ISSN: 2583-9888
DOI: doi.org/10.51219/JAIMLD/lakshmi-narasimha/347

URF PUBLISHERS
connect with research world

Journal of Artificial Intelligence, Machine Learning and Data Science

https://urfpublishers.com/journal/artificial-intelligence

Vol: 1 & Iss: 1 *Research Article*

AI-Driven Oracle Database Monitoring in the Utilities Industry: Unleashing the Potential of Dynatrace

Lakshmi Narasimha Rohith Samudrala*

Citation: Samudrala LNR. AI-Driven Oracle Database Monitoring in the Utilities Industry: Unleashing the Potential of Dynatrace. J Artif Intell *Mach Learn & Data Sci 2022*, 1(1), 1548-1554. DOI: doi.org/10.51219/JAIMLD/lakshmi-narasimha/347

Received: 02 September, 2022; **Accepted:** 18 September, 2022; **Published:** 20 September, 2022

***Corresponding author:** Lakshmi Narasimha Rohith Samudrala, AVCO Consulting, Inc, USA

ABSTRACT

In the utilities industry, where uninterrupted service and operational efficiency are paramount, the performance and reliability of Oracle databases are critical. This paper explores the implementation of AI-driven monitoring using Dynatrace to enhance the performance of Oracle databases within this sector. By leveraging AI, utility companies can achieve real-time anomaly detection and automated root-cause detection, ensuring their databases operate at peak performance. Through a detailed case study, this paper examines the challenges faced in database management, the solutions provided by AI-driven monitoring and the impact on overall system performance and reliability. The findings demonstrate significant improvements in query execution times, resource utilization and operational efficiency, highlighting the potential of AI-driven tools to revolutionize database monitoring in the utilities industry.

Keywords: Oracle Database, Utilities Industry, AI-Driven Monitoring, Dynatrace, Anomaly Detection, Predictive Maintenance, Database Performance, Adaptive Thresholding, Root Cause Analysis, IT Infrastructure

1. Introduction

The utilities industry, encompassing sectors such as electricity, gas and water, is critically dependent on the reliability and performance of its IT infrastructure. Central to this infrastructure are the databases, with Oracle databases playing a significant role. Oracle's database solutions are widely used across the utilities sector for managing large-scale operations, including customer information systems (CIS), billing, energy management and regulatory compliance. With a consistent market share of about 20-25% in the global DBMS market oracle's presence in the utilities industry is substantial, as it is well suited for complex operations[8,10]. Given the scale and complexity of these operations, ensuring the performance and reliability of Oracle databases is crucial for maintaining uninterrupted service delivery and operational efficiency.

Although traditional database monitoring approaches are useful, they often struggle to keep up with the dynamic and complex nature of modern utility operations. Traditional monitoring typically depends on predefined thresholds and reactive monitoring, which may not be sufficient to address the real-time demands and potential anomalies that can arise in such critical environments. As a result, there is a rising need for more advanced monitoring solutions that can proactively manage database performance, predict potential issues and ensure continuous system availability.

The use of Artificial Intelligence (AI) is pivotal to taking database monitoring to the next level. With the use of AI, advanced features such as Adaptive Thresholds, Causational Analysis, Dependency Mapping, Capacity Planning, Self-Healing, etc. can be leveraged. By utilizing AI-driven robust observability platform like Dynatrace, utility companies can achieve real-time anomaly detection, automated root cause analysis and predictive maintenance. These AI-driven capabilities enable utilities to not only monitor their Oracle databases more effectively but also to anticipate and mitigate potential problems before they impact service delivery[1].

In the utilities industry, the stakes of database performance are exceptionally high. A minor delay in processing data can lead to significant operational disruptions, such as delayed billing, energy distribution inefficiencies or even service outages during peak demand periods. Moreover, utility companies are often required to adhere to stringent regulatory standards, which necessitate robust data security and compliance monitoring. AI-driven monitoring tools, when integrated with Oracle databases, can address these challenges by providing deep insights into database performance, enhancing security measures and ensuring regulatory compliance.

This paper explores the application of AI in Oracle database monitoring within the utilities industry, with a focus on how Dynatrace can be leveraged to enhance observability and optimize performance. Through a detailed case study, we examine the challenges faced by a utility company in managing its Oracle databases and how the integration of AI-driven monitoring solutions enabled the company to overcome these challenges, ensuring continuous service delivery and operational resilience. The paper also discusses the best practices for implementing AI-driven monitoring in Oracle database environments, offering practical insights for industry practitioners and decision-makers.

2. Key Terminology

- **Dynatrace:** Dynatrace is a software intelligence platform that provides proactive monitoring for applications, infrastructure and user experience through a combination of artificial intelligence (AI) and automation. It delivers deep observability across the entire technology stack, allowing IT teams to manage the performance of complex applications with ease[1].
- **Environment Active Gate:** An Environment Active Gate is a proxy-like gateway for One Agents. Environment Active Gate takes away the need to connect to the SaaS or the Managed Clusters directly by receives the data and forwarding the data in compressed chunks. This allows to minimize the number of firewalls port openings needed. They are also used for cloud and remote technologies monitoring and to run synthetic monitors from a private location[2].
- **One Agent:** One Agent is a lightweight monitoring agent that collects data from hosts, processes, services and applications in real time. It automatically discovers and instruments applications across environments, providing detailed insights into their performance[3].
- **Oracle Insights:** Oracle Insights is a feature within Dynatrace that provides detailed performance insights specifically for Oracle databases. It helps in identifying bottlenecks, optimizing SQL queries and understanding the overall performance of Oracle database operations[4].
- **SQL ID:** SQL ID is a unique identifier for SQL statements in Oracle databases. It is used to track the execution of SQL queries, helping database administrators analyze performance, troubleshoot issues and optimize query execution[7,9].
- **Execution Plan:** An execution plan is a sequence of steps Oracle Database takes to execute a SQL query. It provides detailed information about the operations Oracle performs, such as table scans and joins, which are crucial for query optimization[7,9].
- **DAVIS AI:** DAVIS (Dynatrace AI Virtual Assistant) is the AI engine in Dynatrace that automates the analysis of monitoring data. It uses machine learning algorithms to identify root-cause of issues, anomalies and potential problems in real time.
- **Adaptive Thresholds:** Adaptive thresholds are dynamic thresholds used in monitoring tools like Dynatrace. Instead of relying on static limits, these thresholds adjust automatically based on historical data and trends, helping to detect anomalies more accurately.
- **Oracle Enterprise Manager (OEM):** Oracle Enterprise Manager (OEM) is a set of web-based tools aimed at managing software and hardware produced by Oracle Corporation as well as by some non-Oracle entities.

3. Literature review

The integration of AI-driven monitoring solutions into IT infrastructure, particularly for managing Oracle databases, has been the subject of extensive research and industry interest. This literature review synthesizes key findings from academic and industry sources, focusing on the role of AI in database monitoring, the challenges faced in the utilities industry and the effectiveness of solutions like Dynatrace.

Artificial Intelligence (AI) has emerged as a critical tool in the field of database monitoring, enabling more sophisticated and proactive management strategies. Traditional monitoring systems often rely on static thresholds and reactive measures, which can lead to delayed responses to performance issues and a higher incidence of false positives[6]. AI, on the other hand, offers adaptive thresholding, real-time anomaly detection and predictive analytics, which can significantly enhance monitoring effectiveness[11].

The utilities industry faces unique challenges in managing Oracle databases, largely due to the scale and complexity of the data involved. Utility companies must handle vast amounts of transactional and operational data, which are critical for maintaining service reliability and meeting regulatory requirements[11]. The traditional monitoring tools are often insufficient for identifying performance bottlenecks in real-time, leading to unplanned downtime and increased operational costs[6]. Their research indicates that utility companies are increasingly turning to AI-driven solutions to enhance their ability to monitor and secure their database environments, ensuring both performance and compliance[11].

Several studies have explored the effectiveness of AI-driven monitoring solutions, with a particular focus on platforms like Dynatrace. A comprehensive evaluation of Dynatrace's AI capabilities in monitoring complex IT infrastructures, showed that its adaptive thresholding and automated root cause analysis features significantly reduce the incidence of false alerts and improve incident response times[4,1]. This also highlights the platform's ability to integrate with Oracle databases seamlessly, providing deep insights into query performance and resource utilization[4,5].

Looking forward, the role of AI in database monitoring is expected to grow as utility companies continue to digitize their operations. The advancements in AI and machine learning will lead to even more sophisticated monitoring solutions, capable of autonomously managing and optimizing database environments. They emphasize the need for ongoing research to refine these technologies and ensure their applicability across different sectors within the utilities industry[11].

The literature clearly supports the significant potential of AI-driven monitoring solutions like Dynatrace in enhancing Oracle database management within the utilities industry. The benefits of real-time anomaly detection, automated root cause analysis and predictive maintenance are substantial. As utility companies continue to evolve and expand their digital capabilities, the adoption of AI-driven monitoring will likely become a critical component of their IT strategies, ensuring both operational efficiency and service reliability in a demanding industry.

4. Methodology

A. Study Design: This paper employs a case study methodology to explore the benefits of AI-driven monitoring in utilities industries. The focus is on improving the performance and reliability of Oracle databases, which are essential for supporting the core operations of utilities industry. These databases handle vast amounts of data and are critical for seamless functioning of various processes, making their optimal performance crucial for uninterrupted service delivery.

The case study method is well suited for this exploration as this allows for an in-depth and contextual understanding of the real-world challenges, step-by-step implementation of the process, solution and the results. By focusing on a particular organization, this paper provides insights into the complexities and nuances of integrating the AI-driven monitoring solution into an existing IT infrastructure.

Through this case study, the research examines the initial conditions of the database, the performance issues, bottlenecks and limitations. It then follows the step-by-step process of implementing the AI-driven monitoring, detailing how the AI capabilities of Dynatrace are leveraged to establish adaptive baselines, detect anomalies and predict issues. Post implementation the paper reviews the performance improvements of the Oracle database.

In the end, this case study offers a holistic examination of the outcomes of implementing AI-driven Oracle database monitoring in a utility company, providing a rich source of knowledge that can aid future projects in this industry.

B. Case Study Selection:

The case study selected for this paper focuses on a large utilities company. This company has a large foot-print of oracle databases and these databases are integral to its daily operations. The company was chosen due to its complex and large-scale data management needs, making it a prime candidate for AI-driven database monitoring.

Problem Faced: The company encountered significant performance issues in multiple applications, particularly in query execution times. During peak periods, these slowdowns led to delays in processing critical data. This caused the applications to not be able to make timely decision and deliver services efficiently. The performance issues were sporadic, making them difficult to predict and manage with the existing monitoring tools.

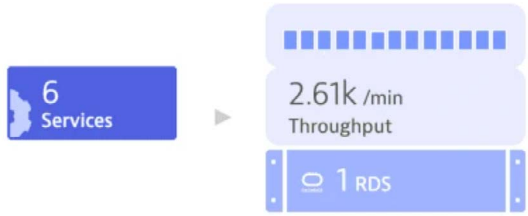

Figure 2: OA identifying applications talking to the database.

The databases frequently ran complex queries that were needed for the regular operations of the company. But these queries often caused significant performance issues, particularly when multiple queries were executed simultaneously. The existing monitoring tools were inadequate for identifying and optimizing these complex queries, resulting in slow response times and inefficient use of database resources.

The company had limited visibility into how the performance issues within the Oracle databases affected end-user experience. This lack of insights made it difficult to understand the impact a problem had on the business. This caused difficulty in prioritizing and addressing problems that had the most significant impact on customers. As a result, customer satisfaction was often compromised, leading to increased complaints and a potential loss of customer trust.

D. Implementation of Dynatrace:

Dynatrace monitors the database in a couple of ways. The first touch point for database monitoring is OneAgent (OA) on the host which is calling the database. The second, is setting up Oracle Insights plugin. Figure 1 shows the different sources of gathering data for database monitoring.

When the OA is deployed on the host it is capable of detecting all the processes and services running on the host. The OA also tracks the requests coming into and going out of a service. When an application which is monitored by OA makes calls to the database, Dynatrace is able to track these calls going to the database. Dynatrace is able to calculate the response time taken by the query based on the time the request was sent to the database from the application and the time the response was received from the database to the application. Figure 2 shows the requests going from service to the database.

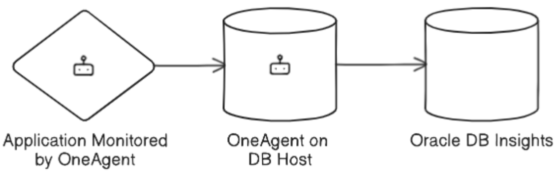

Figure 3: SQL statements captured by Dynatrace through OA on the Application.

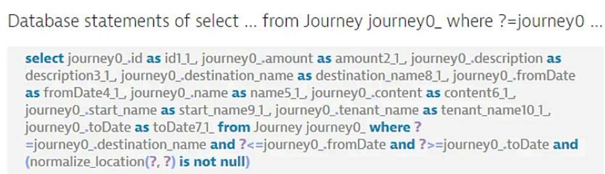

Figure 1: Different sources of data for database monitoring.

The installers for the OA can be found under “Deploy Dynatrace” option in the menu. Clicking on “Deploy Dynatrace” and selecting "Start Installation", provides platform options. This refers to the platform on which the application is running. The installation steps vary depending on the platform the application is running, for example if the application is running on a Linux system, then selecting "Linux" and the installer type (based on the OS architecture) provides three commands. First of the three commands is a "WGET" command that downloads the installer shell script from the tenant. Second, is signature verification. This step confirms if the script downloaded is secure and has not been tampered with. The third and final step is to run the installer with root permissions. Once the agent is installed then depending on the type of monitoring required a restart of all the processes running on the host might be needed (a restart is needed is full stack monitoring is desired and it is not needed in case of infrastructure only monitoring). In this case, full stack monitoring is required so all the processes need to be restarted. This will start deep monitoring of all the processes and services that are running on the host. This provides the telemetry data needed to trace the requests going from the application to the database.

Having the telemetry data (Pure Paths) from Dynatrace shows the performance of the queries from the application's point of view. The trace data will originate from a request and will follow the request through its completion. Along the way it will track and display any SQL statements called to complete the request. These SQL statements are displayed as a part of the trace and contain information such as "Client-Side Start Time", "Client-Side Response Time", Fetches, Rows Returned, type of database, the SQL statement. The SQL statement displayed masks the parameters and is truncated in case of large statements. Figure 3 shows SQL statements captured by Dynatrace through OA on the Application.

This allows quick faulty domain isolation. If in the trace the SQL statement is shown to be consuming majority of the time, then it can be concluded that the time is spent while interacting with the database. That being said, the trace does not answer all the questions related to time consumption. It does not show if the time is spent on the database or enroute to the database. In order to answer such questions more granular data is needed.

Oracle Database Insights (OI), is a plugin provided by Dynatrace that extends Dynatrace's database monitoring ability and provides deeper insights on the database. OI is an Environment Active Gate (AG) plugin, this means that the plugin sits on the AG and connects remotely to the database. This is also ideal in cases where no agents can be deployed on the database. Figure 4 shows the working of OI and how it gathers the data.

- In order to deploy OI, the following prerequisites must be met:
- AG with version 1.173+ must be installed in the default mode. Database insights does not support an AG that is configured for multi-environment support.
- Database insights requires 2.5 MB of RAM per Oracle database endpoint
- Dynatrace Server version 1.173+
- Network communication between the AG and the Oracle Server
- Oracle version 11g to 19c
- An Oracle database user with the necessary permissions

To set up OI the Oracle database user must be provided with the following permissions:

- CREATE SESSION
- SELECT_CATALOG_ROLE (including Dynamic Performance Views)
- EXECUTE permission to DBMS_XPLAN package

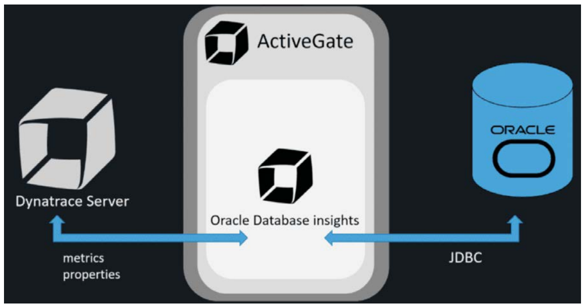

Figure 4: Working of OI and how it gathers the data.

The below code can be used the provision a user with the required permissions.

OI can be configured in Dynatrace by following the below mentioned steps:

- Navigate to "Settings" > "Monitoring" > "Monitored Technologies"
- Find the Database Insights: Oracle row and click the pen icon to edit it
- Define your Oracle database endpoint. All fields are mandatory:

a. Oracle host
b. Port by default the port is set to 1521, this should be changed to the port the database uses
c. Connection type: Service or SID
d. Service/SID: database identifier
e. Database user and Database password
f. Monitored database name
g. Select the check box to accept the redistribution license agreement for the Oracle JDBC driver. Dynatrace uses this to fetch the data from your Oracle server.
h. Click Add database.

```
CREATE USER oracleinsights IDENTIFIED BY
password
default tablespace users
temporary tablespace temp;

GRANT CREATE SESSION,
SELECT_CATALOG_ROLE TO <oracleinsights>;
```

Figure 5: shows the OI configuration page in Dynatrace.

OI provides a deep and granular view of the database and its health. Once configured, OI creates an entity (custom device) for the database instance. This custom device can be found in Dynatrace in a couple of ways.

If the database is already identified through the communication from the application (via application's OA), then an instance of the database can be found in the "Database" section of the menu. Clicking on the database, shows the throughput for the database, number of services talking to the database, SQL transactions, modification, queries/ procedures and availability of the database. Under the throughput a custom device section is found, this is where the OI created custom device for the database can be found.

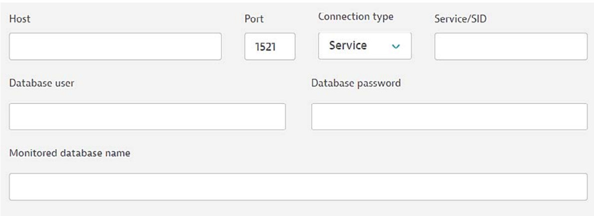

Figure 5: OI configuration page in Dynatrace

The custom devices created can also be directly accessed from the "Technology Overview" section in Dynatrace. The technology overview page provides an option to view all the created custom devices by clicking on the "Custom Devices" button on the top right corner of the page.

Once accessed, the OI page shows a brief health overview of the database and the communications to/ from the database. Below the overview section, OI provides detailed performance, time breakdown, PGA memory and SGA memory metrics. Figure 6 shows the detailed metrics captured by OI.

Also available on this page is the most useful and important feature provided by OI, "Most time-consuming Oracle statements". This feature helps to understand and analyze which Oracle statements are the most expensive and most frequently called. This feature provides up to 100 statements with the highest elapsed time calculated as a number of executions multiplied by the execution time for the selected timeframe.

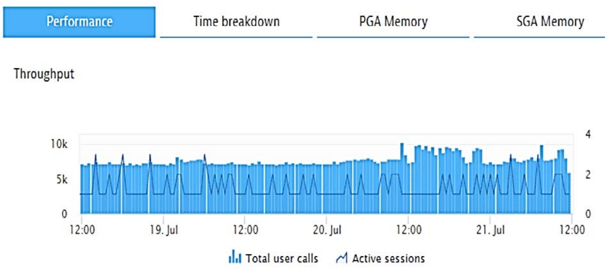

Figure 6: Metrics pulled by OI

This page displays the complete SQL statements along with their schema, SQL id, Elapsed time, Executions and Rows processed. Expanding on the details of a SQL statement provides the statement performance (Elapsed time, Execution count, CPU time, Wait time), Waits (Application wait time, Concurrency wait time, Cluster wait time, User IO wait time) and Execution details (Executions, Processes rows, Buffer gets, Disk reads, Direct writes, Parse SQL calls) as trend over time for the selected timeframe. In the details section of the SQL statement the execution plan for the SQL statement can also be downloaded. Figure 7 displays the details section of a SQL statement.

E. Using the data to optimize performance:

The configuration of OA and OI provides a holistic view of the database health and the health of the application accessing the database. As a part of this case study, a couple of applications that were experiencing performance issues were chosen. The applications were under the suspicion that the database was the faulty domain and the reason for the performance issues. The performance baselines of critical functionalities that were experiencing slowness were noted as pre-Dynatrace implementation baselines.

With AI-driven monitoring provided, by Dynatrace, the identification of faulty domain is automated. Post the configuration of monitoring, running through the problematic workflows generates the monitoring data needed. The AI-engine (DAVIS), in Dynatrace processes through all the data and identifies anomalies. Once identified, DAVIS analyzes the dependencies for the related entities and narrows down the potential root-cause. The problem generated by Dynatrace contains the anomaly identified, the impact of the problem and the potential root-cause. This quickly helps isolate the faulty domain for the problem.

Figure 7: Query details chart in OI.

For one of the workflows that was experiencing performance issues, Dynatrace was quickly able to identify that the faulty domain was not the database but instead it was the service calling the database. This was identified from the potential root-cause derived by DAVIS. To better understand the issue, the services accessing the database were analyzed from the "Database Overview" page in Dynatrace. Using the "backtrace" and "response time hotspots" features, it was found that majority of the time spent was at the database usage step. Charting the service flow, from the service accessing the database, an interesting information was found. The average response time per request in the database was not significant. But, the number of requests made to the database were significant. Also, each request from the service was reaching out to the database over a few hundred times. This is indicative of an N+1 query problem. Where due to poorly written application code, the request reaches out to the database multiple times instead of reaching out the database once and gathering all the needed information. To confirm this, traces (Pure Paths) for the service are accessed from the service-flow page. These traces show all the requests made to the database for individual transaction. Hence, confirming the suspicion of an N+1 query problem. These findings are used to refactor the code and reduce the number of calls to the database by approximately 58% and in-turn improving the performance of the workflow by almost 30%.

Analyzing another workflow that was experiencing slowness, DAVIS identified the root-case as the database. To narrow this down further, few factors were needed to be checked. One is the health of the infrastructure, on which the database is hosted. As OA gathers detailed host health metrics, this can be easily checked using the host overview page. After concluding that the database was residing on healthy hardware, the focus moves to the database instance itself. This second step is checking the tablespaces, PGA & SGA memory, connection pool information etc. After confirming that the database instance is also healthy[5], the workflow is traced in Dynatrace. Looking through these traces in Dynatrace, a few queries were identified which were taking majority of the execution time for the entire transaction. Analyzing these queries in OI show the average execution time to be very slow. Also, looking at the number of executions it can be deduced that these queries are very crucial for the functioning of the workflow and hence were called multiple times. The slow average execution time coupled with the high number of executions made these queries most impactful for the performance of the application and its critical workflow. By downloading the explain plan for these queries the cost of executing the queries can be understood. This can be used as a good indicator for comparison after optimizing these queries. Using OI, the SQL ids for these queries can be identified. This is a very useful piece of information as the SQL ids can be used to find optimizations. Using the SQL ids, the exact queries can be easily identified in Oracle Enterprise Manager (OEM). OEM is a specialized tool for Oracle databases which can be used to find performance optimizations if the exact query is known. Optimization suggestions for the slow queries are identified from OEM. It was found that the queries needed a couple of optimizations, like changes to the indexes, using updated execution plan, modifying the query to pull less projections etc.[9]. After making the optimizations to the queries, the average execution time for the same improved by over 70% greatly improving the performance of the application and all its workflows.

F. Proactive alerting and dashboarding:

Proactive alerts and dashboards can be created for Oracle databases, using Dynatrace. Using OA and OI, Dynatrace gathers multiple crucial metrics for monitoring Oracle databases. Traditional monitoring systems often rely on static thresholds, which can result in false positives or fail to detect critical anomalies. As traditional monitoring expects known thresholds for alerting and that might not always be accurate. In contrast, Dynatrace uses AI-driven adaptive thresholding to dynamically adjust baselines based on continuous real-time data analysis. This ensures that alerts are generated only when deviations are statistically significant and likely to impact system performance. This approach reduces false alerts and enhances the accuracy of incident detection by automatically adapting to variations in workload and operational patterns.

The metrics gathered by OA and OI, are integrated into highly configurable dashboards within Dynatrace, providing comprehensive and real-time insights into the health and performance of Oracle databases. These dashboards offer a consolidated view of key performance metrics, historical data trends and active alerts, enabling IT teams to monitor the database environment efficiently. By leveraging proactive alerting and dashboarding, utility companies can anticipate and mitigate potential performance issues proactively. Thereby ensuring higher system reliability, reduced downtime and improved operational efficiency.

5. Conclusion

The integration of AI-driven monitoring through Dynatrace within Oracle database environments offers a transformative approach to managing the complex and demanding requirements of the utilities industry. This paper has demonstrated how AI can enhance database observability, by providing insightful data, automating root-cause detection, setting up adaptive thresholds for alerting and creating interactive dashboards. By leveraging these advanced capabilities, utility companies can proactively address potential issues, reduce downtime and ensure that their critical database systems operate at peak efficiency, even during periods of high demand.

The case study presented in this paper illustrates the tangible benefits of implementing AI-driven monitoring, including significant improvements in query execution times, resource utilization and overall system stability. The use of AI-driven adaptive thresholding and intelligent alerting has proven to be particularly effective in minimizing false positives and ensuring that the IT teams are alerted only to actionable issues. This not only streamlines operations but also empowers teams to make data-driven decisions.

As the utilities industry continues to evolve, the ability to maintain and optimize Oracle databases will remain a critical factor in ensuring reliable service delivery and operational excellence. The findings of this study suggest that AI-driven monitoring, as facilitated by Dynatrace, provides a robust solution to the challenges faced by utility companies in managing their database environments. By adopting these technologies organizations can enhance their resilience, improve customer satisfaction and position themselves for future growth.

6. Future Work

Future research should extend the findings of this study by exploring the long-term impact of AI-driven monitoring on various aspects of Oracle database management within the utilities industry. Potential areas for further investigation include the cost-benefit analysis of implementing AI-driven tools over extended periods, particularly focusing on the reduction of operational costs and the optimization of resource allocation. Additionally, studies could examine the scalability of AI-driven monitoring solutions in larger and more complex database environments, assessing their effectiveness in managing growing data volumes and increasingly intricate queries.

Another promising area for future research involves the application of AI-driven monitoring across different sectors within the utilities industry, such as water management, waste management and renewable energy. Comparative studies could provide insights into how the unique demands of each sector influence the effectiveness of AI-driven tools, thereby identifying best practices and customization strategies.

Moreover, the integration of AI-driven monitoring with other emerging technologies, such as cloud computing and edge computing, offers an exciting area for exploration. Research could evaluate how these technologies can be combined to further enhance database performance, improve data security and ensure compliance with regulatory standards. Finally, as AI and machine learning algorithms continue to evolve, ongoing studies should focus on refining these technologies to increase their predictive accuracy and automation capabilities, ensuring they remain at the forefront of database management innovation in the utilities industry.

7. References

1. https://docs.dynatrace.com/docs/get-started/what-is-dynatrace
2. https://docs.dynatrace.com/docs/setup-and-configuration/dynatrace-activegate
3. https://docs.dynatrace.com/docs/setup-and-configuration/dynatrace-oneagent
4. https://docs.dynatrace.com/docs/platform-modules/applications-and-microservices/databases/database-services-classic/database-insights
5. https://www.dynatrace.com/news/blog/five-easy-steps-improve-database-performance/
6. https://www.linkedin.com/pulse/3-reasons-why-traditional-rule-based-monitoring-ralph-aboujaoude-diaz/
7. https://ieeexplore.ieee.org/document/8455009
8. https://ieeexplore.ieee.org/document/8731511
9. https://ieeexplore.ieee.org/document/9922783
10. https://scholar.google.com/scholar_lookup?title=Comparative%20study%20of%20relational%20and%20non-relations%20database%20performances%20using%20Oracle%20and%20MongoDB%20systems&publication_year=2014&author=Azhi%20Faraj&author=Bilal%20Rashid&author=Twana%20Shareef
11. Cheng CT, Ganganath N, Lee TK and Fok KY. “Utilities,” in The Fourth Industrial Revolution, 2021;197-213.