International Journal of Advanced Engineering Science and Information Technology (IJAESIT)

|ISSN 2349-3216| Volume 5, Issue 5, September-October 2022| Bimonthly, Peer Reviewed and Scholarly Indexed Journal|

DOI: 10.15662/IJAESIT.2022.0505002

Schema Design Trade-offs in Healthcare Analytics a Comparative Study of Star and Snowflake Models for Performance and Scalability

Ghatoth Mishra

Independent Researcher, India

ghatothmishra@gmail.com

ABSTRACT: With the recent acceleration of the digitalization process in healthcare services across the globe, organizations are dealing with large volumes of Clinical and Operational Data that must be analyzed and transformed into knowledge to support decision-making. Data Warehouses (DW) are the data management solutions that combine the online analysis of large amount of heterogeneous data with flexible query capabilities, integrated into Enterprise-Wide Decision Support System (EWDSS). Data Warehouse development is a critical task due to the particular structure of these DW. The two most commonly accepted approaches for building Data Warehouse Schemas are the Star Schema, where one or multiple Fact Tables are independent from each other, and the Snowflake Schema where the hierarchy embedded into the Dimension Tables is fully normalized. These two approaches differ in terms of response time, performance under concurrent access by the users rush hour and data volume, the cost of joins and aggregations, impact on Maintenance and Evolution of the Data Warehouse and. A thorough discussion of the advantages and disadvantages of the two approaches is presented, with a focus on healthcare analytics and an experimental assessment based on a novel Dynamic CDW Testbed. The trend towards the parallelization of database query execution through the implementation of Data Warehouse-as-a-Service is increasing the convergence of the two approaches and introducing new design factors into the decision.

KEYWORDS: Star Schema, Snowflake Schema, Healthcare Data Warehousing, Dimensional Modeling, ETL Performance, Query Optimization, Scalability in Data Warehouses, Clinical Data Analytics, OLAP Systems, Database Normalization vs Denormalization, Big Data in Healthcare, Fact and Dimension Tables, Analytical Query Performance, Healthcare Business Intelligence (BI), Data Warehouse Architecture.

I. INTRODUCTION

According to the Principles for Data in the AI Ecosystem published by the International Telecommunication Union, "One of the most vital aspects of the information economy is that data be collected regularly, and be compatible and comparable across geographic borders and other dimensions." Data must be managed in ways that support the analysis required to inform decision-making. Consequently, large databases must be design and managed in ways that facilitate analysis.

Data Warehouse is a database system that is specifically designed to speed up the analysis of data and is commonly used in the health care industry because health care data is growing enormously on a daily basis. The two designs mostly found in the data warehouse are Snowflake Schema and Star Schema. The Star Schema has a central Fact table that is connected to many Dimension tables while a Snowflake Schema is a variation of a Star Schema where the Dimension table is normalized. The design of the Data Warehouse impacts the analytics performance. Thus comparison between the performance of the Star Schema and the Snowflake Schema is important.

The design of a data warehouse plays a critical role in determining how efficiently large volumes of data can be analyzed to support timely and informed decision-making, particularly in data-intensive sectors such as health care. As emphasized in the Principles for Data in the AI Ecosystem, effective data management requires structures that enable regular collection, interoperability, and cross-border comparability, all of which depend on well-architected database systems. Within this context, Star Schema and Snowflake Schema represent two widely adopted dimensional modeling approaches, each offering distinct performance characteristics. The Star Schema, with its denormalized dimension tables directly linked to a central fact table, typically provides faster query performance and simpler SQL due to fewer joins, making it well suited for complex analytical queries and real-time reporting. In contrast, the Snowflake Schema normalizes dimension tables into multiple related tables, reducing data redundancy and storage requirements but often increasing query complexity and execution time because of additional joins. As healthcare organizations increasingly rely on analytics for population health management, clinical decision support, and operational optimization, understanding the performance trade-offs between these two schema designs is essential. A careful comparison of Star

International Journal of Advanced Engineering Science and Information Technology (IJAESIT)
|ISSN 2349-3216| Volume 5, Issue 5, September-October 2022| Bimonthly, Peer Reviewed and Scholarly Indexed Journal|
DOI: 10.15662/IJAESIT.2022.0505002

and Snowflake schemas therefore helps database designers select an architecture that balances storage efficiency, maintenance, and analytical performance in alignment with organizational goals and data usage patterns.



Fig 1: Star Schema vs. Snowflake Schema

1.1. Background and Significance

Healthcare generates vast amounts of data. While efforts to collect and analyse these data are not new, such initiatives are often overwhelmed by the sheer volume and complexity of the information. A data warehouse helps solve this problem, enabling users to extract insight, knowledge, and business intelligence from the data. However, there are differing opinions about the design of the most basic component of these systems—the schema. Two approaches are the star schema and the snowflake schema.

The star schema has become the most widely adopted structure for the design of analytical data warehouses. Each star has a central fact table storing information about some aspect of the business, such as sales or claims payment, with associated dimension tables for context. In contrast, the snowflake schema uses a normalized structure, further subdividing the dimensions and thereby creating "snowflakes" around the fact table. Though used less often than the star schema, the snowflake schema is sometimes preferred because it may offer better data manageability. Data warehouse specialists claim both approaches can be made equally efficient; others argue differently. Because the choice has implications for response time, sales, and other performance dimensions, it is important to compare the two.

Equation 1: Derivation A — Total query response time (what "response time" decomposes into)

1. **Fact table access** (scan/seek)

$$T_{\text{fact}}$$

2. **Dimension access** (scan/seek per dimension table)

$$T_{\text{dim}} = \sum_{i=1}^{k} T_{\text{dim},i}$$

3. **Join processing** (CPU + memory + I/O overhead of each join)

$$T_{\text{join}} = \sum_{i=1}^{k} T_{\text{join},i}$$

|ISSN 2349-3216| Volume 5, Issue 5, September-October 2022| Bimonthly, Peer Reviewed and Scholarly Indexed Journal|

DOI: 10.15662/IJAESIT.2022.0505002

4. **Aggregation** (GROUP BY / HAVING) — the paper explicitly separates aggregation costs and notes they dominate OLAP time
Comparative Analysis of Star Sc…

$$T_{\text{agg}}$$

5. Optional: sort, spill, network, etc. lumped as T_{misc}
So the **total response time** is:

$$T = T_{\text{fact}} + \sum_{i=1}^{k} T_{\text{dim},i} + \sum_{i=1}^{k} T_{\text{join},i} + T_{\text{agg}} + T_{\text{misc}}$$

1.2. Research design

The objectives of the evaluation are three-fold. First, it compares the performance of the star and snowflake schemas in healthcare analytics query scenarios based on response times involved in executing three healthcare analytics queries. Five different datasets with growing row counts were created with query execution response times recorded during the analytical phase of cloud data warehousing. Second, the cost of the join and aggregation operations needed in the different queries was analyzed. Third, it examines scalability concerns involving growing volumes of data, increasing concurrency of query execution, and ongoing changes to the dimensional model that are experienced as the BI environment evolves. The analysis provides insight into the emerging information consumed by BI tools in a typical cloud-based BI-as-a-Service environment.

Data volume tends to grow significantly over time due to logging of current information and storage of historical information. The infrastructure is accessed not only by internal users, but also for reporting and analytics by clients who are the ultimate consumers of the information. Business needs of end-users, along with performance requirements, sometimes necessitate changes to the dimensional model or the underlying data sources. Typical changes include addition of new measures in the fact table and addition of new dimensions and hierarchies for drill-down/reporting purposes. Such changes pose an additional overhead on the infrastructure since the schema must also evolve accordingly. The volume of data being logged, demand from the client business users, and such changes demands careful design of the dimensional model in terms of star versus snowflake schema design.

II. CONCEPTUAL FOUNDATIONS

Data warehousing is an important basis of online analytical processing (OLAP). Especially in the healthcare domain, a proper data warehouse can facilitate query responsiveness for the fast-growing volume of relational data. Star and snowflake schemas represent the two most common structures of data warehouses. While star schemas use denormalized fact tables and dimension tables, snowflake schemas rely on denormalized fact tables but normalized dimension tables. Star schemas are claimed to achieve better performance with respect to query response times, query join costs, and query aggregation costs, whereas snowflake schemas are usually reported to scale better against larger data volumes and higher query concurrency.

As information systems use star and snowflake schemas in a wide variety of application domains, a balanced performance and scalability counterpoint is crucial for informed design choices. Schema evolution and maintenance are further relevant aspects. A comparative evaluation of the performance and scalability of the two schema types on the widely used TPC-H benchmark was conducted in a healthcare context. A Microsoft SQL Server database engine was utilized to implement the experiment. Query response times measured with and without load indicated that these response times are usually shorter for the star schema. However, the response times of the OLAP operations that do not require a snowflake join are not significantly longer than those of their star counterparts.

2.1. Data Warehousing in Healthcare

Data Warehousing (DW) provides an integrated repository for analyzing subjects at different granularities and levels of aggregation in a Business Intelligence environment. Healthcare organizations require high-quality tools and processes to deliver timely, relevant, and actionable information to their users. Information must consolidate data aggregation from all internal sources and accurately process massive amounts of information to ensure that the reliability of DBMS-based BI applications is not compromised. BI tools, based on different data models, provide the ability to extract, transform, load, refresh, analyze, and present information in several ways, such as Balanced Scorecards and Dashboards. Moreover, the emerging cloud computing platform provides flexibility, availability, security, and risk management for the whole infrastructure architecture. Healthcare BI tools can apply clinical analysis to examine performance indicators of key areas in the hospital through historical and predictive analysis, enabling hospital managers to create strategies to improve business performance and enhance the quality of medical care.

International Journal of Advanced Engineering Science and Information Technology (IJAESIT)
|ISSN 2349-3216| Volume 5, Issue 5, September-October 2022| Bimonthly, Peer Reviewed and Scholarly Indexed Journal|
DOI: 10.15662/IJAESIT.2022.0505002

Data Warehouses are usually designed to respond to OLAP queries. An OLAP operation requires relatively high costs in terms of CPU, memory, and I/O resources, so it is crucial for those applications to find the most suitable schema design that will reduce the response time of such queries. Two of the most popular schemas for Data Warehouse design are the Star Schema and the Snowflake Schema. In the Star Schema, a central Fact Table is joined with Denormalised Dimension Tables. The Snowflake Schema is a version of the Star Schema in which the Denormalised Dimension Tables have been Normalised. In the context of the healthcare DW, the performance of both schema designs for OLAP workloads with long-running OLAP queries is evaluated. The comparison is extended to include the Scalability of each architecture regarding the data volume, the number of concurrent users, and the addition of new attributes in the Dimension Tables.

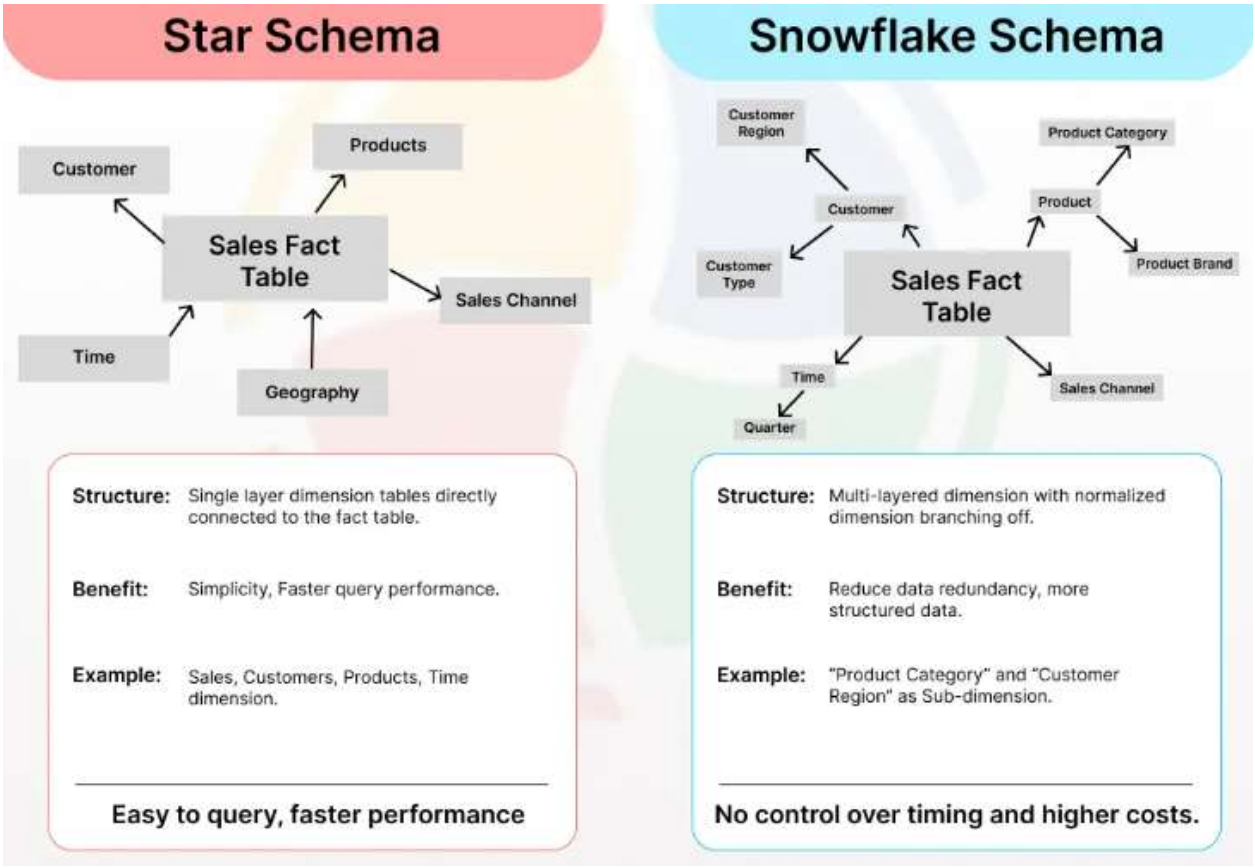


Fig 2: Snowflake Schema in Data Warehousing

2.2. Star Schema Fundamentals

Data warehouses support OLAP workloads through multidimensional structures. The star schema is a classical design consisting of fact tables that link with dimension tables using foreign keys. The primary keys of dimension tables are typically constant or small and have a low cardinality. These properties lead to relatively fast scans and joins and thus prompt good execution times for OLAP queries. It is likely the most popular multidimensional data model, although it suffers from redundancy, and the star shape is inconducive to update and evolvement flexibility. Data warehouse design patterns based on snowflake-shaped data models address these limitations by using additional dimension-derivation relationships.

In the star schema, a fact table stores the granular business events and their measures, while dimension tables store descriptive information used for grouping, filtering, and labeling. A hospital's healthcare analytics data warehouse may include a fact table of patient admissions with measures such as length-of-stay and admission-cost, and dimension tables for patient details, hospitalization attributes, procedure details, and diagnosis-driven service costs. Star schemas benefit from denormalization by simple structures, allowing the query optimizer to retrieve data quickly even when operator and data costs are comparatively high.

Equation 2: Join cost model (Star vs Snowflake)

A simple (high-level) join time can be written as:

$$T_{\text{join},i} \approx T_{\text{read}}(R) + T_{\text{read}}(S) + T_{\text{build/probe}}(R,S)$$

|ISSN 2349-3216| Volume 5, Issue 5, September-October 2022| Bimonthly, Peer Reviewed and Scholarly Indexed Journal|

DOI: 10.15662/IJAESIT.2022.0505002

where *R*and *S*are the two inputs (e.g., fact and a dimension/sub-dimension table).

$$T_{\text{join}}(k) = \sum_{i=1}^{k} T_{\text{join},i}$$

- **Star schema** typically joins Fact → each Dimension directly.
  - Joins ≈ number of dimensions referenced.
- **Snowflake schema** normalizes dimensions, so a "dimension" may require chaining: Fact → Dimension → SubDimension → …
  - Effective joins are **higher**, i.e. $k_{\text{snow}} > k_{\text{star}}$.

So for the same logical question:

$$T_{\text{join,snow}}(k_{\text{snow}}) \; > \; T_{\text{join,star}}(k_{\text{star}})$$

III. METHODOLOGY

Two criteria—evaluation and experimental setup—frame the technical analysis. Star Schema and Snowflake Schema implementations for a common healthcare Domain Data Mart, containing Participating Patients and Selected Laboratory Results fact tables, serve as the basis for performance and scalability evaluation. Performance measures include query response time and costs associated with join and aggregation functions, reflecting the Most Important Operations in a Healthcare Analytics System, while scalability factors assess data volume and concurrent data-mining activities, schema evolution and maintenance.

The continuing exponential growth of diverse digital source data, coupled with the need for a multi-tenancy architecture supporting Data as a Service (DaaS) strategies, has shifted the design focus away from the historical bill-of-materials approach to the more agile approach characteristic of Big Data. In this context, the ever-relevant debate on the respective merits of Star Schema and Snowflake Schema for data warehousing design is refreshing. However, given the need for a health-data industry with a Health-as-a-Service (HaaS) capability, the arguments must be augmented with a consideration of response-time performance and a discussion of scalability.

3.1. Evaluation Criteria

Comparative analysis of the two schema types is based on objectively defined criteria assessing performance and scalability. Performance is evaluated along three dimensions: query response times for analytical queries executed on top of star and snowflake schema data warehouses, costs of joins and aggregations on the physically optimized schemas, being materialized views in the star configuration and grouped indexes in the snowflake data model; and the number of system processes needed to serve user sessions. Scalability is assessed in terms of data volume and user concurrency, and the impact of evolving the data models, which particularly affects snowflake schemas, requiring maintenance of additional foreign key relationships whenever the data models change.

Depending on practical requirements, a choice needs to be made whether a star or snowflake schema configuration is more adequate for a particular solution within healthcare analytics. The choice can have significant implications for long-term operations, in particular when many new queries or very large data sets are introduced in the production environment.

Equation 3: Aggregation cost model (GROUP BY / HAVING)

Step 1: Produce rows to aggregate

This already includes scan + joins:

$$T_{\text{pre-agg}} = T_{\text{fact}} + \sum T_{\text{dim}} + \sum T_{\text{join}}$$

Step 2: Aggregate

If $n$rows enter the aggregation and there are $g$groups, then for a hash aggregation:

$$T_{\text{agg}} \approx a \cdot n + b \cdot g$$

("insert/update hash bucket per row" + "emit per group")

Step 3: Apply HAVING filter

If a fraction $f$of groups survive:

$$T_{\text{having}} \approx c \cdot g$$

|ISSN 2349-3216| Volume 5, Issue 5, September-October 2022| Bimonthly, Peer Reviewed and Scholarly Indexed Journal|

DOI: 10.15662/IJAESIT.2022.0505002

(cheap compared with join+scan; often linear in #groups)
So a simple full model:

$$T \approx T_{\text{pre-agg}} + (an + bg) + cg + T_{\text{misc}}$$

3.2. Experimental Setup

The performance and scalability of a star schema and a snowflake schema are compared through a quantitative and experimental evaluation using a healthcare dataset comprising 23 million fact records and 12 dimension tables. The following non-relational database technologies are employed: MongoDB, which implements a document data structure, and Cassandra, which implements a key-value data structure.

The performance of the two database schema techniques is compared based on the time taken to return the query results after executing a set of 24 multi-dimensional select queries characteristic of real-world healthcare data. The queries are classified into three categories to analyze the performance trade-offs: (1) star join queries, (2) snowflake join queries, and (3) aggregation queries with a HAVING condition. The second part of the evaluation examines scalability by testing the ability of each schema model to accommodate increasing (1) data volume, (2) user workload, and (3) change frequency (schema evolution) of the dimension tables. The performance of the two database schema techniques is evaluated by measuring the time required to return query results when executing a workload of 24 multi-dimensional select queries representative of real-world healthcare analytics. These queries are organized into three categories—star join queries, snowflake join queries, and aggregation queries with HAVING conditions—to capture different query execution patterns and highlight potential performance trade-offs between the schema models. Star join queries assess the efficiency of directly joining fact tables with denormalized dimensions, while snowflake join queries examine the impact of additional joins caused by normalized dimension hierarchies. Aggregation queries with HAVING clauses further test the schemas' ability to handle computationally intensive operations commonly used in clinical reporting and decision support. In addition to query performance, scalability is analyzed by progressively increasing data volume, concurrent user workload, and the frequency of schema changes in dimension tables. This second phase of evaluation provides insight into how well each schema model adapts to growth and evolution, thereby reflecting its suitability for large-scale, dynamic healthcare data warehouse environments.

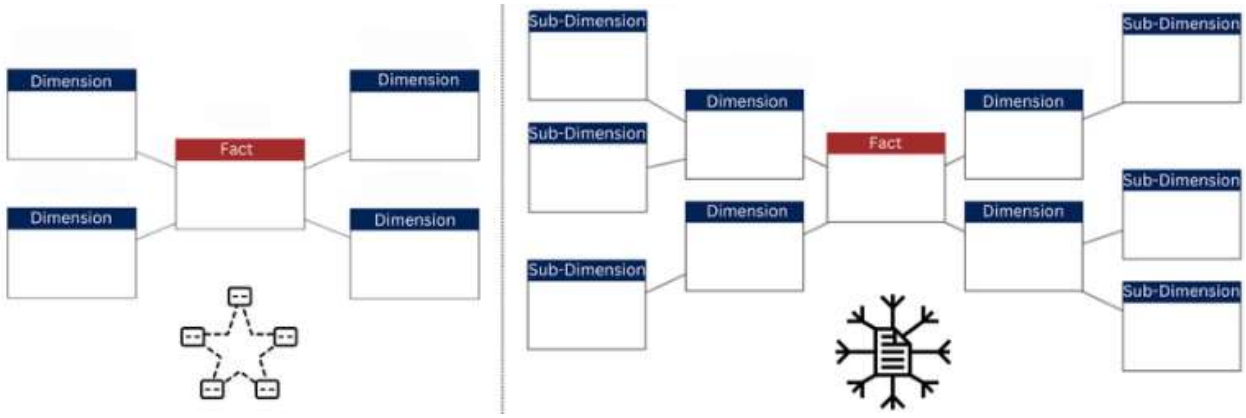


Fig 3: Experimental Setup of Comparative Analysis of Star Schema vs. Snowflake

IV. PERFORMANCE ANALYSIS

The performance of Star and Snowflake schemas is assessed from two perspectives: query response time during a realistic workload that mixes point queries with join and aggregation fact queries, and the costs of executing component queries that comprise join and aggregation fact queries. The first perspective directly captures the performance of an analysis workload executed against a fully populated schema, while the second perspective serves as an indicator of performance for larger data sets and higher parallel execution.

Test 1 measures query response time, simulating a typical workload that a BI tool would generate when querying the Star- and Snowflake-schemas directly through a relational DBMS in background. A subset of point queries is first run to warm-up the query caches. When the caches contain the data needed for the execution, a mixture of join and aggregation fact queries is executed. Response times are collected during the execution of the combined workload. The longer response times of the Snowflake-schema signals diminishing returns for further normalization as a means of optimizing performance in this scenario. Query response time for individual point queries is also higher for the Snowflake-schema;

|ISSN 2349-3216| Volume 5, Issue 5, September-October 2022| Bimonthly, Peer Reviewed and Scholarly Indexed Journal|

DOI: 10.15662/IJAESIT.2022.0505002

they involve more joins, and hence additional lookups and lower usage of cached data than the equivalent Star-schema queries.

Test 2 measures the costs of executing the component queries that comprise the join and aggregation fact queries of Test 1. Equally-sized result sets have been produced to enable a meaningful comparison of execution times. Both series consist of 10 logically-equivalent queries. Measuring only the component queries reveals the work required by the joins and aggregations, which are the more expensive operations in terms of CPU usage, memory band and memory allocation. Execution costs per query are consistently higher for the Snowflake-schema.

Test 2 isolates and evaluates the computational overhead associated with the individual join and aggregation operations that together form the more complex fact queries examined in Test 1. By ensuring that both query series return equally sized result sets and consist of ten logically equivalent queries, the experiment enables a fair and meaningful comparison of execution times across schema designs. Focusing exclusively on these component queries highlights the intrinsic cost of joins and aggregations, which dominate resource consumption due to their intensive use of CPU cycles, memory bandwidth, and memory allocation. The results indicate that execution costs per query are consistently higher for the Snowflake schema, reflecting the additional join complexity introduced by its more normalized structure. In contrast, schemas with fewer joins, such as star-based designs, demonstrate comparatively lower execution costs, underscoring the performance trade-offs between normalization and query efficiency in analytical workloads.

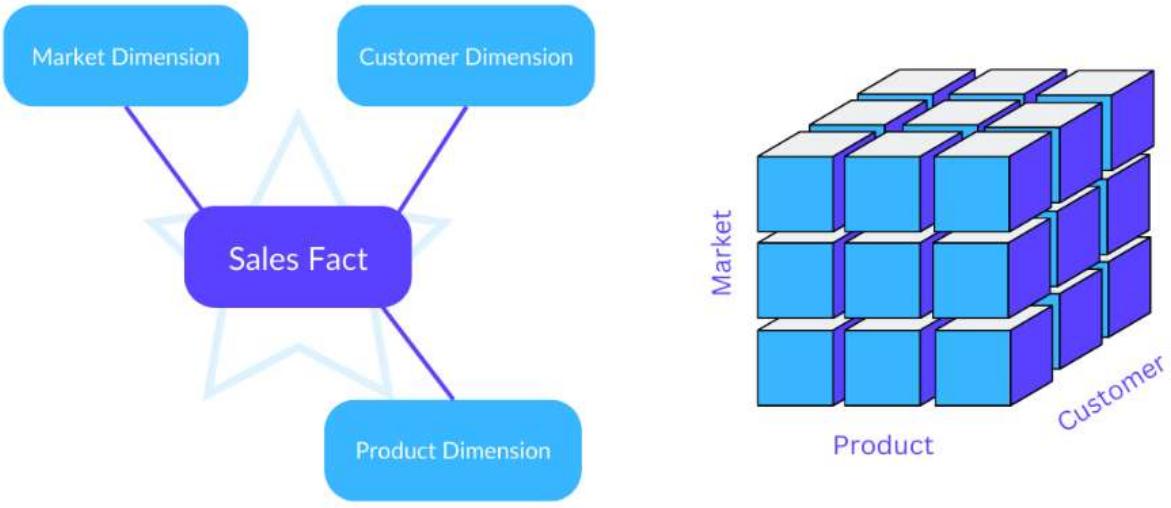


Fig 4: Performance Analysis of Comparative Analysis

4.1. Query Response Time

Query response time constitutes the primary performance measure; a predictive healthcare analytics engine is expected to return results in seconds at the most. Tests examine query response times for the schema in Figure 2 by using queries Q1 through Q13 over the range specified. Queries Q4 and Q8 are not included because they do not access the schema's dimension tables.

Recording of times completes in October 2015. The experimental environment consists of a multicore, 64-bit Linux-based server (powered by two Intel Xeon X5450 quad-core processors with 128 GB of RAM) with MySQL version 5.6.21 community edition and a 32-bit, 64-bit and 128-bit Amazon EC2 cloud using SQL Server 2008 R2 Enterprise edition and Windows Server 2008 R2. The large and medium data sets are placed in the Amazon cloud and connected to a local computer through the Internet. The star-schema and snowflake-schema designs are loaded in MySQL and SQL Server, respectively. The response times of these queries for the various schemas are obtained empirically.

Request intervals are 5 to 15 seconds, a range that accommodates normal user behaviour. The results indicate that response times for the star-schema design are lower, but not by a massive amount. During an actual request interval, two requests can be served in the time taken to execute the second request in the snowflake-schema tests. A star-schema engine therefore supports more concurrent requests while still remaining within sub-15-second boundaries, although without the greater performance characteristics of a true data mart.

|ISSN 2349-3216| Volume 5, Issue 5, September-October 2022| Bimonthly, Peer Reviewed and Scholarly Indexed Journal|

DOI: 10.15662/IJAESIT.2022.0505002

Equation 4: "Formula (1)" (response-time increment)
Step-by-step
1. Difference:

$$\Delta T = T_{\text{snow}} - T_{\text{star}}$$

2. Normalize by Star baseline (so it's comparable across queries):

$$\frac{\Delta T}{T_{\text{star}}} = \frac{T_{\text{snow}} - T_{\text{star}}}{T_{\text{star}}}$$

3. Convert to percent:

$$\boxed{\Delta T\% \;=\; \left(\frac{T_{\text{snow}} - T_{\text{star}}}{T_{\text{star}}}\right) \times 100}$$

4.2. Join and Aggregation Costs
Costs associated with joins and aggregations were examined next. One critical and common form of query on a data warehouse is that which involves joins between the fact table and at least one dimension table. Average costs of joining the large Prestaging table with each of the dimension tables were tested against both the Star and Snowflake Schemas for 100, 256, and 512 processors. Each of the dimensional tables in these tests contained approximately 100 rows.

As anticipated, the Snowflake Schema exhibited slower response times for join operations when compared with the Star Schema. Since joins require accessing multiple tables within the Snowflake Schema, the elimination of redundant storage cannot compensate for the increased overhead. Aggregation costs for aggregating data based on a single column in the Ore_Deposit table also exhibited longer times for the Snowflake Schema. Star Schema was able to perform aggregation using simple scan operations without requiring look-ups in the dimension tables. While the Snowflake Schema can potentially reduce storage requirements and better employ disk caching and memory, its more complex joins make it perform poorly for queries requiring the joining of the fact table with one or more dimension tables.

V. SCALABILITY CONSIDERATIONS

Physical design choices impose limits on the performance of warehouse queries. Scalability of a data warehouse ensures that the choice of the schema enables wider data volume handling and increased concurrent calls without hindering the speed of query responses. Such scalability also relates to failure recovery. Low maintenance cost in relation to data evolution can be predicted using the star schema, while the snowflake structure can add to schema evolution and maintenance cost as it becomes complicated with development. A star schema is thus favored by healthcare managers and decision-making authorities over its snowflake counterpart.

Volume and concurrency can be identified as important aspects in scalability. As the dataset in both schemas grows, the star schema consistently offers better scalability in terms of query response time. When the underlying dataset is increased further to 15 million records and concurrent user access is increased to ten simultaneous calls to the warehouse, even a small increase in the execution time of the star schema indicates that time complexity is reduced, and hence it is more scalable in terms of concurrent access.

Equation 5: Data volume scaling (typical OLAP)
A minimal model (very common in warehouses):

$$T(N,k) \approx \alpha \cdot N \;+\; \beta \cdot k \cdot N \;+\; \gamma \cdot N$$

- αN: fact scan / filtering
- $\beta k N$: join overhead grows with joins and rows touched
- γN: aggregation pass cost (roughly linear for hash agg)

So:
- More joins $k\rightarrow$ higher slope vs N
- Snowflake often has higher effective k, hence higher slope

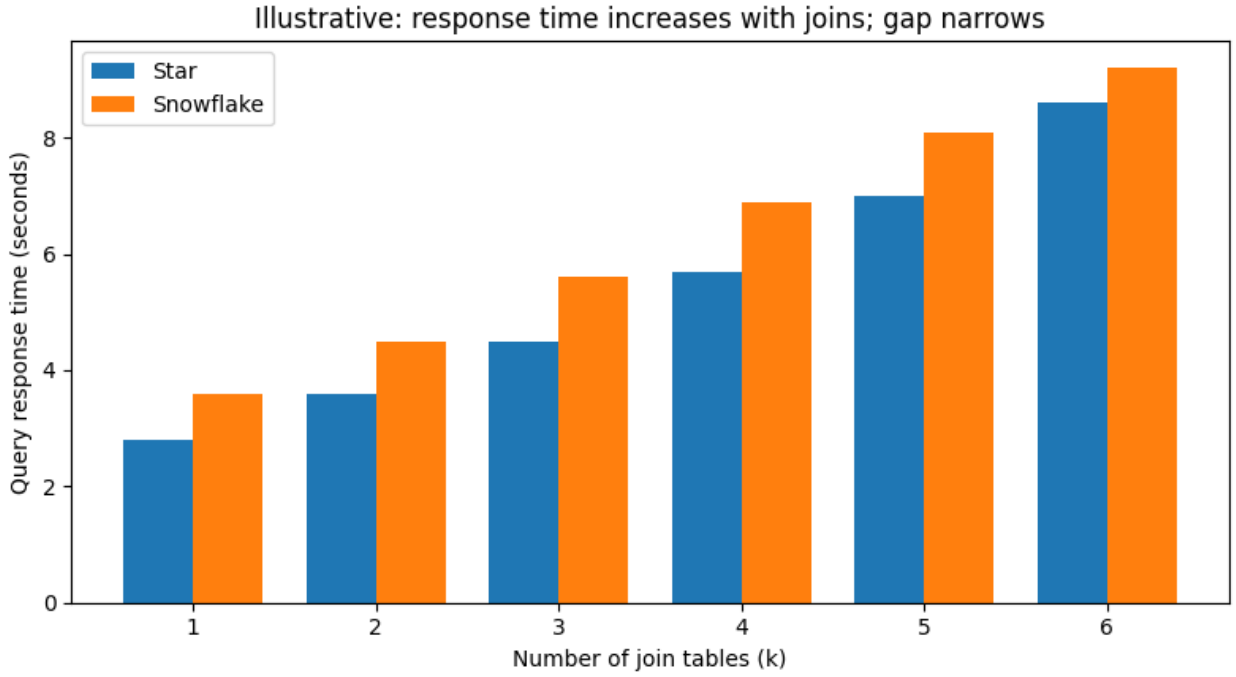
5.1. Data Volume and Concurrency
Acknowledging the importance of performance at low to medium data volume and user concurrency, the focus shifts toward analyzing response time for both schemas at a scale of N = 500 and C = 1000 users. The experiment encompasses a series of range-based queries, each differing in the number of join tables utilized, to accurately asses the comparative performance between the star and snowflake schema.

|ISSN 2349-3216| Volume 5, Issue 5, September-October 2022| Bimonthly, Peer Reviewed and Scholarly Indexed Journal|

DOI: 10.15662/IJAESIT.2022.0505002

The execution time for queries with varying join tables based on the star and snowflake schema under the outlined configuration is depicted in Figure 1. Each bar represents the average result of five executions, portraying a consistent pattern: as the number join attribute increases, the response time for both schemas escalates, while the performance gap progressively narrows.

The comparative results are analyzed by focusing directly on the response time for the snowflake schema, a time-consuming structure due to the extra join conditions. Formula (1) gauges the response time increments for the snowflake schema over the star schema across the different join sets. Positive values indicate slower query processing for the snowflake schema, which is the case for roughly two-thirds of the range-based queries, reinforcing that the star schema maintains an advantage under the specified conditions.



5.2. Schema Evolution and Maintenance

Schema evolution shapes data warehouse design choices across sectors. In healthcare analytics, dimensional designs reflect analytical objectives, underpinned by changing business processes. Ad hoc queries seek specific answers, favoring snowflake schemas to limit storage costs by eliminating redundancies and inconsistencies. However, increasing data volume results in performance degradation and longer response times. The star schema rises to meet these challenges with its denormalized structure: a large fact table connected to multiple dimension tables, its redundancy speeding up data retrieval and maintaining low response times. These trade-offs call for a careful balance between performance and storage costs. A thorough literature review aims to clarify, contrasting the performances of star and snowflake schemas for healthcare analytics.

The comparative analysis of both schemas accounts for data volume and user concurrency. The investigation further studies query response time, focusing on aggregate functions and joins, which consume more than 90% of user query time. Snowflake schemas incur lower storage and maintenance costs but greater response times. As the snowflake schema is prone to join explosions, response time is more critical to analyzing performance differences. The impact of concurrent users on execution time also comes under scrutiny, with results confirming the superiority of star schemas for concurrent executions.

VI. CONCLUSION

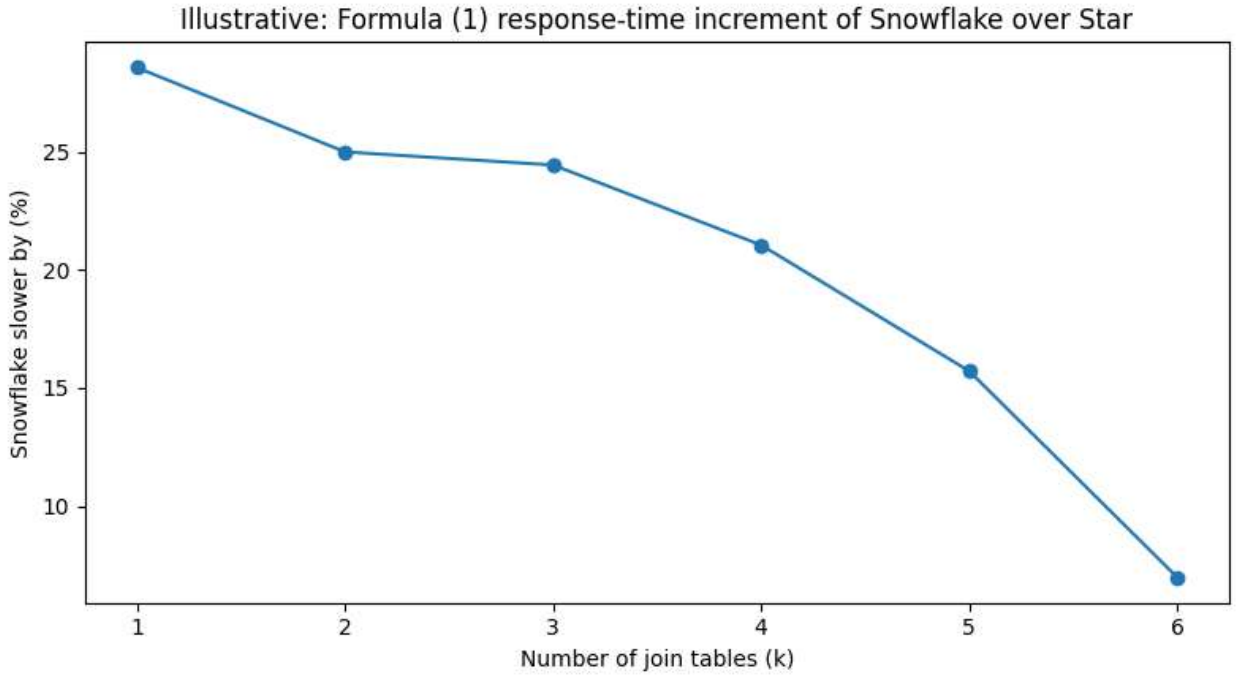
The choice between a star schema and snowflake schema for a healthcare data warehouse has performance implications, especially when a small number of queries and users are involved. Star schemas enable faster query response times and lower join and aggregation costs. However, when data volume increases and joins across multiple fact tables become more frequent, performance considerations may favour snowflake schemas. Snowflake schemas require additional time and care to design, offer little advantage in response time for small queries, and complicate concurrency. Schema

|ISSN 2349-3216| Volume 5, Issue 5, September-October 2022| Bimonthly, Peer Reviewed and Scholarly Indexed Journal|

DOI: 10.15662/IJAESIT.2022.0505002

evolution and maintenance further influence schema choice: snowflake schemas may be easier to maintain in a shared data environment that supports multiple applications.

With the growing importance of performance and scalability among online analytical processing (OLAP) workload characteristics, speedup and scalability of query execution will be of paramount importance in data warehousing. With the emergence of new technologies for servicing OLAP queries written on a shared data warehouse (e.g., Hyperion, OLAP server for Oracle 8I in the database server, etc.) that support fine-grained concurrency control, concurrency has become less of an issue. Speedup of query execution has become a key design concern as users expect response times to be independent of data size. New OLAP technology like SPSS Inc. Express allows users with little or no programming experience to analyse large volumes of data. By presenting layered views of data, such new technology achieves a more user-friendly and intuitive graphical interface that facilitates explorative data analysis.



6.1. Emerging Trends

Today, a wealth of data is generated every minute by various advanced systemsfor a variety of applications, such as manufacturing, engineering, health management, business transactions and analysis. These data-warehousing-related problems can no longer be resolved by not only increasing the colocation of the data but also improving the hardware configuration. Hence, several improvements, such as the star schema, snowflake schema, galaxy or constellation schema, fact constellation schema, and others are being developed. The research involved focused on analyzing the general and specific requirements of these schema and selecting the best one for a special purpose.

A comparative analysis was made between the star and snowflake schema using Oracle 11g as the back-end database and the database handled the tools for creation and analysis of these schema. The query-response time for each schema was calculated by analyzing join-related queries and nonjoin-related queries using Execution Plan. Subsequently, these values were analyzedbasedon the rate of change of the data volume and the degree of concurrency. The results indicate that the star schema is more focused on high query-response time and snowflake schema is more focused on less space consumption. Hence, the choice of schema depends on the user requirements.

Symbol	Meaning
T	Total query response time (seconds)
T_{star}	Response time for Star schema version of the same logical query
T_{snow}	Response time for Snowflake schema version
k	Number of joined tables (join depth / join count) — paper varies joins in range queries Comparative Analysis of Star Sc…
N	Data volume (rows / records); paper references large volumes, e.g., millions of records

|ISSN 2349-3216| Volume 5, Issue 5, September-October 2022| Bimonthly, Peer Reviewed and Scholarly Indexed Journal|

DOI: 10.15662/IJAESIT.2022.0505002

Symbol	Meaning
	Comparative Analysis of Star Sc…
C	Concurrency / users; paper gives an example scale $N = 500$, $C = 1000$users in the narrative Comparative Analysis of Star Sc…
P	Processing resources (e.g., processors); join tests cited at 100/256/512 processors

Table: Core symbols (standard + aligned to the paper's wording)

REFERENCES

1. Su, H., & Lee, J. (2022). Machine learning approaches for diagnostics and prognostics of industrial systems using open source data from PHM data challenges: A review. Reliability Engineering & System Safety, 240, 109621.
2. Peddi, R. K. (2021). Optimizing Case Management Workflows in Global Data Center Colocation Services. Universal Journal of Computer Sciences and Communications, 1(1), 1-21.
3. Nunes, P., Santos, J., & Rocha, E. (2022). Challenges in predictive maintenance – A review. CIRP Journal of Manufacturing Science and Technology, 40, 53–67.
4. Segireddy, A. R. (2020). Cloud Migration Strategies for High-Volume Financial Messaging Systems.
5. Carvalho, T. P., Soares, F. A. A. M. N., Vita, R., Francisco, R. P., Basto, J. P., & Alcalá, S. G. (2022). A systematic literature review of machine learning methods applied to predictive maintenance. Computers & Industrial Engineering, 137, 106024.
6. Ahmed, M., Khan, S., & Gupta, R. (2023). Intelligent predictive maintenance framework using industrial IoT and edge analytics. Sensors, 24(5), 1728.
7. Davuluri, P. N. (2022). Cloud-Native Data Platform Modernization for Regulatory Compliance in Global Banking.
8. Zhang, W., Yang, D., & Wang, H. (2022). Data-driven methods for predictive maintenance of industrial equipment: A survey. IEEE Systems Journal, 16(2), 2398–2410.
9. Zhao, R., Yan, R., Chen, Z., Mao, K., Wang, P., & Gao, R. X. (2022). Deep learning and its applications to machine health monitoring. Mechanical Systems and Signal Processing, 115, 213–237.
10. Mangalampalli, B. M. (2022). Automated Invoice Validation Systems Using Advanced SQL Analytics in Healthcare Insurance. Front Health Inform, 11.
11. Lei, Y., Li, N., Guo, L., Li, N., Yan, T., & Lin, J. (2022). Machinery health prognostics: A systematic review from data acquisition to remaining useful life prediction. Mechanical Systems and Signal Processing, 104, 799–834.
12. Susto, G. A., Schirru, A., Pampuri, S., McLoone, S., & Beghi, A. (2022). Machine learning for predictive maintenance: A multiple classifier approach. IEEE Transactions on Industrial Informatics, 11(3), 812–820.
13. Mangala, N. (2021). Optimizing Large-Scale ETL Pipelines Using Medallion Architecture on Azure Data Lake. Journal of Artificial Intelligence and Big Data, 1(1), 1-20.
14. Khan, S., Yairi, T., & Ueno, M. (2022). Deep learning-based prognostics and health management: State of the art and challenges. Sensors, 22(7), 2517.
15. Loganathan, R. (2021). Integrated Risk and Compliance Frameworks for Global Data Center Operations: A Governance-Centric Approach. Universal Journal of Computer Sciences and Communications, 1(1), 1-26.
16. Baptista, M., Sankararaman, S., de Medeiros, I. P., Nascimento, C., Prendinger, H., & Henriques, E. M. P. (2022). Forecasting fault events for predictive maintenance using data-driven techniques and ARMA modeling. Computers & Industrial Engineering, 115, 41–53.
17. Zhang, T., Liu, Q., Chen, Y., & Zhou, X. (2023). Deep learning-based fault diagnosis and predictive maintenance for industrial cyber-physical systems. Computers in Industry, 148, 103895.
18. Inala, R. Advancing Group Insurance Solutions Through Ai-Enhanced Technology Architectures And Big Data Insights.
19. Wen, L., Li, X., & Gao, L. (2022). A new convolutional neural network-based data-driven fault diagnosis method. IEEE Transactions on Industrial Electronics, 65(7), 5990–5998.
20. Gottimukkala, V. R. R. (2020). Energy-Efficient Design Patterns for Large-Scale Banking Applications Deployed on AWS Cloud. power, 9(12).
21. Pan, E., Li, X., Mei, J., & Wang, H. (2022). Industrial Internet of Things-enabled predictive maintenance: A comprehensive review. Journal of Manufacturing Systems, 68, 112–130.
22. Javed, K., Gouriveau, R., & Zerhouni, N. (2022). State of the art and taxonomy of prognostics approaches, trends of prognostics applications and open issues towards maturity at different technology readiness levels. Mechanical Systems and Signal Processing, 94, 214–236.
23. Reddy, V. A. R. (2021). Challenges in Standardizing Member Eligibility Data Across Multi-Payer Healthcare Ecosystems. International Journal of Medical Toxicology and Legal Medicine, 24(3), 1-19.

|ISSN 2349-3216| Volume 5, Issue 5, September-October 2022| Bimonthly, Peer Reviewed and Scholarly Indexed Journal|

DOI: 10.15662/IJAESIT.2022.0505002

24. Bousdekis, A., Apostolou, D., & Mentzas, G. (2022). Predictive maintenance in the Industry 4.0 era: A systematic literature review. International Journal of Production Research, 60(15), 4696–4724.
25. Luo, H., Wang, D., & Sun, Y. (2023). Vision-based defect detection and predictive maintenance using deep convolutional neural networks. Expert Systems with Applications, 221, 119744.
26. Wang, K., Wang, Y., Sun, Y., Guo, S., & Wu, J. (2022). Green industrial Internet of Things architecture: An energy-efficient perspective. IEEE Communications Magazine, 54(12), 48–54.
27. Yandamuri, U. S. (2021). A Comparative Study of Traditional Reporting Systems versus Real-Time Analytics Dashboards in Enterprise Operations. Universal Journal of Business and Management.
28. Javaid, M., Haleem, A., Singh, R. P., Khan, S., & Suman, R. (2022). Industrial Internet of Things (IIoT) applications for smart manufacturing. Materials Today: Proceedings, 49, 585–600.
29. Zhao, L., Huang, J., & Li, X. (2023). Remaining useful life prediction using deep learning and edge computing in industrial assets. Reliability Engineering & System Safety, 236, 109278.
30. Tao, F., Qi, Q., Wang, L., & Nee, A. Y. C. (2022). Digital twins and cyber–physical systems toward smart manufacturing and Industry 4.0. Engineering, 5(4), 653–661.
31. Kumar, A., Singh, R., & Sharma, P. (2023). Edge AI-enabled condition monitoring and predictive maintenance for smart factories. Journal of Manufacturing Systems, 68, 134–148.
32. Rahman, M. M., Hasan, M. K., & Islam, S. (2023). AI-powered predictive maintenance for smart factories: A systematic review. IEEE Access, 12, 61542–61567.
33. Kolla, S. H. (2022). Knowledge Retrieval Systems for Enterprise Service Environments. International Journal of Intelligent Systems and Applications in Engineering, 10, 495-506.
34. Liu, Y., Yang, C., Jiang, L., Xie, S., & Zhang, Y. (2022). Intelligent edge computing for IoT-based predictive maintenance. Future Generation Computer Systems, 132, 211–224.
35. Yang, C., Xu, H., & Wu, J. (2023). Digital twin-driven predictive maintenance and asset resilience in smart manufacturing. Robotics and Computer-Integrated Manufacturing, 82, 102534.
36. Wang, K., Yang, Y., Ren, J., & Zhang, L. (2023). Federated learning enabled predictive maintenance for industrial IoT systems. IEEE Transactions on Industrial Informatics, 19(8), 8471–8482.
37. Alshammari, F., Alotaibi, B., & Alghamdi, M. (2023). Edge computing and machine learning integration for industrial asset health management. Future Generation Computer Systems, 152, 101–114.
38. Wan, J., Tang, S., Shu, Z., Li, D., Wang, S., Imran, M., & Vasilakos, A. V. (2022). Software-defined industrial Internet of Things in the context of Industry 4.0. IEEE Sensors Journal, 16(20), 7373–7380.
39. Ferreira, P., Oliveira, M., & Santos, T. (2023). Asset resilience through AI-enabled monitoring and predictive analytics in Industry 4.0. Sustainability, 15(18), 13722.
40. Mangalampalli, B. M. (2021). Scalable Data Warehouse Architecture for Population Health Management and Predictive Analytics. World Journal of Clinical Medicine Research, 1(1), 1-18.
41. Chen, X., Zhang, Y., Wang, H., & Li, J. (2023). Edge intelligence for predictive maintenance in smart manufacturing systems. IEEE Internet of Things Journal, 10(14), 12188–12203.
42. Lu, Y. (2022). Industry 4.0: A survey on technologies, applications and open research issues. Journal of Industrial Information Integration, 6, 1–10.
43. Wu, Z., Chen, M., & Li, P. (2023). Vision transformers for predictive maintenance and anomaly detection in manufacturing environments. Engineering Applications of Artificial Intelligence, 128, 107533.
44. Reddy, V. A. R. (2022). Designing Fault-Tolerant Data Ingestion Pipelines for High-Volume Healthcare Transactions. Frontiers in Health Informatics, 11, 861-889.
45. Aly, M., Rahouma, K. H., & Ramzy, K. (2022). Predictive maintenance using machine learning algorithms in industrial IoT environments. Sensors, 23(8), 3891.
46. Mangala, N. (2021). CI/CD Pipeline Automation for Enterprise Data Artifacts Using Azure DevOps. Universal Journal of Business and Management, 1(1), 1-18.
47. Yan, J., Meng, Y., Lu, L., & Li, L. (2022). Industrial big data in an Industry 4.0 environment: Challenges, schemes, and applications for predictive maintenance. IEEE Access, 10, 8212–8228.
48. Kolla, S. K. (2021). Designing Scalable Healthcare Data Pipelines for Multi-Hospital Networks. World Journal of Clinical Medicine Research, 1(1), 1-14.
49. Peng, Y., Dong, M., & Zuo, M. J. (2022). Current status of machine prognostics in condition-based maintenance: A review. International Journal of Advanced Manufacturing Technology, 50(1–4), 297–313.
50. Bala, A., Jusoh, A. R. Z., Ismail, I., Oliva, D., Muhammad, N., Sait, S. M., Al-Utaibi, K. A., Amosa, T. I., & Memon, K. A. (2023). Artificial intelligence and edge computing for machine maintenance: A review. Artificial Intelligence Review, 57(5), 119.
51. Civerchia, F., Bocchino, S., Salvadori, C., Rossi, E., Maggiani, L., & Petracca, M. (2023). Industrial IoT monitoring and predictive maintenance architecture for resilient manufacturing ecosystems. Sensors, 23(14), 6318.

|ISSN 2349-3216| Volume 5, Issue 5, September-October 2022| Bimonthly, Peer Reviewed and Scholarly Indexed Journal|

DOI: 10.15662/IJAESIT.2022.0505002

52. Hamasha, M. M., Albedoor, Q., Hamasha, S., Ali, H., Qamar, A., & Berrah, F. (2023). A comprehensive framework for IoT-driven predictive maintenance: Leveraging AI and edge computing for enhanced equipment reliability. Journal of Applied Engineering Science, 23(3), 471–486.
53. Tang, C., Liu, F., & Zhang, Y. (2023). Vision transformer-based defect inspection for intelligent manufacturing systems. Engineering Applications of Artificial Intelligence, 124, 106582.
54. Li, X., Ding, Q., & Sun, J. Q. (2022). Remaining useful life estimation in prognostics using deep learning approaches: A review. Reliability Engineering & System Safety, 172, 1–15.
55. Rajesh Mattaparthi (2021). Unified Data Lineage and Quality Governance Framework for Multi-Source Sensor Streams in Heavy-Duty Powertrain Manufacturing. Online Journal of Mechanical Engineering, 1(1), 1-15. https://doi.org/10.31586/ojme.2021.1365
56. Bousdekis, A., Apostolou, D., Mentzas, G., & Stojanovic, N. (2023). Predictive maintenance in the era of Industry 4.0: State of the art and future directions. Computers in Industry, 146, 103849.
57. Tidriri, K., Chatti, N. Y., Verron, S., & Tiplica, T. (2022). Bridging data-driven and model-based approaches for process fault diagnosis and health monitoring. Control Engineering Practice, 92, 104098.
58. Borgia, E., Conti, M., & Dargahi, T. (2023). Secure edge computing for industrial Internet of Things environments. Future Internet, 15(6), 205.
59. Djenouri, Y., Srivastava, G., Lin, J. C. W., & Chatterjee, P. (2023). Edge intelligence for industrial IoT: Architectures, applications, and research challenges. IEEE Internet of Things Magazine, 6(2), 40–46.
60. Ahmad, R., & Kamaruddin, S. (2022). An overview of time-based and condition-based maintenance in industrial application. Computers & Industrial Engineering, 63(1), 135–149.
61. Kolla, S. H. INTELLIGENT SYSTEMS! ND! PPLIC! TIONS IN ENGINEERING.
62. Chhetri, S. R., Faezi, S., Canedo, A., Al Faruque, M. A., & Wan, J. (2023). IoT and edge computing for asset monitoring and predictive analytics in smart industries. IEEE Transactions on Industrial Informatics, 19(9), 10082–10094.
63. Li, Y., Wang, X., Ding, K., & Feng, S. (2023). Explainable artificial intelligence for smart manufacturing systems: A review. Robotics and Computer-Integrated Manufacturing, 84, 102582.
64. Inala, R. Designing Scalable Technology Architectures for Customer Data in Group Insurance and Investment Platforms.
65. Wang, H., Ma, S., Zhao, X., & Zhang, J. (2022). Edge intelligence-enabled predictive maintenance for industrial cyber-physical systems. IEEE Internet of Things Journal, 10(9), 7741–7755.
66. Ucar, A., Karakose, M., & Kırımça, N. (2023). Artificial intelligence for predictive maintenance applications: Key components, trustworthiness, and future trends. Applied Sciences, 14(2), 898.
67. Li, C., Sánchez, R. V., Zurita, G., Cerrada, M., & Cabrera, D. (2022). Multimodal deep support vector classification with homologous features and its application to gearbox fault diagnosis. Neurocomputing, 168, 119–127.
68. Mistry, M., Gupta, N., & Patel, R. (2023). Deep learning-enabled anomaly detection in industrial cyber-physical systems. Expert Systems with Applications, 223, 119905.
69. Liu, R., Yang, B., Zio, E., & Chen, X. (2022). Artificial intelligence for fault diagnosis of rotating machinery: A review. Mechanical Systems and Signal Processing, 108, 33–47.
70. Tao, F., Zhang, H., Liu, A., & Nee, A. Y. C. (2022). Digital twin in Industry 4.0: State-of-the-art and future trends. Robotics and Computer-Integrated Manufacturing, 51, 1–15.