International Journal of Emerging Trends in Computer Science and Information Technology
ISSN: 3050-9246 | https://doi.org/10.63282/3050-9246.IJETCSIT-V5I4P118
Eureka Vision Publication | Volume 5, Issue 4, 156-165, 2024

Original Article

Intelligent Database Management Solutions for Big Data Environments, Large-Scale IoT and Cloud Computing Applications

Srikanth Reddy Keshireddy[1], Venkata Teja Nagumotu[2], Harsha Vardhan Reddy Kavuluri[3], Akhil Kumar Pathani[4], Ajay Dasari[5], Venkata Kishore Chilakapati[6]
[1]Senior Software Engineer, Keen Info Tek Inc.
[2]Sr Network Engineer, Techno-bytes Inc.
[3]Lead database administrator, Wissen infotech Inc.
[4]Network Engineer, Ebay.
[5]Senior Support Engineer, Microsoft.
[6]Support Escalation Engineer, Microsoft.

Abstract - The transformation of the data ecosystem in a contemporary way has been largely influenced by the rapid growth of Big Data, the extensive application of the Internet of Things (IoT), and cloud computing. This has led to a whole new range of issues related to storage, processing, indexing, scalability, and smart decision-making that were not present before. Relational database systems have limited possibilities to follow the speed, volume, and variety requirements of such environments. This review is focused on intelligent models of database management systems that use AI techniques for improving automation, adaptability, and performance in next-generation databases intelligent big-data-oriented database models, distributed and parallel data models, IoT-oriented time-series management systems, and cloud-oriented intelligent storage and query strategies. It also reviews the use of NoSQL/NewSQL models, graph stores, context-aware IoT data management, edge as well as fog/cloud integration, and hybrid/multi-cloud database strategy implementation that revolve around scalability, data heterogeneity, privacy, interoperability, and real-time analytics future research challenges of intelligent database systems in digital ecosystems at a large scale.

Keywords - Big Data, Intelligent Database Management, Nosql, Iot Data Management, Cloud Computing, Distributed Systems, AI-Enabled Databases.

1. Introduction

The fast growth of digital ecosystems has led to a call for smart database management that can handle intricate, changing, and vast data operations. Conventional database systems cannot manage the complexity of modern applications any longer, where automation, adaptability, and real-time optimization are the requirements [1]. DBMSs are complex, mission-critical software systems Since companies are dependent on data-driven decision-making more and more, intelligent database solutions become the basis for the improvement of the system's performance [2][3], its reliability, and the organization's operational efficiency.

Big Data situations create large and complicated datasets that have high volume, velocity, and variety characteristics and require extremely advanced processing and management capabilities. Smart DBMSs use ML, PA, and automated tuning to optimize performance and resource utilization. By doing real-time analysis, scalable distributed storage, and adaptive data handling strategies, these intelligent systems are able to convert raw, unstructured data into understandable information [4]. Hence, they become a very efficient tool in solving the problem of increasing data complexity while, at the same time, satisfying the requirements for fast, accurate, and efficient decision-making.

The large-scale IoT networks that keep sending sensor data streams from billions of interconnected devices, the need for advanced data management has become very critical. These kinds of environments require ultra-low latency processing, very strong integration of heterogeneous data, and smart anomaly detection to guarantee system accuracy and stability [5][6]. Intelligent databases fulfill these requirements by allowing very fast data ingestion, context-aware processing, and automated decision-making capabilities. All these functionalities, in a real-time setup, make monitoring and scalability possible, thus emphasizing the indispensable role of intelligent database solutions in the upkeep of efficient and resilient IoT ecosystems as they grow worldwide.

Cloud computing applications in digital infrastructures serve as the scalable and on-demand platforms that are necessary to accommodate both Big Data and IoT operations. Smart database systems in the cloud further raise the bar for this ecosystem by providing elastic scalability, automated resource optimization, and efficient distributed processing [7][8]. With these features, companies can balance fluctuating workloads and, at the same time, keep a high level of performance and reliability [9]. As a result, the integration of Big Data, IoT, and cloud technologies represents a single operational framework in which smart database systems are the main support that makes the uninterrupted data flow, advanced analytical capabilities, and operational stability over time possible.

1.1. Structure of the Paper

This review covers intelligent database management for Big Data, IoT, and cloud environments. Section I introduces AI-driven DB needs, Section II discusses architectures, Section III IoT solutions, Section IV cloud strategies, Section V literature, and concludes with challenges and future directions.

2. Intelligent Database Management Frameworks for Big Data Environments

The challenges of big data systems with single-architectural solutions have also given rise to intelligent database management frameworks. It is necessary to revaluate these data management platforms in light of the numerous new problems and possibilities brought forth by big data, while keeping some of the more positive features. Collecting, storing, processing, and visualizing data are the four main components of a Big Data Management System (BDMS), which is a complicated set of functionalities. Because of its multi-purpose design, BDMS architecture is more adaptable to a wide range of needs than that of conventional database management systems [10]. BDMS as datar, an overarching paradigm for BDMS development and design [11]. As seen in Figure 1, the integrated big data paradigm encompasses integrated representation, storage and administration, processing, and visual analysis.

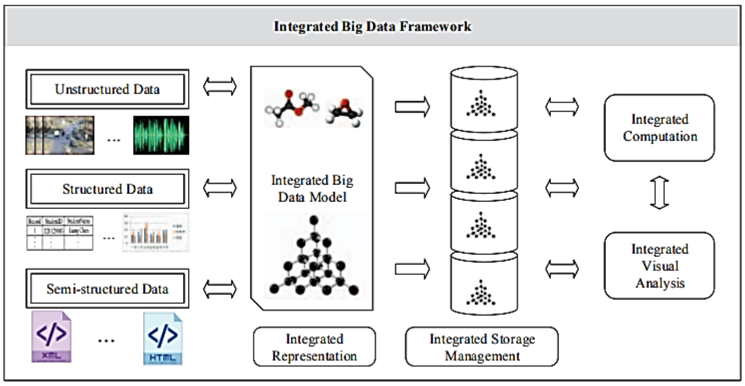

Figure 1. Framework of Integrated Big Data in Database System

- The SSU data is unified by integrated representation, which also maps huge data onto the same space.
- A unified storage architecture with the goal of effective management including speedy insert, delete, update, query, and joint operation on big data views SSU data as a "same type," storing huge data.
- Deep computing approaches can understand the intrinsic characteristics and distill buried information in huge data; integrated computation deals with SSU data simultaneously based on integrated representation and storage. Learning features, analyzing associations, clustering, classifying, and predicting are all aspects of the integrated computation.
- Big data's attributes and associations can be presented in a multivariate and associated format using integrated visual analysis, which uses SSU data for unified display.

A unified model can be employed to characterize the SSU data. And lastly, a compatible and consistent storage model is used to store the united heterogeneous data. And so, big data computing.

2.1. Distributed and Parallel Database Models for Big Data

In this age of Big Data, there are several pressing issues that need immediate and coordinated responses from different scientific groups. These include the following: the increasing difficulty in controlling the flow of data; the weakening of data engineers' faith in conventional methods of data management and querying; the slow but steady rise of efficient and reliable data models; and the overwhelming desire for data-driven applications. These are the characteristics that should characterize a perfect NoSQL model: low latency, efficient storage, high availability, high scalability, and diminished administrative and operational expenses. MongoDB is well-suited for master-slave replication. Master processes reads and writes, whereas slaves are limited to copying data received from masters, reading data, and backing up data. Orient DB is a distributed database that may be spread out over multiple servers. Modern, cutting-edge Multi master replication distributed system is supported. The combination of a map and a reduction function in CouchDB promotes parallelism [12] and independent counting. Classification is used to classify the models together:

- Document-Oriented Store.
- Graph Data Model.

2.1.1. Document-Oriented Store

A primary type of non-relational database is the document-oriented database. For the storage, management, and retrieval of structured or semi-structured data in document form, document-oriented databases are utilized. The "document" is the central notion in document-oriented databases; it's similar to a record in relational databases but has numerous differences, such as a more flexible structure and more formats for storing data.

- Mongo dB: The MongoDB database system is free and open-source. Unlike "classical" relational databases, which store data in tables, MongoDB stores structured data as JSON-like documents with dynamic schemas. This allows for faster and easier data integration in specific applications. It stands as a C++ program.
- CouchDB: The document storage database CouchDB falls under the "NoSQL" database category. In contrast to databases that use schemas, CouchDB databases do not have pre-defined data structures like tables. One-way CouchDB stores data is as a JSON document. As requirements change, the data or document's structure can adapt on the fly. Written in the Erlang programming language.
- Ravendb: Raven DB provides a versatile data model tailored to meet the needs of real-world applications; it is an open-source, transactional Document Database built in.NET. For efficient querying, use Linq queries from.NET code or the Restful API with other tools; Raven DB stores data schema-less as JSON documents.

2.1.2. Graph Data Model:

Graph database systems provide operations like neighbourhood traversal and pattern matching that are based on graphs, and they represent data using graph structures. Most of the systems that were taken into consideration use either the property graph model (PGM) or the resource description [13] framework (RDF) as their data model. For some systems, generic graph models are being utilized.

2.1.2.1. Application scope of Graph Data Model

- OLTP workload: Create, read, update, and delete (CRUD) operations on edges and vertices.
- Graph analytics: execution of graph algorithms like page rank or detecting frequent substructures.

2.1.2.2. Storage techniques of Graph Data Models

- Native storage approach: customized to the properties of graph database models to facilitate efficient traversal of edges.
- Computing clusters: consolidating the database onto multiple nodes to enhance read performance.
- Partitioned graph storage: processing queries across several nodes.

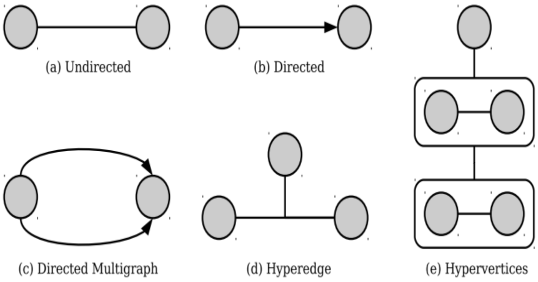

Figure 2. Structural Comparison of Graph Models.

Those are some of the typical graph representations shown in the Figure 2.

- Undirected graph: Edges are not directed meaning that there is a relationship that is reciprocal or two ways.
- Directed graph: There are arrowheads between the edges that define the direction of relationship between one node and another.
- Directed multigraph: Multiple directed edges are permitted between two nodes, and there may be parallel or repeated edges.
- Hyperedge: There is the possibility of an edge between more than two nodes and this is used to describe group relationships.
- Hypervertices: Organize the nodes in higher level composite units so that hierarchical or clustered structures can be represented in the graph structure.

2.2. Intelligent Storage, Indexing and Retrieval for Big Data Enviornments

In the last few decades, the Internet has made it very easy to get too much knowledge. Every day, Yahoo! and Twitter exchange more than three billion messages, while Twitter receives more than 100 million tweets. Finding data samples that are comparable within a specific database is largely related to the core issue of closest neighbour search. In practical large-scale contexts, the linear time complexity makes exhaustively comparing a query point with each database sample impractical. [14]

For decades, hashing algorithms have been the subject of much research and found widespread use across numerous domains. For quick inverse lookups, see Figure 3 for an illustration of the linear projection-based binary hashing, indexing, and hash table generation process.

- Storage: A hash code can save storage costs by a factor of hundreds—if not thousands—of times. The original size of 80 million little photos (32 × 32 pixels) was approximately 600G bytes, but they can be reduced to 600M bytes when compressed using 64-bit binary coding.
- Indexing: A hash table is used for inverse table lookups, where hash codes are organized. The amount of time it takes to do a hash table lookup is constant. The KD tree, the ball tree, the metric tree, and the vantage point tree are all examples of tree-based indexing methods.
- Retrieval: The ideal indexing strategy for artificial neural networks (ANNs) would have query times that are either constant or logarithmic, or at least very quick. Quickly calculate the Hamming distance by utilizing the logical xor function between binary codes.

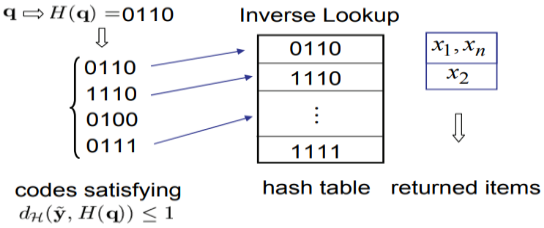

Figure 3. Illustration of Linear Projection

The Figure 3 depicts a hash-based look up procedure. The query q is first coded with a binary hash coded H(q) = 0110. Each of the codes within a Hamming distance at most 1 of this code is then generated-e.g. 0110, 1110, 0100, 0111. These code numbers find an inverse look up in a hash table where each of the code numbers points to the stored items. The items represented by these similar hash bucket.

3. Intelligent Database Solutions for Large-Scale Iot Ecosystems

The Internet of Things (IoT) is becoming more popular, which means that data management methods need to change. IoT solutions require integration and storage of massive amounts of data collected from sensors in the form of time-series input streams [15]. IoT apps that aim to deliver value across several domains in real-time must be able to swiftly retrieve, process, and analyze this data. Therefore, it is essential that IoT data management systems enable the gathering, integrating, and analyzing of time-series data.

- TritanDB solution: An IoT-specific solution, TritanDB, was developed by studying the properties of IoT time-series data, how it can be better stored, indexed, and retrieved; how it is represented in RDF; and the structure of analytical queries. TritanDB outperforms other state-of-the-art time-series, NoSQL, and relational databases in terms of storage space, reads, and writes; and it supports rich data models, such as RDF, which promote semantic interoperability [16]. An innovative design for an IoT time-series database is derived from this, utilizing a re-ordering buffer and an immutable, time-partitioned store to optimize the storage and retrieval of time-series data in the realms of data structures, indexing, and compression.

3.1. Time Series Management Systems in IoT Ecosystems:

The capacity to efficiently evaluate the collected data and the proliferation of sensors to monitor massive industrial systems have made possible the unprecedented scale of use of automation and remote management. The data collected by any given sensor network can be expressed as a time series of values, regardless of the size of the network (from a single smart light bulb to hundreds of wind turbines spread out over a vast region). Time series are collections of data points that are ordered by time and can be finite or infinite. Time series can be used to represent sensor readings [17], therefore it's important to find ways to efficiently transfer, store, and analyze them. This allow sensor networks to grow in size and be employed in more areas.

3.2. Context-Aware and Adaptive Data Management in IOT:

A growing number of sensors are being placed all around the globe. These sensors constantly produce massive volumes of data. There is no use in gathering data from all the sensors unless those sensors can provide useful insights that aid in solving the problems. When there are a lot of sensors to pick from, it can be difficult and time-consuming to find the right one. The concept for sensing as a service is anticipated to be constructed atop the services and infrastructure of the Internet of Things. Additionally, it foresees a future when sensors may be accessed online using middleware solutions, and their usage can be either free or paid for. a number of middleware solutions are now in development with the expectation that they enable this architecture [18]. The integration of sensing devices with software systems and associated functionality is the primary emphasis of these middleware solutions. However, search functionality becomes crucial as the number of Internet-connected sensors increases.

4. Cloud-Centric Intelligent Database Management

Since the last decade, as heterogeneous systems emerged on the DBMS landscape, the classification of various data models for semi-structured data has been a focus. A result of this diversity in approaches to semi-structured data is the evolution of the acronym NoSQL. In a similar vein, NewSQL DBMSs ushered in unprecedented changes to the structured data paradigm. In recent years, a number of surveys have examined NoSQL and NewSQL data models, as well as the current DBMS, considering both the data models and the particular non-functional elements related to them. Horizontal scalability, managing elastic workload patterns [19], and fault tolerance are cloud-centric issues in conducting distributed DBMS operations. Not only that, it sorts nineteen different database management systems according to characteristics like consistency, security, partitioning, and replication.

4.1. Types of Cloud-Centric Database Management Models:

4.1.1. Relational Data Models

Data management solutions that are relational are essential in modern computing. Opportunities to provide database management systems as an outsourced service have been growing in recent years, thanks to hosted cloud computing and storage. This is true for both Amazon RDS and Microsoft SQL Azure. There are two main selling points of this kind of database as a service, or DBaaS. First, consumers could expect significantly reduced hardware and energy prices as a result of economies of scale. In addition, a well-planned DBaaS have prices that are directly related to real consumption, sometimes known as "pay-per-use" [20]. Efficient multi-tenancy, elastic scale-out, and database privacy are the three constraints that shaped the architecture.

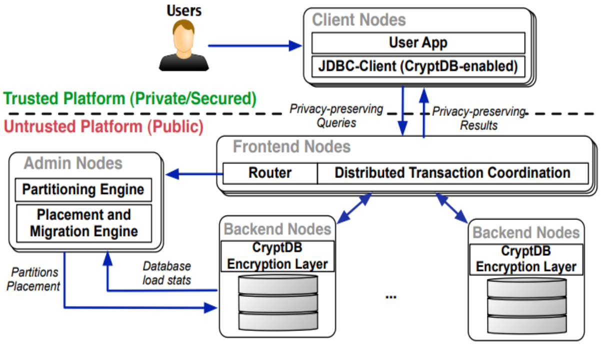

Figure 4. Relational Cloud Architecture

The architecture that is discussed in the Figure 4 is a secure and distributed architecture using Crypt DB. Privacy-preserving queries are made by client nodes with the help of Crypt DB-enabled JDBC client. Frontend nodes routes through these queries and distributes coordination of many transactions. Data are stored and processed by the backend nodes through an encryption layer of information which makes computation confidential. Admin nodes control the partitioning of data, its placement, and migration, and base optimization on workload statistics. The system isolates the trusted elements of the private system and the untrusted elements of the infrastructure and retains end to end encryption of query execution.

4.1.2. NewSQL

Modern relational database management systems like NewSQL aim to scale as well as NoSQL for online transaction processing (OLTP) read-write workloads while still ensuring that transactions are ACID-compliant. Aiming for the same scalability as NoSQL DBMSs from the 2000s, these systems retain the relational paradigm (with SQL) and transaction support of legacy DBMSs from the 1970s and 1980s. Because of this, applications can use SQL (rather than a proprietary API) to run many concurrent transactions, allowing the database to absorb new information and change its state. While working with a NewSQL DBMS, developers are spared the [21] hassle of dealing with eventually consistent updates, unlike with a NoSQL system.

4.1.3. Time-Series

Several current compression algorithms were evaluated in order to decrease storage overhead, and the feasibility of establishing an in-memory time series database was assessed. Data points inside a time series are compressed by Gorilla without any extra compression applied across time series. There are a time stamp and a value associated with each data point, and they are both 64 bits in length. Using history data, timestamps and values are compressed independently. Figure 4 exhibits the interleaving of time stamps and values in the compressed block, which is a visual representation of the overall compression technique [22]. As an alternative to keeping complete timestamps.

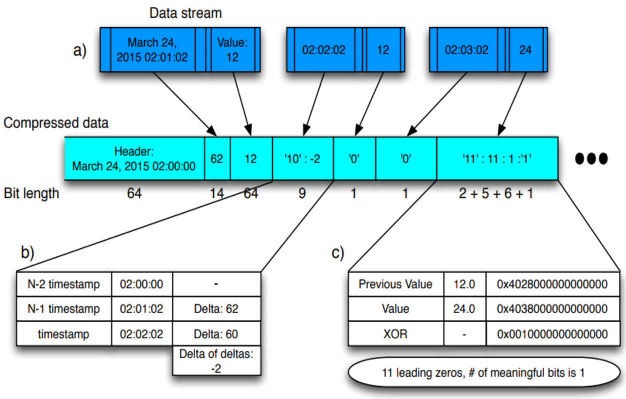

Figure 5. Visualizing the Entire Compression Algorithm.

The Figure 5 illustrates the compression of a time-series, with the help of timestamp deltas and XOR-based encoding of values. The storing of timestamps is based on differences (and delta-of-delta to further reduce) and storing the values is based on the XORing of values with the previous value and storing only the meaningful bits. This results in a binary stream which is small in size and it maintains the original data.

4.2. Multi-Cloud and Hybrid Cloud Database Integration:

The ability to access, save, and transmit data via the Internet is fundamental to the development and maintenance of institutions and individuals. File storage technologies like scalable parallel file systems and robust Cloud computing platforms have been mentioned previously [23]. Data storage on the cloud opens up new possibilities for providers to provide services, which in turn opens up new markets for them. The proliferation of suppliers offering cloud storage services (e.g., Dropbox, Google Drive, Copy, Amazon S3, OneDrive, etc.) is driving enterprise growth in this space. The services provided by cloud storage companies are not always credible. For instance, files may become temporarily or permanently unavailable for a variety of causes, such as a hardware or software failure, an unexpected surge in client demands, a Denial of Service (DoS) assault, or legal issues like bankruptcy.

5. Literature Review

In this section, recent studies underline the rising importance of intelligent database management for big data, large-scale IoT, and cloud environments. They show how smarter data handling and automated processing improve system performance. Table I presents a focus area, methodological approach, key findings, existing challenges, and suggested future research directions.

Wannalai and Mekruksavanich (2019) Businesses can greatly benefit from the database management system when it comes to tackling this problem. One problem with traditional database management systems is their inability to effectively handle massive amounts of data and information, which leads to information anxiety. This paper proposes an integration strategy for building an AI-powered database system that brings together AI and a database [24].

Liu (2019) Revolutionary changes in society's way of life and thinking brought about by big data. However, there are a lot of security concerns with big data's huge data and its possible worth. A data privacy protection paradigm for big data platforms is suggested to address the aforementioned issues. The first step is an in-depth introduction to the big data privacy protection model for data owners. This includes the design of protocols, logic, complexity, and security analyses. Afterwards, the paradigm for protecting ordinary users' privacy when querying massive data is presented in detail. This includes the design of both the query protocol and the query mode [25].

Wang and Tang (2019) The characteristics of online public opinion pose significant problems for current data analysis methodologies in this age of big data, thus it's important to assess its current state and future directions. Papers published between 2010 and 2017 that involved online public opinion research in a big data environment are compiled in this publication using CNKI. In this review, Use bibliometrics analysis and the CiteSpace5.0 visualization application software to identify the most promising areas and cutting-edge trends in online public opinion research conducted in a big data setting. This section examines the current state of the field, its difficulties, and potential future research directions. The aim is that this work might serve as a foundation for similar initiatives in the future [26].

Qu, Liu and Meng (2018) data produced in real-time by CIR devices in vast quantities. Incorporating big data technologies for its collection, pre-processing, and storage. Next, a revised Tri-training algorithm is developed to meet the real needs of CIR fault detection. Lastly, a strategy for CIR fault diagnosis is put forward, which is based on this enhanced algorithm. The results of the experiment reveal that the suggested approach is capable of making good use of unlabeled CIR data to boost the model's performance and guarantee accurate and real-time CIR problem identification. Its practicality is greatly enhanced [27].

Zhang and Wu (2018) a strategy for building an evaluation model for the allocation of multimedia instructional resources related to traditional Chinese culture in a big data setting. To begin, it uses a plane to randomly distribute Chinese traditional culture multimedia teaching resources; next, it chooses a group similarity coefficient from a small to a large teaching model with different teaching characteristics and uses it to conduct clustering analysis in a big data environment [28].

Ito et al. (2018) a novel Information-Centric Network (ICN) design that is well-suited to managing the metadata of data that is either permanently stored (called Live Data) or is generated continuously (usually by IoT devices). Taking the highly localized nature of live data into account, the suggested ICN offers two approaches. Data retrieval metadata is saved for queries in metadata databases (DB) or data generation terminals (publishers), depending on the data's popularity and location. When a subscriber issues a query, it is sent to either the DB or the publishers in the local area. The metadata databases are comprised of a single global database and numerous local databases, the latter of which covers exclusively local data and is used to offload the initial one [29].

Sadegh, Shahidi and Valaee (2017) the challenge of indoor localization in a large-scale schema using Wi-Fi fingerprinting, where the user is not localized on a single floor or within a single building, but rather to thousands of buildings and/or Wi-Fi access points (AP) in the fingerprinting database. investigation into the intricacy of the basic weighted kNN algorithm demonstrate that existing methods are insufficient for execution on such a massive database, on a mobile device, or in the cloud [30].

Table 1. Summary of Recent Studies on Intelligent Database Management, Big Data, Iot sand Cloud Systems

Reference	Study On	Approach	Key Findings	Challenges / Limitations	Future Directions
Wannalai and Mekruksavanich (2019)	Intelligent database systems for business data management	Integration of artificial intelligence techniques with traditional DBMS	Conventional DBMS fail with massive data; integrated AI+DB can reduce information overload and improve system intelligence	High complexity of integrating AI; limited scalability of traditional DBMS; lack of automation	Develop fully autonomous intelligent DB systems; improve large-scale data handling
Liu (2019)	Big data security and privacy protection	Privacy protection model for big data: protocol design, logic design, complexity & security analysis	Proposed a data-owner privacy model and user query privacy model; improves secure processing of large-scale data	Computational overhead; scalability issues in real-time privacy protection	Enhance efficiency of privacy models; integrate with distributed big data platforms
Wang and Tang (2019)	Online public opinion analysis in big data environments	Bibliometrics and CiteSpace-based trend analysis on 2010–2017 research	Identified hotspots, frontier trends, and gaps in public opinion analysis under big data	Data quality issues; insufficient predictive models; limited real-time analysis	Improve real-time public opinion mining; integrate AI for predictive public sentiment modeling
Qu, Liu and Meng (2018)	CIR device fault diagnosis using big data	Improved Tri-training algorithm using unlabeled data for fault detection	Enhanced accuracy in fault diagnosis; effective use of unlabeled CIR data; supports real-time detection	Requires continuous data preprocessing; may suffer from class imbalance issues	Expand to more IoT devices; increase robustness of semi-supervised learning
Zhang and Wu (2018)	Multimedia teaching resource allocation under big data	Clustering-based evaluation model using similarity coefficients and recursive allocation	Provides optimized teaching resource allocation; reveals distinct teaching characteristics	Limited generalizability; high dependency on clustering thresholds	Improve multi-source clustering; apply adaptive resource personalization models
Ito et al. (2018)	Metadata	Proposed ICN	Efficient retrieval	Requires complex	Improve metadata

	management for IoT live data	architecture with local/global metadata DBs	for short-lifecycle IoT data; reduces load on global DB using locality-based metadata distribution	metadata synchronization; storage overhead in local DBs	caching; enhance scalability for massive IoT deployments
Sadegh, Shahidi and Valaee (2017)	Large-scale Wi-Fi fingerprinting localization	Complexity analysis of weighted kNN for multi-building fingerprint DB	Current algorithms inefficient for huge DBs; significant latency in cloud and mobile execution	High computational cost; slow query response for large datasets	Develop lightweight localization algorithms; use ML-based dimensionality reduction

6. Conclusion and Future Work

Big Data environments, large-scale IoT deployments, and cloud-based infrastructures has made the use of smart database management solutions a necessity. The traditional database systems are not capable of handling the speed, heterogeneity, and complexity of the modern data streams, and thus they are less effective in dynamic and distributed settings. On the other hand, intelligent database add-on solutions that feature AI-driven optimization, distributed processing models, adaptive indexing, and context-aware data management, etc., extend the scalability, responsiveness, and overall system performance to a great extent. These database upgrades make it possible to extract insights that can be put into practice much faster while at the same time, they ensure trustworthiness in decentralized, resource-intensive architectures. Nevertheless, the issue of achieving seamless data integration, consistent performance, and efficient coordination across Big Data, IoT, and cloud platforms is still a major challenge that requires more research and innovation. Future Research in the future should focus on smart database management ought to emphasize on creating independent, self-learning database systems which can adjust themselves dynamically in Big Data, extensive IoT, and cloud computing settings. sophisticated AI techniques like deep learning and reinforcement learning may lead to significant improvements in query optimization in real-time, resource allocation based on prediction, and fault detection that is fully automated. There is still a lot of work to be done in ensuring seamless interaction between different data sources, and also in safeguarding data confidentiality while providing data security, besides making large-scale distributed systems energy efficient.

References

[1] J. M. Hellerstein, M. Stonebraker, and J. Hamilton, “Architecture of a Database System,” vol. 1, no. 2, pp. 141–259, 2007, doi: 10.1561/1900000002.

[2] V. M. L. G. Nerella, “MIGRATE: A Rollback-Enabled Framework for Automated Oracle XTTS-Based Cross-Platform Database Migrations,” *J. Electr. Syst.*, vol. 14, no. 4, pp. 85–95, Jan. 2018, doi: 10.52783/jes.9054.

[3] S. S. Reddy, “Comparative Analysis of CPU Scheduling Algorithms for Performance Efficiency,” 2019.

[4] M. Nasr and S. Ouf, “Cloud Computing: The Future of Big Data Management,” *Int. J. Cloud Appl. Comput.*, vol. 5, pp. 53–61, 2015.

[5] S. Garg, “Predictive Analytics and Auto Remediation using Artificial Inteligence and Machine learning in Cloud Computing Operations,” *Int. J. Innov. Res. Eng. Multidiscip. Phys. Sci.*, vol. 7, no. 2, 2019, doi: 10.5281/zenodo.15362327.

[6] A. Kushwaha, P. Pathak, and S. Gupta, “Review of Optimize Load Balancing Algorithms in Cloud.,” *Int. J. Distrib. Cloud Comput.*, vol. 4, no. 2, p. 1, 2016.

[7] A. F. Mohammed, V. T. Humbe, and S. S. Chowhan, “A Review of Big Data Environment and Its Related Technologies,” *Proc. IEEE*, 2016.

[8] J. Ruiz-Rosero, G. Ramirez-Gonzalez, J. Williams, H. Liu, R. Khanna, and G. Pisharody, “Internet of Things: A Scientometric Review,” *Symmetry (Basel).*, vol. 9, no. 12, p. 301, Dec. 2017, doi: 10.3390/sym9120301.

[9] S. Malallah, Y. Zalah, and R. Karne, “An Analysis of the Advanced Encryption Standard and Threats Associated,” 2018.

[10] Y. Wu and H. Guan, “biggy : An Implementation of Unified Framework for Big Data Management System,” 2018.

[11] Z. Chen, F. Zhong, X. Yuan, and Y. Hu, “Framework of integrated big data: A review,” in *2016 IEEE International Conference on Big Data Analysis (ICBDA)*, IEEE, Mar. 2016, pp. 1–5. doi: 10.1109/ICBDA.2016.7509815.

[12] S. Sharma, “An Extended Classification and Comparison of NoSQL Big Data Models,” pp. 1–23, 2015.

[13] M. Junghanns, A. Petermann, M. Neumann, and E. Rahm, “Management and Analysis of Big Graph Data : Current Systems and Open Challenges,” pp. 457–505, 2017.

[14] J. Wang, W. Liu, S. Kumar, and S. Chang, “Learning to Hash for Indexing Big Data - A Survey,” 2015.

[15] S. Basu and P. K. Pattnaik, “A consistency preservation based approach for data-intensive cloud computing environment,” in *2017 8th International Conference on Computing, Communication and Networking Technologies (ICCCNT)*, 2017, pp. 1–5. doi: 10.1109/ICCCNT.2017.8204097.

[16] E. Siow, T. Tiropanis, X. I. N. Wang, and W. Hall, “TritanDB : Time-series Rapid Internet of Things Analytics,” 2018.

[17] S. K. Jensen, T. B. Pedersen, and C. Thomsen, “Time Series Management Systems: A Survey,” *IEEE Trans. Knowl. Data Eng.*, vol. 29, no. 11, pp. 2581–2600, Nov. 2017, doi: 10.1109/TKDE.2017.2740932.

[18] C. Perera, A. Zaslavsky, C. H. Liu, M. Compton, P. Christen, and D. Georgakopoulos, “Sensor Search Techniques for Sensing as a Service Architecture for the Internet of Things,” *IEEE Sens. J.*, vol. 14, no. 2, pp. 406–420, Feb. 2014, doi: 10.1109/JSEN.2013.2282292.

[19] S. Mazumdar, D. Seybold, K. Kritikos, and Y. Verginadis, “A survey on data storage and placement methodologies for Cloud-Big Data ecosystem,” *J. Big Data*, vol. 6, no. 1, p. 15, Dec. 2019, doi: 10.1186/s40537-019-0178-3.

[20] C. A. Curino, E. P. C. Jones, R. A. Popa, and N. Malviya, “Relational Cloud : A Database-as-a-Service for the Cloud,” 2011.

[21] A. Pavlo and M. Aslett, “What’s Really New with NewSQL?,” *ACM SIGMOD Rec.*, vol. 45, no. 2, pp. 45–55, Sep. 2016, doi: 10.1145/3003665.3003674.

[22] T. Pelkonen *et al.*, “Gorilla: a fast, scalable, in-memory time series database,” *Proc. VLDB Endow.*, vol. 8, no. 12, pp. 1816–1827, Aug. 2015, doi: 10.14778/2824032.2824078.

[23] A. Celesti, A. Galletta, M. Fazio, and M. Villari, “Towards Hybrid Multi-Cloud Storage Systems: Understanding How to Perform Data Transfer,” *Big Data Res.*, vol. 16, pp. 1–17, 2019, doi: https://doi.org/10.1016/j.bdr.2019.02.002.

[24] N. Wannalai and S. Mekruksavanich, “The Application of Intelligent Database for Modern Information Management,” in *2019 Joint International Conference on Digital Arts, Media and Technology with ECTI Northern Section Conference on Electrical, Electronics, Computer and Telecommunications Engineering (ECTI DAMT-NCON)*, 2019, pp. 105–108. doi: 10.1109/ECTI-NCON.2019.8692242.

[25] H. Liu, “Research on Feasibility Path of Technology Supervision and Technology Protection in Big Data Environment,” in *2019 International Conference on Intelligent Transportation, Big Data & Smart City (ICITBS)*, 2019, pp. 293–296. doi: 10.1109/ICITBS.2019.00077.

[26] J. Wang and J. Tang, “Research on Hotspot and Trend of Online Public Opinion Research in Big Data Environment,” in *2019 IEEE 8th Joint International Information Technology and Artificial Intelligence Conference (ITAIC)*, 2019, pp. 1022–1025. doi: 10.1109/ITAIC.2019.8785889.

[27] J. Qu, F. Liu, and H. Meng, “A method for CIR fault diagnosis based on improved tri-training in big data environment,” in *2018 IEEE 3rd International Conference on Cloud Computing and Big Data Analysis (ICCCBDA)*, IEEE, Apr. 2018, pp. 213–218. doi: 10.1109/ICCCBDA.2018.8386514.

[28] X. Zhang and K. Wu, “The Construction of Evaluation Model of Chinese Traditional Culture Multimedia Teaching Resources Allocation in Big Data Environment,” in *2018 International Conference on Intelligent Transportation, Big Data & Smart City (ICITBS)*, 2018, pp. 133–136. doi: 10.1109/ICITBS.2018.00042.

[29] T. Ito, M. Kataoka, H. Noguchi, Y. Yamato, and T. Murase, “Network Architecture with Categorizing Metadata by Locality and Lifetime for IoT Database Management,” in *2018 21st International Symposium on Wireless Personal Multimedia Communications (WPMC)*, 2018, pp. 177–181. doi: 10.1109/WPMC.2018.8713014.

[30] S. M. J. Sadegh, S. Shahidi, and S. Valaee, “An efficient database management for cloud-based indoor positioning using Wi-Fi fingerprinting,” in *2017 IEEE 28th Annual International Symposium on Personal, Indoor, and Mobile Radio Communications (PIMRC)*, 2017, pp. 1–6. doi: 10.1109/PIMRC.2017.8292265.

[31] Mamidala, J. V., Enokkaren, S. J., Attipalli, A., Bitkuri, V., Kendyala, R., & Kurma, J. (2023). Machine Learning Models Powered by Big Data for Health Insurance Expense Forecasting. *International Research Journal of Economics and Management Studies IRJEMS*, *2*(1).

[32] Nadella, V. M. (2023). Zero Trust Architecture for Telecom Operations. *International Journal of Emerging Research in Engineering and Technology*, *4*(3), 115-129.

[33] Bitkuri, V., Kendyala, R., Kurma, J., Enokkaren, S. J., & Mamidala, J. V. (2023). Forecasting Stock Price Movements With Deep Learning Models for time Series Data Analysis. *Journal of Artificial Intelligence & Cloud Computing. SRC/JAICC-531. DOI: doi. org/10.47363/JAICC/2023 (2)*, *489*, 2-9.

[34] Nadella, V. M. (2023). Anomaly Detection and Fault Prediction using ML in Telecom Operations. *International Journal of Emerging Trends in Computer Science and Information Technology*, *4*(3), 134-143.

[35] Kosaraju, P., & Nadella, V. M. (2022). Security and Privacy in IoT Ecosystems. *Universal Library of Engineering Technology*, (Issue).

[36] Singh, A. A. S. S., Mania, V., Kothamaram, R. R., Rajendran, D., Namburi, V. D. N., & Tamilmani, V. (2023). Exploration of Java-Based Big Data Frameworks: Architecture, Challenges, and Opportunities. Journal of Artificial Intelligence & Cloud Computing, 2(4), 1-8.

[37] Routhu, K. K. (2023). AI-driven succession planning in Oracle HCM Cloud: Building resilient leadership pipelines through predictive analytics. *International Journal of Science, Engineering and Technology*, *11*(5).

[38] Tamilmani, V., Namburi, V. D., Singh Singh, A. A., Maniar, V., Kothamaram, R. R., & Rajendran, D. (2023). Real-Time Identification of Phishing Websites Using Advanced Machine Learning Methods. *Available at SSRN 5837142*.

[39] Routhu, K. K. (2023). AI-driven succession planning in Oracle HCM Cloud: Building resilient leadership pipelines through predictive analytics. International Journal of Science, Engineering and Technology, 11(5). https://doi.org/10.5281/zenodo.17292018

[40] From Fragmentation to Focus: The Benefits of Centralizing Procurement. (2023). International Journal of Research and Applied Innovations, 6(6), 9820-9833. https://doi.org/10.15662/

[41] Routhu, K. K. (2023). Embedding fairness into the digital enterprise, data driven DEI strategies with Oracle HCM Analytics. *International Journal of Scientific Research in Computer Science, Engineering and Information Technology*, *9*(8), 266-274.

[42] Routhu, K. K. (2023). AI-driven skills forecasting in Oracle HCM Cloud: From static competencies to predictive workforce design. *International Journal of Science, Engineering and Technology*, *11*(1).

[43] Padur, S. K. R. (2023). AI-Augmented Enterprise ERP Modernization: Zero-Downtime Strategies for Oracle E-Business Suite R12. 2 and Beyond. *Available at SSRN 5605510*.

[44] Routhu, K. K. (2022). From Case Management to Conversational HR: Redefining Help Desks with Oracle's AI and NLP Framework. *International Journal of Science, Engineering and Technology*, *10*(6).

[45] Vattikonda, N., Gupta, A. K., Polu, A. R., Narra, B., Buddula, D. V. K. R., & Patchipulusu, H. H. S. (2022). Blockchain Technology in Supply Chain and Logistics: A Comprehensive Review of Applications, Challenges, and Innovations. *International Journal of Emerging Trends in Computer Science and Information Technology*, *3*(3), 72-80.

[46] Attipalli, A., BITKURI, V., Mamidala, J. V., Kendyala, R., & KURMA, J. (2022). Empowering Cloud Security with Artificial Intelligence: Detecting Threats Using Advanced Machine learning Technologies. *Available at SSRN 5741263*.

[47] Padur, S. K. R. (2022). Intelligent resource management: AI methods for predictive workload forecasting in cloud data centers. *J. Artif. Intell. Mach. Learn. & Data Sci*, *1*(1), 2936-2941.

[48] Nadella, V. M. (2022). Digital Twins for Predictive Network Management and System Simulation. *International Journal of AI, BigData, Computational and Management Studies*, *3*(3), 100-111.

[49] Routhu, K. K. (2022). From RFID to Geofencing: IoT-Enabled Smart Time Tracking in Oracle HCM Cloud. *International Journal of Science, Engineering and Technology*, *10*(4).

[50] Nadella, V. (2019). Extracting road traffic data through video analysis using automatic camera calibration and deep neural networks.

[51] Polam, R. M., Kamarthapu, B., Kakani, A. B., Nandiraju, S. K. K., Chundru, S. K., & Vangala, S. R. (2022). Data Security in Cloud Computing: Encryption, Zero Trust, and Homomorphic Encryption. *International Journal of Emerging Trends in Computer Science and Information Technology*, *3*(4), 31-41.

[52] Padur, S. K. R. (2022). AI augmented platform engineering, transforming developer experience through intelligent automation and self optimizing internal platforms. *International Journal of Science, Engineering and Technology*, *10*(5), 10-5281.

[53] Kosaraju, P. , & Nadella, V. M. (2021). Quality of Experience (QoE) and Network Performance Modelling for Multimedia Traffic. Journal of Artificial Intelligence and Big Data, 1(1), 1-13. https://doi.org/10.31586/jaibd.2021.1358.