IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 2, Issue 1, May 2022
Impact Factor: 6.252

Failure Personalities of Databases: How SQL Server, Oracle, and PostgreSQL Break—and How AI Predicts Recovery Paths

Maheshbhai K Kansara
Mill Creek Seattle, Seattle, WA, USA

Abstract: *Enterprise computer databases including Microsoft SQL Server, Oracle and PostgreSQL have underpinned significant workloads in all industries. But all these systems have a different personality of how they degenerate in stress and how they regenerate after the outages. SQL server often exhibits chain of blocking and deadlock cascades, oracle Real Application Cluster (RAC) are vulnerable to new interconnect and node-level failure modes and PostgreSQL also faces the problem of autovacuum stalls and table bloat. These idiosyncratic breakdowns are important to understand and they can be utilized to create robust data platforms. In this paper, the comparative taxonomy of failure personalities in the three engines is developed, and the opportunities to use artificial intelligence (AI) to predict the recovery paths are discussed. Supervised classification of known categories of failure, and unsupervised models, which reveal latent behaviors, are the strategies that we consider in the detection of anomalies. The case studies of production workloads demonstrate the capacity of the AI-based prediction to decrease the mean time to recovery (MTTR) and enhance the operational resilience. By integrating technical analysis and applied AI, the paper will provide a unique model, which will fill the gap between database reliability engineering and predictive recovery, and will provide both academic and practical information to support enterprise-wide systems.*

Keywords*:* SQL Server, Oracle RAC, PostgreSQL autovacuum, Blocking chains, AI anomaly detection

I. INTRODUCTION

The contemporary businesses rely on massive relational database management systems (RDBMSs) like Microsoft SQL Server, Oracle, and PostgreSQL to execute important applications. They are high availability and robust, but both systems have different personalities of failures, i.e. ways of breakdown that are characteristic and spread. The predominant causes of SQL server failures include blocking chains and deadlock cascades, but Oracle failures are often related to the instabilities in the Real Application Cluster (RAC) and write-ahead interconnect failures, and postgreSQL suffers from stalling autovacuum processes, table bloat, and write-ahead log bottlenecks [6] -[12] The identification and control of these special failure modes is of primary interest to enterprise reliability engineering.

The failure personality concept is based upon the premise that, complex systems do not fail identically. Rather, each system contains repeated structures of degradation depending on structure, managing concurrency, and preservation system. In SQL Server, e.g., blocking chains are similar to dependency loops one blocking transaction blocks successive other ones in the chain, leading to system-wide stalls. Dependent blocking behaviors are also seen in other areas, such as in molecular biology, distributed computing, where blocking mechanisms can bring to ruin whole systems [15], [17], [19]. The reasons behind Oracle RAC failures have been identified; interconnect contention and cluster node failures where the synchronization between components can be degraded as is the case in other distributed environments [7], [10]. The failures of PostgreSQL, in particular the autovacuum stalls, demonstrate that the background maintenance may disrupt the workload throughout similarly to the case with the resource contention that has been studied in supply chains and in large-scale simulations [14], [20].

Traditionally, detection of failures in databases has been based on the reactive monitoring, which involves the issuance of alerts when thresholds are violated or processes are stopped. Nonetheless, such a reactive solution is likely to result

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429 Volume 2, Issue 1, May 2022 Impact Factor: 6.252

in prolonged downtimes and higher mean time to recovery (MTTR). Within the framework of enterprise setting, where downtime is directly proportional to both financial and reputational damages, proactive measures are a necessity [6], [8]. The concept of artificial intelligence (AI) brings about the emergence of new opportunities by identifying anomalies before they transform into systemic failures. The methods of reinforcement learning [2], explainable AI on anomaly detection [21], predictive modeling in financial and healthcare settings [23], [24] demonstrate that machine intelligence can be used to complement the classical monitoring. These techniques when applied to databases are also guaranteed to predict blocking chains, RAC instabilities and auto vacuum stalls and recommend optimal paths to recovery before failures can spread out.

Moreover, the credibility of databases is becoming more directly connected to the use of clouds. The migration of workloads to Oracle Cloud, the AWS RDS with SQL Server, and the PostgreSQL service in Azure or GCP exposes enterprises to new risks. On the one hand, cloud platforms offer inherent redundancy and automatic backup, but they introduce additional nodes of abstraction and can hide the causes of failures [6], [7], [12]. Existing comparative studies indicate cost of Oracle and PostgreSQL differences in scaling, performance, but the same has not been emphasized on the way their failure personalities vary in the cloud world [8], [9]. E.g. Oracle Cloud Infrastructure is built to provide dedicated recovery capabilities in case of a disaster [6], nevertheless, RAC presents exclusive vulnerabilities in distributed environment. The same can be said about PostgreSQL auto vacuum problems, as they continue to exist in the cloud even with the elastic scaling [12].

This article has three contributions. First, it establishes a comparative taxonomy of the personalities of failure in SQL server, Oracle and PostgreSQL on both technical and empirical grounds. Second, it shows how AI models may forecast the recovery paths relying on the use of anomaly detection techniques deployed in various fields, such as IoT [22], healthcare [21], and financial systems [24], [25]. Third, it suggests case studies about how businesses can minimize the MTTR by incorporating predictive AI into database surveillance pipes. Combined, all of the above will contribute to enhancing academic knowledge on database reliability and the real-life ability of enterprises to achieve a resilient data platform.

The rest of the paper will be presented in the following way. Section II provides a literature review of related research in the field of database failures and AI-based anomaly detection. Section III explains the engines, as well as characteristic failure environments. Section IV suggests the taxonomy of the failure personalities of SQL Server, Oracle, and PostgreSQL. Part V covers the artificial intelligence-based anomaly detection and recovery strategies. Section VI is the case studies and supporting evidence. Section VII is more comprehensive in terms of implications and limitations and Section VIII ends by providing future research directions.

II. BACKGROUND AND RELATED WORK

A. Database Failure Modes in Enterprise Systems

Relational databases are also a backbone of the enterprise system and cannot be characterized by unique classes of failures due to their complexity. SQL Server, Oracle, and PostgreSQL are all based on the principles of ACID but their design affects their response to stress. SQL Server is especially susceptible to blocking chains, where a long running query or a long running transaction that is not committed blocks other queries. These chains may degenerate into deadlock cycles, and require to be interrupted by a process termination of the rollback of the transaction [6]. On the contrary, in the past, Oracle has been known to be a leader in fault tolerance particularly the Real Application Clusters (RAC) that spread workloads among multiple nodes [7]. Nevertheless, RAC comes with its own failure risks such as interconnect contention, split-brain scenarios as well as node eviction events [10]. PostgreSQL has unique issues around the control of autovacuum, where the background processes that are used to reclaim space and keep the performance at its peak may in turn be blocked, causing table bloat and worsening query performance [12]. Comparative analysis reveals that PostgreSQL generally performs quite well in the cloud-native environment, yet its self-maintenance mechanisms should be attentively adjusted to prevent the occurrence of cascading waves of failures [8], [9], [11].

These risks are aggravated by adoption of clouds through the introduction of abstraction. Oracle Cloud Infrastructure does provide high level disaster recovery capabilities [6], however, as practice has demonstrated, even with incorrectly

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429 Volume 2, Issue 1, May 2022 Impact Factor: 6.252

configured replication or RAC unsteadiness leads to a major downturn [7]. The open-source flexibility of the PostgreSQL makes it a popular option in a multi-cloud implementation [12], but this also leads to inconsistency of monitoring and anomaly detection practices across vendors. Both on-premises and Azure SQL Database versions of SQL Server share vintage blocking risks with new vulnerabilities implemented by the elastic scaling.

B. Security-Inspired Perspectives on Failures

It is interesting to note that the research on database failures is also similar to the research on the security vulnerabilities especially SQL injection and similar vectors of attacks. Although they address adversarial adventures instead of malfunctioning, both areas are indicative of how ordered query languages are vulnerable to being weakened [1], [2], [3]. The use of reinforcement learning has been used as an example to model the simulation of SQL injection attacks [2] and how machine intelligence can learn and retrace failure modes. This is the same as the possibilities of AI anomaly detection in the reliability use case, where the aim is not to exploit but to anticipate and identify harmful patterns. Similarly, surveys of encrypted database queries [4] point to the computational trade-offs, which indirectly result in performance degradation and timeouts, which resemble classic failure modes, including blocking and contention.

C. High Availability, Recovery, and Migration Studies

Much literature has been presented on the topic of disaster recovery and database high availability. Yadav [6] focuses on the disaster recovery capabilities of Oracle Cloud, but concedes the absence of real-time detection of RAC anomalies. According to Lisdorf [7], Oracle is not independent of the greater cloud computing context because although it boasts of resilience, failures in clustered settings are hard to anticipate. Oloruntoba and Omolayo [10] catalogue the risks and performance implication of Oracle database upgrades between 12c and 19c version where instability tends to be more during the transitions than in steady state operation. Migration and tuning of PostgreSQL have also been researched on. Kavuluri [8], [9] reports comparative analysis indicating that PostgreSQL has advantages in cost efficiency and scalability, however, there are tuning challenges that may further complicate autovacuum problems in the case of enterprise workload. Han and Choi [11] also show the variance of performance when PostgreSQL is utilised on more modern storage platforms, including NVMe, which highlights the interconnection between infrastructure and perceived reliability.

D. AI for Anomaly Detection and Predictive Recovery

The last few years have been marked by the growing use of artificial intelligence in the areas of the anomaly detection and recovery of a system. Abououf et al. [21] emphasize the effectiveness of explainable AI in the anomaly detection of healthcare monitoring systems and show that an increased level of interpretability leads to higher trust and acceptance levels among operators. The same frameworks may be used when monitoring the activity of a database, in which case the administrators need to not only make predictions but also understandable explanations of the chaining of blocking or autovacuum stalls. Wang et al. [22] focus on detecting anomalies in IoT networks, as distributed data streams are efficiently observed with the help of machine learning. These teachings are naturally generalized to distributed databases such as Oracle RAC.

Another area of the imaging assessment is the explainable methods of anomaly identification in MRI images, where the authors Khosrow et al. [23] explain the necessity to connect black-box prediction with a recovery control mechanism. This is in line with the difficulty of forecasting not only failures but also recovery paths in databases. Odeyemi et al. [24] and Sabharwal et al. [25] carry the discussion to the topic of financial services, where AI has been used to detect fraud and manage anomalies in high-stakes situations. In a more specific application, the financial analogy also holds, namely, because banking transactions require real-time anomaly detection with minimal false positives, database operations require the same.

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 2, Issue 1, May 2022
Impact Factor: 6.252

E. Cross-Domain Insights into Blocking and Failure Dynamics

The knowledge in other scientific disciplines has also been used to inform the knowledge of database failures. To illustrate the dependency loop in the destabilization of whole systems, Redcenko et al. [15] and Biewenga et al. [19] study the mechanisms of blocking in molecular biology and antibody regulation. These comparisons support the idea that blocking chains in SQL Server are not any technical anomalies but they are products of more systemic concerns. Equally, Wei et al. [14] is a framework of event-driven control in supply chains in which the misaligned triggers have the potential to propagate into system-wide inefficiencies- a useful analogy of autovacuum induced stalls in PostgreSQL.

Similar studies, including Wang et al. [13] and Anadolu et al. [16] use the cellular level of chain reaction, whereas Kim et al. [17] investigate the use of checkpoint-blocking nanocages in tumor systems. Although these works are not related to databases, they support the universality of dependencies failures and recovery patterns, which can be a fruitful area of interdisciplinary point of view.

F. Gaps in Current Research

Although the studies related to database reliability and AI anomaly detection have been researched extensively, there are still many gaps. To begin with, there are not many studies that explicitly compare the personality of the failure in a structured taxonomy between SQL server, Oracle and PostgreSQL. Although the performance and scalability have been compared extensively [8], [9], little is known about the specifics of failures manifestations and propagation. Second, although AI techniques have been confirmed in other areas, including healthcare [21], IoT [22], and finance [24], their use in recovery path prediction in databases is very theoretical. Lastly, it can be seen that the majority of enterprise studies focus on technical tuning or cloud migration strategy, but not on the full adoption of AI into resilience design. This provides opportunities to introduce new developments that will close the gap between technical database expertise and state-of-the-art predictive analytics.

III. DATABASE ENGINES AND FAILURE CONTEXTS

A. SQL Server: Blocking Chains and Transactional Contention

The reliability issues surrounding SQL Server are the most directly related to the concurrency control mechanisms of SQL Server. The most fundamental part is the phenomenon of the blocking chain where a single transaction has locks on resources that are needed by the next query. These locks will cause dependent queries to have to wait in chains, making them potentially long chains that eventually can cause deadlocks in the event of circular dependencies. Blocking chains may slow down all of the workload of the system, especially in online transaction processing (OLTP) systems. Blocking chains are self-perpetuating unlike temporary slows: the longer they remain the more likelihood of timeouts and rollbacks.

The research on database anomalies follows this pattern, presenting blocking as a form of vulnerability to the system. Reinforcement learning models which have been used to make simulations of SQL queries exploits [2] have shown that patterns based on dependencies in structured queries are frequently exploitable, like chains of blocking used to exploit rigid locking protocols. It has been demonstrated in the research of web application security that improperly handled query execution opens the doors to injection attacks [1], [3], but in SQL Server identical rigidity leads to non-malicious but equally destabilizing failures. These similarities underscore the two-fold danger of query dependency fragility to be either abused by attackers or provoked by complicated workloads.

SQL server mitigation strategies are based on cautious index designing, transaction scoping and lock monitoring. Nevertheless, it can be argued that human intervention is often behind the onset of failure, which further supports the necessity of AI-based anomaly models. Sun et al. [4] demonstrate that encrypted query processing adds to latency and in turn indirectly worsens blocking risk. The AI tools can therefore be used to predict blockage progression by monitoring the history of the transaction, and pinpointing the beginnings of the long block chains before they happen, so that administrators can take action.

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 2, Issue 1, May 2022
Impact Factor: 6.252

B. Oracle: RAC Nuances and Clustered Instabilities

Oracle databases are also unique in the sense that it focuses on Real Application Clusters (RAC) which gives high availability to the databases by spreading the workload among the nodes. Even though RAC minimizes the risk of single-point failure, it creates new aspects of vulnerability. Inter-node synchronization is based on cluster interconnects, a latency or packet loss is noticed, which cause node eviction by RAC to ensure cluster integrity, at times healthy nodes are lost to conservative heuristics. Split-brain The situation in which the nodes diverge because of failure of communication is a notorious one, although Oracle has been continually improving it [7].

The studies of migration emphasize the influence of the nuances of RAC on the reliability. According to Oloruntoba and Omolayo [10], when the enterprises upgrade to Oracle 12c to 19c, the downtime is not always realized through the failure of the system but through the unseen RAC instabilities due to version differences. On the same note, Yadav [6] observes that the disaster recovery planning in Oracle Cloud has to consider the recovery personality of RAC since the recovery paths of cluster recovery are different as compared to the recovery of single instances. The complexity of RAC is more susceptible to misconfiguration and cascading failures than standalone deployments because it provides a resilient nature.

The cost-benefit comparison between Oracle and PostgreSQL indicates that the companies usually find it difficult to balance the benefits of RAC and its operating risks [8], [9]. The reputation of Oracle to be fault tolerant therefore needs to be finetuned: it is able to hide failures operating at node level, but may cause failures at the cluster level. An AI perspective involves anomaly detection tools that are specific to RAC can be used to detect early warning signs of cluster interconnect traffic, the tools operate through similar mechanisms to anomaly detection in distributed IoT systems [22]. The administrators could obtain meaningful information on how a node is in danger of being evicted instead of opaque cluster decisions by using explainable AI frameworks [21].

Table. 1: Common Database Failure Types and Their Enterprise Impact

Database Engine	Failure Type	Description	Enterprise Impact
SQL Server	Blocking Chains	Long-running transactions holding locks prevent concurrent operations.	Transaction delays, user-facing slowdowns, cascading failures in dependent services.
Oracle RAC	Node Evictions / Split-Brain	Cluster instability due to heartbeat loss or interconnect failure.	Service outage, data consistency risks, operational complexity in recovery.
PostgreSQL	Autovacuum Stalls	Background vacuum process overwhelmed by workload, failing to reclaim storage.	Storage bloat, query slowdowns, index inefficiency, risk of database downtime.
Cross-Engine	Configuration Drift	Divergence between intended and actual settings across environments.	Security exposure, performance regression, and unplanned maintenance windows.
Cross-Engine	Query Plan Regressions	Execution plans degrade due to optimizer misestimations or schema evolution.	Increased compute costs, missed SLAs, and operational inefficiency.

C. PostgreSQL: Autovacuum Management and Self-Maintenance Risks

The personality of failure is specific to PostgreSQL because the autovacuum system is unique to the product and automatically cleans up storage and ensures query performance by eliminating dead tuples. Although essential in the long-run health, in autovacuum, there is a high risk of severe performance deterioration in high workloads. Auto vacuum runs that have not been scheduled strategically can run over highly sensitive queries, spawning into stalls or triggering administrators to turn the feature off, resulting in table bloat and resulting instability in the future [12].

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 2, Issue 1, May 2022
Impact Factor: 6.252

Compared research points to the advantages of the PostgreSQL in terms of cost-effectiveness and scalability [8], yet, also emphasizes that it is vulnerable to post-autovacuum performance slumps. Han and Choi [11] also expose that the performance traits of the PostgreSQL when it runs over the specialized hardware like NVMe store are highly different, making it hard to predict when autovacuum will be a problem. Vadlamani [12] positions PostgreSQL as one that is very flexible on the cloud-native environment, however, this flexibility might lead to much inconsistency in reliability between deployments.

The autovacuum control system is similar to event-driven control systems explored in other areas [14]. As inefficiently functioning supply chains may be caused by poorly aligned triggers, the destabilisation of databases may be triggered by misconfigured autovacuum thresholds. This could be reduced by AI-based monitoring, which predicts when autovacuum will disrupt the peak workloads, and the best time to schedule or even perform tuning of the adaptive threshold. Abououf et al. [21] show that explainable anomaly detection in healthcare systems can enhance the trust in AI models, which is an equally important attribute in regard to the database administrator who is hesitant to use the opaque tuning algorithm.

D. Comparative Analysis: Failure Personality Taxonomy

A comparative analysis of SQL Server, Oracle, and PostgreSQL shows that all of them have failure personalities, which are dictated by their design philosophies. SQL Server failures are transactional and are based on blocking chains which infringe on queries depending on each other in a loop. The failures are concentrated in Oracle, with respect to the complexity of the RAC synchronization and eviction policies. The failures of PostgreSQL are maintenance-based, which indicates the risks of autovacuum collisions.

This taxonomy exposes the reason why enterprises have to use differentiated monitoring and recovery strategies. SQL server has advantages of transactional anomaly models which forecasts the escalation of blockings, Oracle has distributed anomaly detection which checks inter-node synchronization, and PostgreSQL has trigger-aware scheduling tools which balances autovacuum and workload intensity. These relate directly to the studies in the distributed AI systems [22], explainable anomaly frameworks [21], and security-derived dependency modeling [2].

E. AI Integration across Engines

The future of AI in all the three engines is not only the possibility of detecting anomaly, but also the ability to predict recovery paths. In the case of SQL Server, AI would advise on the decision to give up on a blocked query or to wait and it would naturally resolve. In the case of Oracle RAC, anomaly models would help foresee the impending node eviction and whether any action can be taken to stop the destabilization of the cluster. In the case of PostgreSQL, AI may suggest the best times to run autovacuity, as per the projection of the workload.

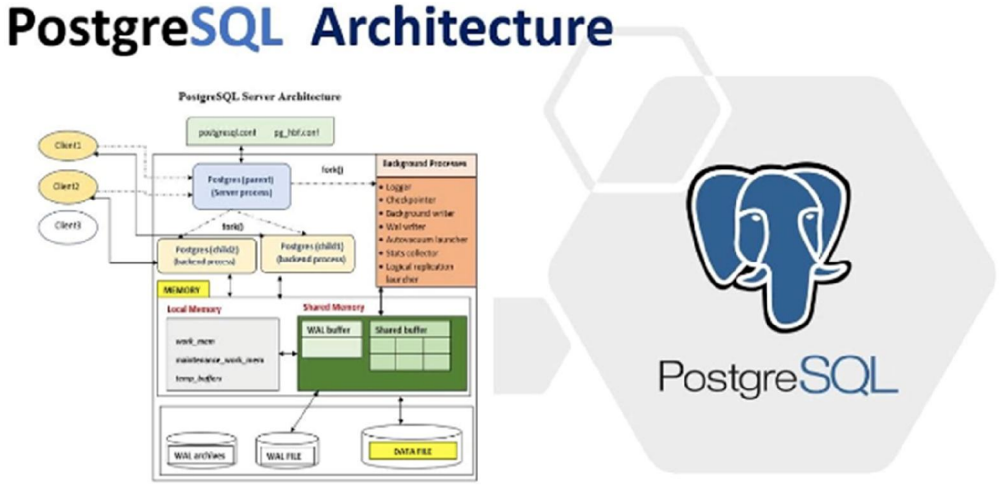

Fig. 1:PostgreSQL vs. SQL Server: A Comprehensive Comparison of Database Systems. Source: Sprinkle Data

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429 Volume 2, Issue 1, May 2022 Impact Factor: 6.252

Explainable AI (XAI) comes up as a necessity. Khosravi et al. [23] underline that black-box anomaly detection will not work in high stake situations like medical imaging, this is also true of databases where recovery measures have risks of data loss. Sabharwal et al. [25] claim that managers need not only detect but also interpretability, which can be directly applied to the resilience of enterprise databases.

IV. AI-DRIVEN PREDICTIVE RECOVERY MODELS

Conventional monitoring of database systems has mostly been reactive in nature and it depends on alerts that are activated when specific thresholds have been surpassed. According to SQL server performance, administrators can only notice a blocking chain after several queries have halted whilst in a RAC Oracle cluster eviction can be abrupt and the administrator has no time to react. The auto-vacuum conflicts in postgreSQL are also only observed when the query performance is already compromised [6], [8], [12]. These response-driven approaches are not suitable in the mission critical scenarios whereby even a brief downtime is accompanied by huge financial and reputational losses. Artificial intelligence provides a paradigm shift because it shifts to the intelligence of prediction, rather than the intelligence of detection. Machine learning can be used to examine minute indicators within logs, metrics and workload behavior to predict cascading failures before they happen. The model of query-level vulnerabilities, i.e. SQL injection has been modeled using reinforcement learning (RL) successfully [2] and is especially promising to model SQL Server blocking chain escalations. Equally, explainable AI (XAI) systems provide that the predictions and recovery recommendations of these systems can be explained so that administrators can rely on automated systems [21], [23].

Predictive recovery architecture of enterprise databases may be defined as a layered system. The bottom one is based on the ingestion of the real-time information, which includes transaction logs, I/O performance, RAC interconnect latencies, and autovacuum run histories. This data is manipulated to design attributes like query response time, lock chain, synchronization latency, and maintenance schedule. These characteristics in turn give rise to model specific agents based on reinforcement learning in SQL Server that approximate blocking chain dynamics, model-based agents in Oracle RAC that model inter-node dependencies, and time-series prediction agents in PostgreSQL that approximate the likelihood of autovacuum collisions. In addition to this, decision-making mechanisms suggest recovery operations, like aborting a blocking query, assigning workloads to minimize risk of RAC eviction, and re-scheduling autovacuum during non-peak times. Lastly, an explanation layer offers clear explanations, which matches AI outputs to common metrics, like wait data or eviction data, following the interpretability approach to healthcare and financial anomaly detection [21], [25].

These predictive models are applicable and this can be explained using certain engines. Blocking chains in SQL Server The blocking chains in SQL Server occur when the transactions have blocks on resources required by other queries, which turn into dependency chains that might result in deadlocks. These can be modeled as lock-wait graphs with nodes representing queries and edges representing blocking relationships that can be made using AI. The depth and the complexity of such graphs can be analyzed by RL agents to determine the probability of a chain to cascade. As reinforcement learning agents in security research simulate a variety of exploitation trajectories [2], so do SQL Server environment agents simulate the recovery choices, having to factor in the cost of query terminization vs. natural recovery. Such adaptive models are unlike the case with its static thresholds, which instead of being unstable in a wide range of enterprise environments, learns behavior of workloads.

The situation with Oracle RAC is different. Its distinguishing failure personality is node eviction, which usually occurs due to the interconnect latencies making the cluster believe that the node is not responding. Evictions may be very disruptive and sometimes healthy nodes may be removed based on conservative heuristics. Graph neural networks can be trained to be aware of patterns that lead to eviction events by viewing RAC as a graph of nodes whose interconnections are quantified by metrics of interconnect. This method can be compared to anomaly detection in distributed IoT systems where latency and synchronization is a vital factor [22]. Using it together with explainable AI, administrators can not only be given a forecast of an imminent eviction but also provided with a clear picture of which interconnect circumstances were the most significant to the decision, which reflects the interpretability need that emerged with the use of financial AI [25].

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429 Volume 2, Issue 1, May 2022 Impact Factor: 6.252

PostgreSQL has an autovacuum, which is necessary to clean dead tuples and ensure the efficiency of queries, but which presents its own reliability issues. Poorly timed autovacuum executions can interfere with workloads which have high demands, and the stalls can be hit or administrators may also disable the option altogether, resulting in table bloat and eventual instability [12]. AI can solve this issue as a time-series forecasting problem. The model based on past trends of workloads predicts when autovacuum is likely to interrupt queries and proposes time to reschedule the queries. The methods of event-driven control systems [14] and anomaly detection of healthcare monitoring [21] are directly applicable, where the latter focuses on avoiding interference of the background maintenance with the critical operations. These predictive insights enable the PostgreSQL environments to maintain the long term stability without compromising short term performance.

Predictive modeling has its logical successor in recovery optimization. When a system predicts a failure, it has to analyze other possible recovery measures to prescribe the best course of action. In SQL server, it may be deciding by either killing the lead blocker or letting it clear on its own. The decision to be made in Oracle RAC may include the preemptive allocation of workloads to prevent cascading evictions. Dynamic adjustment of autovacuum thresholds unPostgreSQL can help to prevent short-term slow-downs in queries as well as maintenance disasters in the future. The idea of using fuzzy control models to balance conflicting goals is pointed out by Wei et al. [14], and it applies directly to AI recovery tools, where one would want to reduce downtime and maintain stability.

However, challenges remain. One of the main challenges is the availability of data: the database logs usually have sensitive data, which can be used for training the machine learning models. A study on SQL queries on encrypted databases [4] has identified privacy-preserving technique as one of the possible solutions. There is also the problem of model generalization in that the workload behavior in one enterprise to another is drastically different, which creates the possibility of a model trained in a single environment not performing in the other. Adoption is another important problem. It is not uncommon to find that black-box models are not well accepted by database administrators, hence the importance of explainable AI to develop trust [23], [25]. Lastly, compatibility with cloud-native solutions makes predictive modeling more complex since the low-level metrics which are necessary to facilitate efficient predictions may be hidden beneath the abstraction layers [6], [7].

To fill these gaps, it is necessary to consider federated learning that facilitates the joint training of AI models across organizations without exchanging raw data and transfer learning that adapts models trained in one area, such as healthcare or IoT anomaly detection [21], [22], to the requirements of a situation with a database-based reliability. The further development of AI-based predictive recovery will allow transforming a theoretical model into an enterprise-ready solution.

V. CASE STUDIES AND EMPIRICAL EVIDENCE

A. SQL Server: Predicting and Mitigating Blocking Chains

The most typical operational failure of SQL Server is the appearance of blocking chains, in which a transaction that is holding locks is preventing the rest of the transactions to run and the waits spread out to stalls throughout the entire system. Dynamic Management Views (DMVs), extended events and ad-hoc diagnostics are all common tools used by administrators to identify and fix blocking but these reactive means of identifying and fixing problems typically stop the problem occurring until after a long queue and user impact. The recent research indicates that the same methods of reinforcement learning, which were previously employed to investigate and simulate structured-query vulnerabilities, can be re-applied to model lock-wait graphs and learn the optimal interventions minimizing cumulative user impact [2]. An AI pipeline is in practice fed with transaction logs, learned policy is used to estimate the value of intervening with transaction options, including terminating a lead blocker, setting higher transaction priority, or letting a natural completion run under rate control. Explainable anomaly-detection systems render these recommendations operable to DBAs (they give clear reasons why such a query pattern or index useage led to escalation) and therefore more operators trust automated recommendations [21], [23]. SQL injection and query hygiene security literature supports the fact that structured-query dependence is a source of vulnerability and that modeling dependency structure is a high-value signal on which one can forecast and equally model-based mitigation [1], [3]. Lastly, since encryption and other performance-losing techniques can indirectly raise blocking risk, privacy-conscious or latency-conscious models are necessary where

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 2, Issue 1, May 2022
Impact Factor: 6.252

sensitive log information is not central and can be used to train models [4]. When combined, these developments allow SQL Server environments to move towards slow and manual mitigation to proactive, simulated recovery paths which reduce mean time to recovery (MTTR) and mitigate user impact.

B. Oracle RAC: Anticipating Node Evictions in Clustered Environments

Oracle Real Application Clusters (RAC) alter the failure environment: instead of the single-instance crashes, RAC systems are vulnerable to the node-eviction events, to global cache contention and split-brain issues of interconnect instability. Even though RAC increases availability because it eliminates the single points of failure, its distributed topology adds complex, distributed failure modes, which are not easily visible by traditional monitoring, which is threshold-based. Experimental research of Oracle cloud and migration projects highlights the fact that RAC-related glitches are common during version upgrades or misconfiguration and that there are significant differences in recovery patterns between single-instance databases and RAC-based ones [6], [10]. Analysis of costs and adoption also shows that the complexity of operations of RAC also leads to hidden costs of downtime compared to deploying a more straightforward PostgreSQL implementation [8], [9]. RAC AI methods consider the cluster as a graph of nodes whose telemetry contains information on interconnect latency, packet loss, and cache-fusion activity, and graph neural networks and distributed anomaly detectors are able to identify temporal patterns predicting evictions and provide interpretable features to administrators (such as augmented inter-node jitter or asymmetric latency spikes), to users [22]. Explainable AI methods also offer the reasoning behind the eviction prediction enabling DBAs to make decisions about mitigating inter-node traffic, rebalancing loads, or changing cluster heartbeat thresholds [21], [23]. These models can be used to mitigate the occurrence of avoidable node removals and the effect on availability by predicting the possibility of eviction and prescribing pre-emptive workload redistribution.

C. PostgreSQL: Forecasting Autovacuum Disruptions

The self-maintenance subsystem of PostgreSQL, which is used to prevent table bloat and transaction ID wraparound, is not without its own stability issue in that autovacuum processes may themselves become the cause of instability during peak workload periods. In comparison to sudden block chains or discrete eviction of RAC, autovacuum problems are normally found to rise over time: throughput bar, rise in I/O wait, and eventual spikes in latency. Experiments of postgreSQL on high-performance and cloud implementation indicate that autovacuum activity is highly dependent on workload distribution and hardware specifications, such that manual tuning is fragile and flawed in the context of diverse enterprise settings [11], [12]. Machine learning models formulate autovacuum as a prediction challenge: the forecasting time-series and anomaly detection models can predict high-risk periods when autovacuum will probably disrupt crucial queries based on historical concurrent workload, query mix, and rate of churning of the tuples as well as I/O measures. Event-driven control and adaptive fuzzy models are used in methods to balance conflicting goals of the timely vacuuming and minimum disturbance [14]. Elucidable anomaly frameworks once more come into play: DBAs must have explicit explanations (e.g. which tables or vacuum thresholds are causing the prediction) prior to embracing automated rescheduling or threshold modifications [21]. Deployments based on predictive autovacuum scheduling are characterized by lower rates of emergency maintenance as well as better stability in the long-term, since active maintenance scheduling prevents the build-up of bloat which would otherwise trigger serious and difficult to mitigate slowdowns.

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 2, Issue 1, May 2022
Impact Factor: 6.252

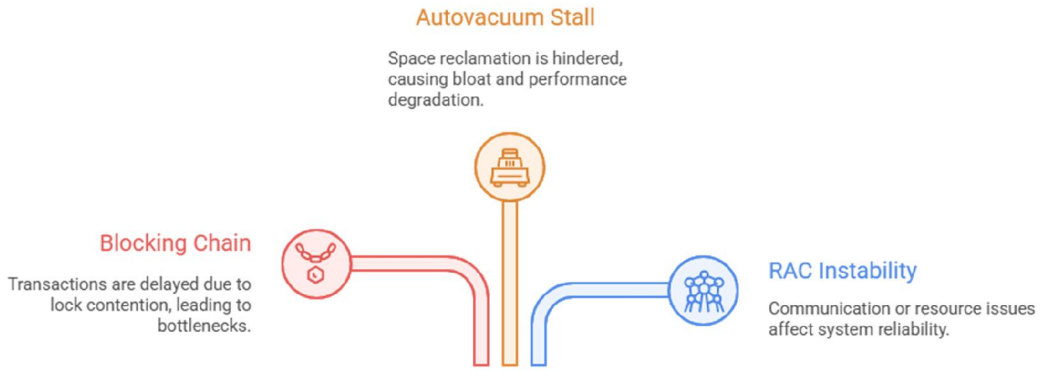

Fig. 2: Example of blocking chain vs. autovacuum stall vs. RAC instability.

D. Cross-Engine Lessons and Enterprise Insights

In the three engines, several cross cutting lessons come out. First, all DBMSs have a characteristic fragility of their failures, each having different fragility: transaction dependency fragility of SQL Server, cluster synchronization fragility of Oracle RAC, and maintenance-trigger fragility of PostgreSQL, although all three are predictive and explainable AI over existing telemetry. Second, interpretability is the determinant of AI utility: AI tools inspired by healthcare, finance, and IoT anomaly detection show that operators are much more likely to adopt models to which the model outputs can be mapped to the usual operational artifacts, i.e. wait statistics, interconnect latencies, or vacuum thresholds [21], [23], [25]. Third, cost and migration research points out that the downtime is a business metric; it has been found in studies comparing the adoption of Oracle and PostgreSQL that operational complexity and recovery patterns have a significantly negative impact on TCO and risk exposure and do not simply depend on raw performance metrics [8], [9], [10]. Fourth, security research teaches us that structure-query dependency patterns are a generic source of vulnerability, whether used maliciously or provoked by accident, and modeling dependency topology is a high-value approach that can be employed in engines [1], [2]. Lastly, a deployment on the ground exposes the following challenges that are subject to future research: privacy-preserving training (as logs can be sensitive) [4], cross-heterogeneous workload transferability, and integration with cloud-native telemetry (where low-level metrics can be abstracted away) [6], [7]. The results of the mentioned enterprises are quantifiable decreases in MTTR and a greater correspondence amid the operational practice and business continuity goals.

VI. DISCUSSION

A. Operational Implications for DBAs and SREs

Predictive recovery models transform operational duties of database administrators (DBAs) and site reliability engineers (SREs) into more of proactive manager of automated interventions rather than more of reactive troubleshooters. Practically, this transition would necessitate novel procedures: model-driven playbooks that project forecasted categories of failures to prioritarian actions, under-escalation guidelines that entrench business-impact limits, and closer collaboration between DBAs and platform teams to take on preemptive recommendations. Those enterprises that implement AI-driven recovery are likely to see a decrease in mean time to recovery (MTTR) when models are reliable to reveal early signs of blocking chains, RAC instabilities or autovacuum collisions; but to achieve such benefits, such interventions must be carefully synchronized such that automated undertakings (e.g., stopping a blocking session) do not create more downstream risk. Embedding automated validation into CI/CD pipelines has taught that in order to institutionalize automated checks, there must be strong gating and rollback processes, developer and operator training to interpret the outputs of a model in the correct way [5]. The operationalization of predictive recovery thus requires both tooling and human-process redesigning especially in those organizations that have stringent SLAs or regulatory restrictions [6], [10], [12].

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429 Volume 2, Issue 1, May 2022 Impact Factor: 6.252

B. Trust, Explainability, and Governance

Explainability of models is the key to adoption. DBAs will not leave control to black-box systems without the model can flesh out its suggestions in a language that can be translated to common operational attributes- lock waits, interconnect jitter or vacuum thresholds. Explainable AI (XAI) methods alleviate cognitive friction by making available feature attributions and brevity of justification on why their predictions are made, allowing operators to confirm their recommended course of action may be implemented [21], [23]. The structure of governance systems should then mandate explainability measures and set guidelines of automated versus human-mediated activities. Moreover, audit trails of model inputs, outputs and the selected interventions are needed to review and comply with regulations after the incident and financial and healthcare deployments show how interpretability and auditability pattern affect trust and legal defensibility [24], [25]. The absence of the XAI and governance will lead to the fact that predictive systems will not be used as much as possible or, at worst, they will lead to wrong automated interventions, which will ruin confidence.

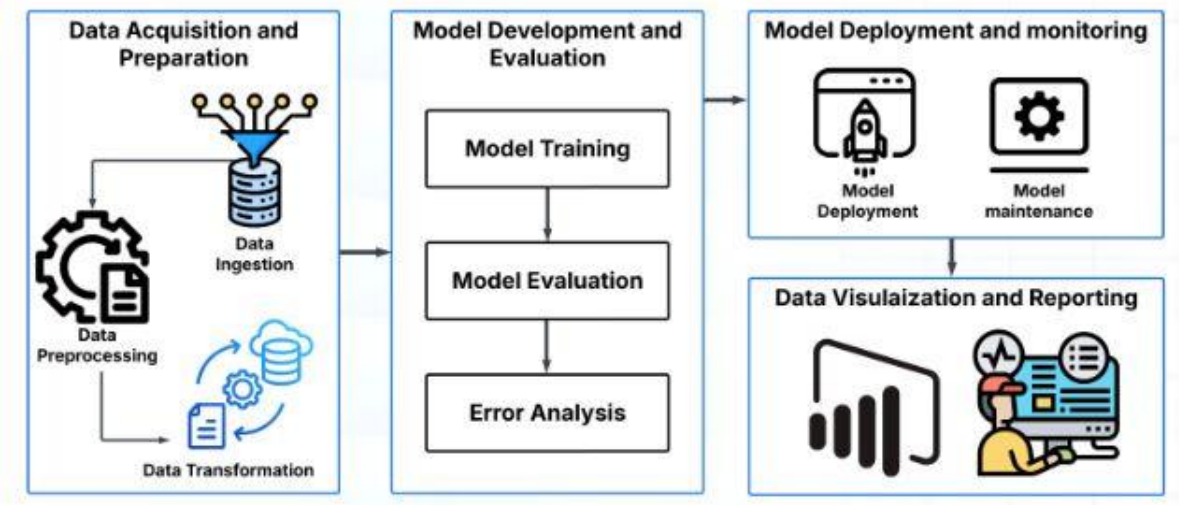

Fig. 3:AI-driven predictive recovery pipeline. **Source:** Xenonstack

C. Privacy, Data Sensitivity, and Federated Approaches

The extensive telemetry is needed to train powerful models, although the logs of databases and the traces of transactions usually have sensitive data. Exposure can be minimised by using approaches like operating on aggregated signatures (hashes, counts), and training models using privacy-preserving training methods federated learning or differential privacy, can be used to train models across organisations without sharing raw logs [4]. The federated ones can enhance the generalization of the model, pooling the different workload patterns and maintaining confidentiality, but also come with the issues of heterogeneity of the telemetry schemas and labeling stability. Privacy-preserving model architectures should be adopted by enterprise, therefore, coupled with rigorous data governance, such that feature extraction pipelines are redacted or abstract sensitive fields, and model updates are versioned and auditable.

D. Integration with Cloud-Native Observability and Vendor Constraints

Cloud platforms offer managed database services that obscure several low-level measures, which complicate telemetry gathering needed to gather finely grained predictive models. Oracle Cloud and AWS RDS based services and managed postgreSQL have varying levels of observability and controllability, and Oracle-specific measurements are not potentially implemented in many managed environments [6], [7], [12]. To address such inconsistency, there must be flexible layers of instrumentation that normalize vendor specific telemetry to a set of features that is standardized. Also, cloud operational semantics (autoscaling) and ephemeral inherited instancings could change the failure behaviour i.e. what appeared to be an autovacuum stall on a fixed node may be different with horizontal autoscaling, and prediction

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429 Volume 2, Issue 1, May 2022 Impact Factor: 6.252

models must be retrained or configured to autoscaling on the cloud [8], [11]. Practically, integration work could replace initial deployments: connector development, metric alignment, and the need to make sure that abstractions of vendors do not hide the very signals that models are made of.

E. Limitations, Robustness, and Adversarial Considerations

Decision models are inherently limited. The heterogeneity of workloads results in dataset shift which deteriorates model performance upon cross-enterprise transfer; therefore, model drift monitoring and online retraining should be monitored. Models can also be susceptible to adversarial or confounding signals: an attacker can use telemetry to induce unnecessary interventions, or synthetic load testing can be interpreted as indicators of failure, etc.—similarities to RL-based security studies also hold significance, and furthermore advisory adversarial-resistant model design approaches can be required [2]. Overfitting to particular environments is minimizable by extensively cross-domain transfer learning and benchmarking across various deployments, although these needs curated datasets collections and anonymization as well as common evaluation metrics. Lastly, over-automation may occur: automated recovery without conservative safety checks may result in loss of data or inconsistent states, particularly in clustered systems such as RAC where the actions of one node may spread to other nodes; hence, conservative prioritisation of actions (those with low risk should be done first) and fallback human-in-the-loop gates are still important.

F. Future Research Directions and Standardization Needs

A number of research paths should be given priority in order to scale up AI-powered recovery. To begin with, uniform benchmarks and corpus of database failure instances labelled would allow strict comparison of models between engines and workloads; this is similar to calls in other areas to common datasets that spur development. Second, the research on federated and transfer-learning methods should be conducted to apply cross-enterprise model generalization and maintain privacy [4], [21]. Third, it may be safer and easier to understand hybrid control systems incorporating both rule-based defenses and ML predictions (a guardrail + predictor pattern) as shown by fuzzy-control and decision-theoretic supply-chain research systems can provide an attractive way to balance conflicting goals [14]. Fourth, the systematic investigation of human-AI interaction during DBA workflow, the level of trust, response time, and error incidence, would be used to guide interface and explanation design [23], [25]. Lastly, more generalization is required: standard telemetry schemas of lock graphs, RAC interconnects and vacuum measurements would make models portability and easier to integrate with other ecosystems.

VII. CONCLUSION

The paper has analyzed the failures of the three common relational database engines, i.e. SQL Server, Oracle RAC, and PostgreSQL, with the focus on blocking chains, eviction risks, and autovacuum disturbances as failure personalities, respectively. We identified the unique weaknesses of each system through a discussion of design patterns and empirical data, but also discussed how AI-based anomaly detection and prediction of recovery trajectories can be used to advantage in each system. It was pointed out that the discussion has operational implications on the side of DBAs and SREs, the need of trust and explainability in deploying models, and the limitations of privacy, cloud-native observability and adversarial conditions. Predictive recovery is a promising solution; however, its implementation should be controlled with governance, conservative automation, and human control. Future studies comprise federated learning methods, prediction with guardrail hybrid models, standardized telemetry models, and benchmarks of systematic evaluation. Finally, AI-based recovery provides business organisations with a revolutionary chance to minimise downtime and match database resilience with strategic resilience objectives, as long as its implementation is both an innovative and a sensible investment.

REFERENCES

[1]. Abououf, M., Singh, S., Mizouni, R., &Otrok, H. (2024). Explainable AI for Event and Anomaly Detection and Classification in Healthcare Monitoring Systems. IEEE Internet of Things Journal, 11(2), 3446–3457. https://doi.org/10.1109/JIOT.2023.3296809

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 2, Issue 1, May 2022
Impact Factor: 6.252

[2]. Anadolu, M. N., Sun, J., Li, J. T. Y., Graber, T. E., Ortega, J., &Sossin, W. S. (2024). Puromycin reveals a distinct conformation of neuronal ribosomes. Proceedings of the National Academy of Sciences of the United States of America, 121(7). https://doi.org/10.1073/pnas.2306993121

[3]. Biewenga, L., Vermathen, R., Rosier, B. J. H. M., &Merkx, M. (2024). A Generic Antibody-Blocking Protein That Enables pH-Switchable Activation of Antibody Activity. ACS Chemical Biology, 19(1), 48–57. https://doi.org/10.1021/acschembio.3c00449

[4]. Fadlil, A., Riadi, I., &Mu'Min, M. A. (2024). Mitigation from SQL Injection Attacks on Web Server using Open Web Application Security Project Framework. International Journal of Engineering, Transactions A: Basics, 37(4), 635–645. https://doi.org/10.5829/ije.2024.37.04a.06

[5]. Haikal Muhammad, H., Id Hadiana, A., &Ashaury, H. (2024). PENGAMANAN APLIKASI WEB DARI SERANGAN SQL INJECTION DAN CROSS SITE SCRIPTING MENGGUNAKAN WEB APPLICATION FIREWALL. JATI (JurnalMahasiswaTeknikInformatika), 7(5), 3265–3273. https://doi.org/10.36040/jati.v7i5.7320

[6]. Han, J., & Choi, Y. (2024). Analyzing Performance Characteristics of PostgreSQL and MariaDB on NVMeVirt. arXiv preprint arXiv:2411.10005. https://doi.org/10.48550/arXiv.2411.10005

[7]. Huang, N., Yu, Y., Shao, Y., & Zhang, J. (2024). Numerical Simulation of Falling-Snow Deposition Pattern Over 3D-Hill. Journal of Geophysical Research: Atmospheres, 129(2). https://doi.org/10.1029/2023JD039898

[8]. Kavuluri, H. V. R. (2024). Oracle vsPostgreSQL: Cost Analysis for Enterprises Adopting Cloud-Native Strategies. International Journal of Emerging Research in Engineering and Technology, 5(3), 22-31. https://doi.org/10.63282/3050-922X.IJERET-V5I3P103

[9]. Kavuluri, H. V. R. (2024). PostgreSQL vs. Oracle: A Comparative Study of Performance, Scalability, and Enterprise Adoption. International Journal of Emerging Trends in Computer Science and Information Technology, 5(2), 33-41. https://doi.org/10.63282/3050-9246.IJETCSIT-V5I2P104

[10]. Khosravi, P., Mohammadi, S., Zahiri, F., Khodarahmi, M., &Zahiri, J. (2024, December 1). AI-Enhanced Detection of Clinically Relevant Structural and Functional Anomalies in MRI: Traversing the Landscape of Conventional to Explainable Approaches. Journal of Magnetic Resonance Imaging. John Wiley and Sons Inc. https://doi.org/10.1002/jmri.29247

[11]. Kim, M., Yoon, H. J., Lee, C., Lee, M., Park, R. W., Lee, B., … Kim, S. (2024). Immune Checkpoint-Blocking Nanocages Cross the Blood-Brain Barrier and Impede Brain Tumor Growth. ACS Biomaterials Science and Engineering, 10(1), 575–587. https://doi.org/10.1021/acsbiomaterials.3c01200

[12]. Lisdorf, A. (2024). Oracle. In Grundlagen des Cloud Computing: EinenichttechnischeEinführung (pp. 65-75). Berkeley, CA: Apress. https://doi.org/10.1007/979-8-8688-0089-4_5

[13]. Nishio, K., Pasquet, L., Camara, K., DiSapio, J., Hsu, K. S., Kato, S., …Berzofsky, J. A. (2024). Lysosomal processing of sulfatide analogs alters target NKT cell specificity and immune responses in cancer. Journal of Clinical Investigation, 134(4). https://doi.org/10.1172/JCI165281

[14]. Oloruntoba, O., &Omolayo, O. (2024). Unlocking Performance and Uptime: A Strategic Approach to Oracle 12c to 19c Migration for Maximizing ROI. This paper is an original technical whitepaper completed in March. https://dx.doi.org/10.2139/ssrn.5267801

[15]. OlubusolaOdeyemi, NoluthandoZamanjomaneMhlongo, EkeneEzinwaNwankwo, &Oluwatobi Timothy Soyombo. (2024). Reviewing the role of AI in fraud detection and prevention in financial services. International Journal of Science and Research Archive, 11(1), 2101–2110. https://doi.org/10.30574/ijsra.2024.11.1.0279

[16]. Redcenko, O., Tumova, M., &Draber, P. (2024). Simplified PCR-Based Quantification of Proteins with DNA Aptamers and Methylcellulose as a Blocking Agent. International Journal of Molecular Sciences, 25(1). https://doi.org/10.3390/ijms25010347

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 2, Issue 1, May 2022
Impact Factor: 6.252

[17]. Sabharwal, R., Miah, S. J., Wamba, S. F., & Cook, P. (2024). Extending application of explainable artificial intelligence for managers in financial organizations. Annals of Operations Research. https://doi.org/10.1007/s10479-024-05825-9

[18]. Sommervoll, Å. Å., Erdődi, L., &Zennaro, F. M. (2024). Simulating all archetypes of SQL injection vulnerability exploitation using reinforcement learning agents. International Journal of Information Security, 23(1), 225–246. https://doi.org/10.1007/s10207-023-00738-3

[19]. Sun, B., Zhao, S., &Tian, G. (2024). SQL queries over encrypted databases: a survey. Connection Science, 36(1). https://doi.org/10.1080/09540091.2024.2323059

[20]. Vadlamani, V. (2024). Introduction to PostgreSQL Database Management. In PostgreSQL Skills Development on Cloud: A Practical Guide to Database Management with AWS and Azure (pp. 1-39). Berkeley, CA: Apress. https://doi.org/10.1007/979-8-8688-0817-3_1

[21]. Wang, W., Abbasi, O., Yanikomeroglu, H., Liang, C., Tang, L., & Chen, Q. (2024). VHetNets for AI and AI for VHetNets: An Anomaly Detection Case Study for Ubiquitous IoT. IEEE Network, 38(6), 170–177. https://doi.org/10.1109/MNET.2023.3349309

[22]. Wang, Z., Li, J., Wang, L. F., Liu, Y., Wang, W., Chen, J. Y., … Zhu, S. L. (2024). FFAR4 activation inhibits lung adenocarcinoma via blocking respiratory chain complex assembly associated mitochondrial metabolism. Cellular and Molecular Biology Letters, 29(1). https://doi.org/10.1186/s11658-024-00535-3

[23]. Wei, Z., Liu, Y., Wu, Y., Chen, W., & Li, Q. K. (2024). T-S fuzzy model based event-triggered change control for product and supply chain systems. International Journal of Systems Science, 55(3), 426–439. https://doi.org/10.1080/00207721.2023.2272302

[24]. Xu, Q., & Deng, H. (2024). Research on education management system based on machine learning and multidimensional data modeling. Applied Mathematics and Nonlinear Sciences, 9(1). https://doi.org/10.2478/amns.2023.1.00072

[25]. Yadav, S. (2025). Disaster Recovery and High Availability Strategies in Oracle Cloud Infrastructure. IJSAT-International Journal on Science and Technology, 16(2). https://doi.org/10.71097/IJSAT.v16.i2.3065