. Nguyen and A. Boros, "Applied sciences for future smart cities," *Appl. Sci. Artic.*, vol. 10, 2020, Art. no. 8933. doi: 10.3390/app10248933.
[40] J. Kim and B. Yang, "A smart city service business model: Focusing on transportation services," *Sustainability*, vol. 13, no. 19, 2021, Art. no. 10832. doi: 10.3390/su131910832.
[41] J. K. Eom, K. S. Lee, S. Ko, and J. Lee, "Exploring travel mode preference of external trips for smart city transportation planning: Sejong, Korea," *Sustainability*, vol. 14, no. 2, 2022, Art. no. 630. doi: 10.3390/su14020630.
[42] H. Ugale, P. Patil, S. Chauhan, and N. Rao, "Design of intelligent transportation system for smart city," in *Lecture Notes in Electrical Engineering*, vol. 825, pp. 157–167, 2022. doi: 10.1007/978-981-16-7637-6.
[43] K. L. M. Ang, J. K. P. Seng, E. Ngharamike, and G. K. Ijemaru, "Emerging technologies for smart cities' transportation: Geo-information, data analytics and machine learning approaches," *ISPRS Int. J. Geo-Info.*, vol. 11, no. 2, 2022, Art. no. 85. doi: 10.3390/ijgi11020085.
[44] C. S. Lai *et al.*, "A review of technical standards for smart cities," *Clean Technol.*, vol. 2, no. 3, pp. 290–310, 2020. doi: 10.3390/cleantechnol2030019.

[45] R. Kumar, S. Goel, V. Sharma, L. Garg, K. Srinivasan and N. Julka, "A multifaceted vigilare system for intelligent transportation services in smart cities," *IEEE Internet Things Mag.*, vol. 3, no. 4, pp. 76–80, 2021. doi: 10.1109/IOTM.0001.2000041.

[46] M. Whaiduzzaman *et al.*, "A review of emerging technologies for IoT-based smart cities," *Sensors*, vol. 22, no. 23, pp. 1–28, 2022. doi: 10.3390/s22239271.

[47] A. Abdollahi, J. G. Keogh, and M. Iranmanesh, "Smart city research: A bibliometric and main path analysis," *J. Data, Inf. Manage.*, vol. 4, pp. 343–370, 2022. doi: 10.1007/s42488-022-00084-4.

[48] A. Bruska and N. Boichuk, "Is sustainable aligning with smartness in transport domain ?-marketing perspective of smart city rankings," *Sci. Journals Marit. Univ. Szczecin*, vol. 2023, pp. 1–8, 2022.

[49] M. Mazhar Rathore, A. Ahmad, and A. Paul, "IoT-based smart city development using big data analytical approach," in *2016 IEEE Int. Conf. Autom. ICA*, 2016.

[50] C. Gokulnath, J. Marietta, R. Deepa, R. Senthil Prabhu, M. Praveen Kumar Reddy and B. R. Kavitha, "Survey on IOT based smart city," *Int. J. Comput. Trends Technol.*, vol. 46, no. 1, pp. 23–28, 2017. doi: 10.14445/22312803/IJCTT-V46P105.

[51] S. M. Wu, T. Chun Chen, Y. J. Wu, and M. Lytras, "Smart cities in Taiwan: A perspective on big data applications," *Sustainability*, vol. 10, no. 1, pp. 1–14, 2018. doi: 10.3390/su10010106.

[52] X. Huang, J. Fan, Z. Deng, J. Yan, J. Li and L. Wang, "Efficient IoT data management for geological disasters based on big data-turbocharged data lake architecture," *ISPRS Int. J. Geo-Info.*, vol. 10, no. 11, 2021, Art. no. 743. doi: 10.3390/ijgi10110743.

[53] R. Sánchez-De-Madariaga, A. Muñoz, A. L. Castro, O. Moreno, and M. Pascual, "Executing complexity-increasing queries in relational (MySQL) and NoSQL (MongoDB and exist) size-growing ISO/EN 13606 standardized EHR databases," *J. Vis. Exp.*, vol. 2018, no. 133, pp. 1–11, 2018.

[54] G. Ding, H. Sun, J. Li, C. Li, R. Wei and Y. Fei, "An efficient relational database keyword search scheme based on combined candidate network evaluation," *IEEE Access*, vol. 8, pp. 30863–30872, 2020. doi: 10.1109/ACCESS.2020.2973217.

[55] K. A. Eldahshan, A. A. Alhabshy, and G. E. Abutaleb, "A comparative study among the main categories of NoSQL databases," *Al-Azhar Bull. Sci.*, vol. 31, no. 2, pp. 51–60, 2020.

[56] H. Matallah, G. Belalem, and K. Bouamrane, "Comparative study between the MySQL relational database and the MongoDB NoSQL database," *Int. J. Softw. Sci. Comput. Intell.*, vol. 13, no. 3, pp. 38–63, 2021. doi: 10.4018/IJSSCI.

[57] A. Hillenbrand, U. Störl, S. Nabiyev, and M. Klettke, "Self-adapting data migration in the context of schema evolution in NoSQL databases," *Distrib. Parall. Databases*, vol. 40, no. 1, pp. 5–25, 2022. doi: 10.1007/s10619-021-07334-1.

[58] W. McClay, "A Magnetoencephalographic/Encephalographic (MEG/EEG) brain-computer interface driver for interactive iOS mobile videogame applications utilizing the hadoop ecosystem, MongoDB, and Cassandra NoSQL databases," vol. 6, no. 4, pp. 1–34, 2018.

[59] S. Khan, X. Liu, S. A. Ali, and M. Alam, "Bivariate, cluster, and suitability analysis of NoSQL solutions for big graph applications," *Adv. Comput.*, pp. 0–48, 2022.

[60] G. A. Schreiner, D. Duarte, and R. S. Dos Mello, "When relational-based applications go to NoSQL databases: A survey," *Information*, vol. 10, no. 7, pp. 1–22, 2019. doi: 10.3390/info10070241.

[61] F. Hamami, I. A. Dahlan, S. W. Prakosa, and K. F. Somantri, "Big data analytics for processing real-time unstructured data from CCTV in traffic management," in *2020 Int. Conf. Data Sci. its Appl. (ICoDSA 2020)*, 2020, pp. 6–10.

[62] D. Mahajan, C. Blakeney, and Z. Zong, "Improving the energy efficiency of relational and NoSQL databases via query optimizations," *Sustain. Comput. Inform. Syst.*, vol. 22, pp. 120–133, 2019. doi: 10.1016/j.suscom.2019.01.017.

[63] AichaAggoune, "An overview on the mapping techniques in NoSQL databases," *Int. J. Info. Appl. Math.*, vol. 3, no. 2, pp. 53–65, 2020.

[64] Nurhadi, R. A. Kadir, and E. S. M. Surin, "Complex SQL-NoSQL query translation for data lake management," *J. Comput. Sci.*, vol. 18, no. 12, pp. 1179–1188, 2022. doi: 10.3844/jcssp.2022.1179.1188.

[65] O. Alotaibi and E. Pardede, "Transformation of schema from relational database (RDB) to NoSQL databases," *Data*, vol. 4, no. 4, pp. 1–11, 2019. doi: 10.3390/data4040148.
[66] S. Sicari, A. Rizzardi, and A. Coen-Porisini, "Security&privacy issues and challenges in NoSQL databases," *Comput. Netw.*, vol. 206, 2022, Art. no. 108828. doi: 10.1016/j.comnet.2022.108828.
[67] M. Garba, "A comparison of NoSQL and relational database management systems (RDBMS)," *Kasu J. Math. Sci.*, vol. 1, no. 2, pp. 61–69, 2020.
[68] L. Marrero, V. Olsowy, F. Tesone, P. Thomas, L. Delia and P. Pesado, "Performance analysis in NoSQL databases, relational databases and NoSQL databases as a service in the cloud," in *Communications in Computer and Information Science*, Cham: Springer, 2021, vol. 1409, pp. 157–170. 10.1007/978-3-030-75836-3_11.
[69] A. Gupta, S. Tyagi, N. Panwar, S. Sachdeva, and U. Saxena, "NoSQL databases: Critical analysis and comparison," in *2017 Int. Conf. Comput. Commun. Technol. Smart Nation, IC3TSN 2017*, 2018, pp. 293–299.
[70] N. Chaudhry and M. M. Yousaf, *Architectural Assessment of NoSQL and NewSQL Systems*. USA: Springer, 2020, vol. 38, no. 4.
[71] H. B. S. Reddy, R. R. S. Reddy, R. Jonnalagadda, P. Singh, and A. Gogineni, "Analysis of the unexplored security issues common to all types of NoSQL databases," *Asian J. Res. Comput. Sci.*, vol. 14, pp. 1–12, 2022. doi: 10.9734/ajrcos/2022/v14i130323.
[72] B. Lakhe, "Practical hadoop migration: How to integrate your RDBMS with the hadoop ecosystem and re-architect relational applications to NoSQL," 2016. Accessed: Jul. 12, 2024. https://www.amazon.com/Practical-Hadoop-Migration-Re-Architect-Applications/dp/1484212886.
[73] P. P. Khine and Z. S. Wang, "Data lake: A new ideology in big data era," *ITM Web Conf.*, vol. 17, 2018, Art. no. 3025. doi: 10.1051/itmconf/20181703025.
[74] S. E. E. Profile, "Information technology-new generations," *Adv. Intell. Syst. Comput.*, vol. 558, pp. 443–451, 2018.
[75] C. Li and J. Gu, "An integration approach of hybrid databases based on SQL in cloud computing environment," *Softw. -Pract. Exp.*, vol. 49, no. 3, pp. 401–422, 2019. doi: 10.1002/spe.2666.
[76] N. Miloslavskaya and A. Tolstoy, "Application of big data, fast data, and data lake concepts to information security issues," in *Proc. 2016 4th Int. Conf. Futur. Internet Things Cloud Work. (FiCloudW)*, Vienna, Austria, 2016, pp. 148–153.
[77] S. S. Hong, J. Lee, S. Chung, and B. Kim, "Fast real-time data process analysis based on NoSQL for IoT pavement quality management platform," *Appl. Sci.*, vol. 13, no. 1, 2023, Art. no. 658. doi: 10.3390/app13010658.
[78] T. N. Khasawneh, M. H. Al-Sahlee, and A. A. Safia, "SQL, NewSQL, and NOSQL databases: A comparative survey," in *2020 11th Int. Conf. Inf. Commun. Syst. (ICICS 2020)*, 2020, pp. 13–21.
[79] B. Malysiak-Mrozek, M. Stabla, and D. Mrozek, "Soft and declarative fishing of information in big data lake," *IEEE Trans. Fuzzy Syst.*, vol. 26, no. 5, pp. 2732–2747, 2018. doi: 10.1109/TFUZZ.2018.2812157.
[80] K. Stockinger, R. Bödi, J. Heitz, and T. Weinmann, "ZNS-efficient query processing with ZurichNoSQL," *Data Knowl. Eng.*, vol. 112, pp. 38–54, 2017.
[81] Z. Aftab, W. Iqbal, K. M. Almustafa, F. Bukhari, and M. Abdullah, "Automatic NoSQL to relational database transformation with dynamic schema mapping," *Sci. Program.*, vol. 7, pp. 1–13, 2020. doi: 10.1155/2020/8813350.
[82] K. A. ElDahshan, A. A. A. AlHabshy, and G. E. Abutaleb, "Data in the time of COVID-19: A general methodology to select and secure a NoSQL DBMS for medical data," *PeerJ Comput. Sci.*, vol. 6, 2020, Art. no. e297. doi: 10.7717/peerj-cs.297.
[83] A. B. M. Moniruzzaman and S. A. Hossain, "NoSQL database: New era of databases for big data analytics-classification, characteristics and comparison," *J. Humanit. Appl. Sci.*, no. 29, pp. 1–21, 2013.
[84] K. J. Kim and H. -Y. Kim, "Lecture notes in electrical engineering 621 information science and applications," in *Info. Sci. Appl. Proc., ICISA 2020*, 2019, p. 384.

[85] M. J. Aqel, A. Al-Sakran, and M. Hunaity, "A comparative study of NoSQL databases," *Biosci. Biotech. Res. Commun.*, vol. 12, no. 1, pp. 17–26, 2019. doi: 10.21786/bbrc/12.1/3.

[86] K. Sultan, H. Ali, and Z. Zhang, "Big data perspective and challenges in next generation networks," *Futur. Internet*, vol. 10, no. 7, pp. 1–20, 2018. doi: 10.3390/fi10070056.

[87] D. Preuveneers and W. Joosen, "Automated configuration of NoSQL performance and scalability tactics for data-intensive applications," *Informatics*, vol. 7, no. 3, 2020, Art. no. 29. doi: 10.3390/informatics7030029.

[88] A. A. Imam, S. Basri, R. Ahmad, and M. T. González-Aparicio, "Schema proposition model for NoSQL applications," *Adv Intell. Syst. Comput.*, vol. 843, pp. 30–39, 2019. doi: 10.1007/978-3-319-99007-1.

[89] E. A. Khashan, A. I. El Desouky, and S. M. Elghamrawy, "A framework for executing complex querying for relational and NoSQL databases (CQNS)," *Eur. J. Electr. Eng. Comput. Sci.*, vol. 4, no. 5, pp. 1–12, 2020. doi: 10.24018/ejece.2020.4.5.195.

[90] P. Lehotay-Kéry, T. Tarczali, and A. Kiss, "P system-based clustering methods using nosql databases," *Computation*, vol. 9, no. 10, pp. 1–18, 2021.

[91] A. Mohan, M. Ebrahimi, S. Lu, and A. Kotov, "A NoSQL data model for scalable big data workflow execution," in *Proc. 2016 IEEE Int. Congr. Big Data (BigData Congr.)*, Washington, DC, USA, 2016, pp. 52–59.

[92] S. V. Oprea and A. Bara, "Machine learning algorithms for short-term load forecast in residential buildings using smart meters, sensors and big data solutions," *IEEE Access*, vol. 7, pp. 177874–177889, 2019. doi: 10.1109/ACCESS.2019.2958383.

[93] H. El Massari, S. Mhammedi, and N. Gherabi, "Bridging the gap between the semantic web and big data: Answering SPARQL queries over NoSQL databases," *Int. J. Electr. Comput. Eng.*, vol. 12, no. 6, pp. 6829–6835, 2022. doi: 10.11591/ijece.v12i6.pp6829-6835.

[94] N. Soussi, A. Boumlik, and M. Bahaj, "Mongo2SPARQL: Automatic and semantic query conversion of MongoDB query language to SPARQL," in *2017 Intell. Syst. Comput. Vis., ISCV 2017*, 2017.

[95] M. A. Abdel-fattah, W. Mohamed, and S. Abdelgaber, "A comprehensive spark-based layer for converting relational databases to NoSQL," *Big Data Cogn. Comput.*, vol. 6, no. 3, pp. 1–28, 2022.

[96] A. Almassabi, O. Bawazeer, and S. Adam, "Top NewSQL databases and features," *Int. J. Database Manage. Syst. (IJDMS)*, vol. 10, no. 2, pp. 11–31, 2018.

[97] A. Nambiar and D. Mundra, "An overview of data warehouse and data lake in modern enterprise data management," *Big Data Cogn. Comput.*, vol. 6, no. 4, 2022, Art. no. 132.

[98] G. B. Solanke and K. Rajeswari, "SQL to NoSQL transformation system using data adapter and analytics," in *Proc. -2017 IEEE Int. Conf. Technol. Innov. Commun. Control Autom., TICCA 2017*, Chennai, India, 2018, pp. 59–63.

[99] H. Lee, *Schema Design for NoSQL Wide Column Stores Schema Design for NoSQL Wide Column Stores*, Woodstock, New York, Association for Computing Machinery, 2022, vol. 1, no. 1.

[100] R. Sellami and B. Defude, "Complex queries optimization and evaluation over relational and NoSQL data stores in cloud environments," *IEEE Trans. Big Data*, vol. 4, no. 2, pp. 217–230, 2017. doi: 10.1109/TBDATA.2017.2719054.

[101] M. N. Mami, D. Graux, S. Scerri, H. Jabeen, and S. Auer, "Querying data lakes using spark and presto," in *Web Conf. 2019-Proc. World Wide Web Conf., WWW 2019*, San Fransisco, CA, USA, 2019, pp. 3574–3578.

[102] J. Ziegler, P. Reimann, F. Keller, and B. Mitschang, "A graph-based approach to manage CAE data in a data lake," *Procedia CIRP*, vol. 93, pp. 496–501, 2020. doi: 10.1016/j.procir.2020.04.155.

[103] H. Dibowski, S. Schmid, Y. Svetashova, C. Henson, and T. Tran, "Using semantic technologies to manage a data lake: Data catalog, provenance and access control," *CEUR Workshop Proc.*, vol. 2757, pp. 65–80, 2020.

[104] E. Khashan, A. Eldesouky, and S. Elghamrawy, "An adaptive spark-based framework for querying large-scale NoSQL and relational databases," *PLoS One*, vol. 16, pp. 1–25, 2021. doi: 10.1371/journal.pone.0255562.

[105] V. Piccialli and M. Sciandrone, "Nonlinear optimization and support vector machines," *Ann Oper. Res.*, vol. 314, no. 1, pp. 15–47, 2022. doi: 10.1007/s10479-022-04655-x.
[106] J. Nalepa and M. Kawulok, "Selecting training sets for support vector machines: A review," *Artif. Intell. Rev.*, vol. 52, no. 2, pp. 857–900, 2019. doi: 10.1007/s10462-017-9611-1.
[107] Y. C. Zhang and L. Sakhanenko, "The Naive Bayes classifier for functional data," *Stat Probab. Lett.*, vol. 152, pp. 137–146, 2019. doi: 10.1016/j.spl.2019.04.017.
[108] F. Davardoost, A. B. Sangar, and K. Majidzadeh, "An innovative model for extracting OLAP cubes from NOSQL database based on scalable Naïve Bayes classifier," *Math. Probl. Eng.*, vol. 2022, pp. 1–11, 2022. doi: 10.1155/2022/2860735.
[109] J. Agnelo, N. Laranjeiro, and J. Bernardino, "Using orthogonal defect classification to characterize NoSQL database defects," *J. Syst. Softw.*, vol. 159, 2020, Art. no. 110451. doi: 10.1016/j.jss.2019.110451.
[110] S. Ghule and R. Vadali, "Transformation of SQL system to NoSQL system and performing data analytics using SVM," in *Proc. Int. Conf. Trends Electron. Informatics, ICEI 2017*, Tirunelveli, India, 2018, pp. 883–887.