ISSN: 2093-5374 / E-ISSN: 2093-5382

AI-Driven Data Loss Prevention in Oracle and PostgreSQL Intelligent Monitoring for PII and FTI Protection

Harsha Vardhan Reddy Kavuluri[1*]

[1*]WISSEN Infotech INC, Wisconsin, United States. kavuluri99@gmail.com,
https://orcid.org/0009-0005-6003-6464

Received: February 05, 2026; Revised: March 12, 2026; Accepted: April 30, 2026; Published: June 30, 2026

Abstract

Enterprise database systems deal with large amounts of sensitive organizational information and, therefore, have to have powerful monitoring and protection policies to avoid unauthorized access, insider abuse, and possible data breaches. The conventional methods of monitoring databases are based on the traditional rule and preset thresholds, which restricts their powers in detecting complex anomalous queries as well as dynamic insider threats in the current business environment. Recent developments in machine learning and behavioral analytics showed that the capabilities to identify abnormal database operations have been improved, but most of the current methods do not have adaptive learning capabilities and support heterogeneous databases across platforms. This research study suggests an artificial intelligence-based monitoring system combining both database activity monitoring and behavioral anomaly detection to improve data protection of enterprises. The suggested framework is the systematic analysis of the activity logs of the database, detection of sensitive data access patterns, and the assessment of the levels of risk associated with queries with intelligent detection models. A behavior analysis module obtains the regular patterns of user interaction and identifies the unusual behavior, which can suggest the appearance of some suspicious or unauthorized actions. The framework was tested and assessed with database activity logs on the enterprise level provided on the Oracle and PostgreSQL database environments and were based on millions of query records reflecting the actual operational loads. In experimental analysis, it has been shown that the suggested monitoring framework is much better in detecting anomalies than legacy rule-based monitoring systems. The system reached about 92% detection, 88% anomaly recall and 90% precision in detecting suspicious database queries as well as abnormal access pattern. The framework effectively identified sensitive data access attempts and potential data leakage scenarios while maintaining low false-positive rates during high-volume database operations. The results indicate that integrating artificial intelligence with behavioral analytics substantially strengthens enterprise database monitoring capabilities and provides a scalable solution for protecting sensitive organizational data in complex database infrastructures.

Keywords: Database Security, Anomaly Detection, Database Activity Monitoring, Artificial Intelligence, Behavioral Monitoring, Enterprise Data Protection.

1 Introduction

The rapid development of enterprise information systems has resulted in a simultaneous increase in the amount of sensitive data captured by those systems and stored in relational database infrastructures. Database systems such as Oracle and PostgreSQL are central to most modern organizations and are used

Journal of Wireless Mobile Networks, Ubiquitous Computing, and Dependable Applications (JoWUA), volume: 17, number: 2 (June - 2026), pp. 73-101. DOI: *10.58346/JOWUA.2026.I2.005*

*Corresponding author: WISSEN Infotech INC, Wisconsin, United States.

to record operations, manage financial transactions, store customer data, and maintain data subject to regulatory control. These database systems are also used to process large volumes of sensitive data, including Personally Identifiable Information (PII) and Federal Tax Information (FTI), and as a consequence the security of databases has become a primary focus of an organization's cyber security efforts (Islam, 2024). The continued growth of digital services that are powered by databases means that an organization's ability to describe and implement relevant monitoring and control systems will need to keep pace with the sophistication of the unauthorized access to sensitive datasets.

Relational database management systems (RDBMS) provide the foundational technology for enterprise applications in numerous industries, including healthcare, banking, and government. RDBMS handle massive amounts of structured data, which is accessed via SQL and utilized in enterprise information systems by both applications and users. With the rapid implementation of distributed database systems and cloud data storage, the ability to assess and mitigate risks associated with the exposure of sensitive information has rapidly increased in difficulty (Karri, 2021), and the continuous generation of massive quantities of query logs and operational telemetry data in database enterprise environments has obligated the implementation of continuous processes to protect sensitive information from unauthorized disclosures (Fadolalkarim et al., 2020; Bhat, & Jayaram, 2025). The growing complexity of enterprise databases has made database monitoring a requisite for enterprise security systems.

Cyber security incidents involving sensitive information are more frequently occurring in enterprise database systems where access to sensitive data is obtained through valid SQL queries. Malicious users masquerade as performing typical database activities by using valid database operation privileges to obtain sensitive data. Insider attacks, compromised access credentials, and abnormal patterns of database query execution are a few of the primary architecture flaws that contribute to data breaches associated with database systems (Balogun et al., 2025). These types of attacks are very difficult to detect as the malicious queries may look very similar to normal operational queries.

Security practices using conventional monitoring systems are traditionally rule-based systems in which a user's access and database operational restrictions are determined by static policies. While these systems serve as a baseline for security, still leave the systems vulnerable to attacks as the strategies are not sophisticated. Because static systems are designed for static systems, unable to accommodate changes in systems and behaviors dynamically. The nature of the database workloads has made the behavioral anomalies increasingly complex, to the point where the inadequacies of rule-based systems in monitoring behaviors of the enterprise databases have become paramount in studies (Le & Zincir-Heywood, 2021). Thus, there is a necessity for adaptive enterprise database monitoring systems to identify the anomalous behavior of queries.

In the last few decades of database monitoring, Artificial Intelligence (AI) and machine learning have emerged as impactful improvements. Their use has revolutionized behavior pattern analysis in large databases. Taking into consideration the context of the behavior, such as, purposes of queries, temporal trends of queries, and behavioral activity of users, AI has been able to detect activities that have not been observed. This has been the overwhelming justification for the application of machine learning in enterprise databases. Similar advances in machine learning have demonstrated strong capability in extracting patterns from complex datasets; for example, predictive modeling approaches using Random Forest algorithms have successfully identified hidden relationships in multidimensional datasets and achieved high predictive accuracy in toxicity modeling tasks (Pérez Santín et al., 2021).

Several studies have attempted to explore machine learning methods for the monitoring of database activities and the discovery of anomalies. Such works have attempted behavioral profiling, various

clusters, and various deep learning techniques to tackle the analyses of SQL queries and workloads to discover unusual patterns of executions. Such intelligent monitoring systems learn a baseline of query behavioral patterns from historical workloads and identify deviations from the norm that signal suspicion (Kotenko & Saenko, 2022). These methods are a considerable improvement on active database monitoring compared to the usual rule-based monitoring systems. In parallel, the integration of neural networks into database system components has also been explored, where neural network–based indexing mechanisms learn mappings between key spaces and storage structures to significantly improve query response times and data retrieval efficiency in large-scale databases (Sappa, 2025).

However, there is a significant amount of room for advancement in regards to the monitoring systems that are driven by machine learning. Current generation systems are often designed to monitor a single database, and are not designed to monitor a plurality of databases. These situations contribute to the difficulty of monitoring access to sensitive data. Existing measures for database security are not designed to handle the problem posed by the integration of multiple relational databases - for example, the inclusion of both Oracle and PostgreSQL databases. Monitoring access to sensitive data across multiple databases poses a significant challenge for database security frameworks (Oloruntoba, 2025). This is due to the need for a monitoring framework that is capable of being applied to multiple database systems for the purposes of monitoring access to sensitive data.

Another notable limitation of most current monitoring methods is their inability to detect queries that target highly sensitive attributes of databases. Numerous anomaly detection methods examine the structure of SQL queries but ignore the sensitivity of the data fields in question (Gajula, 2023; Edozie et al., 2025). Monitoring methods that do not consider sensitive data classification may ignore the most dangerous database activities involving sensitive data, such as PII and FTI (Herath et al., 2024). Therefore, research within the framework of enterprise data governance has highlighted the necessity of developing methods of sensitive data classification for monitoring databases to enhance the identification of risks of data leaks (Ponde et al., 2022). This sensitive data classification method for monitoring systems enables the assessment of both the query activity and the sensitive nature of the query.

Enterprise database activity on a larger scale continues to increase challenges for security monitoring tasks. Large enterprise databases produce millions of database accesses and SQL queries, resulting in billions of operational telemetry events which must be analysed in real time for telemetry operational. Manual monitoring activity logs is impractical and often results in greater latencies for detecting anomalous activities (Islam et al., 2025). For database enterprise monitoring systems, automated anomaly detection combined with data classification by level of sensitivity is a very promising area of research (Goswami, 2024). These intelligent monitoring systems provide various analytic capabilities for large scale database telemetry to organizations. Similar to how hybrid engineering systems combine multiple energy sources to improve performance and operational efficiency in complex environments, integrated monitoring architectures can combine multiple analytical mechanisms to enhance detection capabilities in large-scale systems (Fascista, 2022).

The main issue the research addresses is the abnormal query behavior of sensitive data in heterogeneous environments of relational databases and how to effectively handle it. It is often the case that large organizations contain sensitive data dispersed over several platforms of relational databases, which increases the difficulties of the monitoring systems that are built to detect possible data leaks and data value losses. To engage in this activity, it is necessary to have monitoring systems that can detect, analyze, and understand the complexity of the query and the data value at the same time. Addressing

this challenge requires the development of intelligent monitoring architectures that can operate across heterogeneous database systems.

Data breaches that lead to the unauthorized acquisition of sensitive information lead to serious financial losses, harm to the organization's reputation, and legal liability. Organizations that experience data breaches that involve sensitive, personally identifiable information, as well as data breaches that involve sensitive tax information, legally liable to the affected customers and lose their trust. Therefore, organizations that experience such data breaches need to have effective monitoring systems that help to identify potential data breaches and smaller scales rather than only reacting after large-scale data breaches. To solve these problems, this study proposes a framework for data loss prevention that is AI-based and is specifically designed to recognize irregular database access in the form of anomalous behavior. The framework is based on the combination of intelligent analysis of data access, behavior modeling, and data leakage threats.

The rest of this study is structured in the following way. Part 2 is the review of the literature on the topic of SQL optimization and AI-based database management systems. Section 3 discusses the suggested AI-based SQL optimization model and architecture. Section 4 is a description of the experimental setup and implementation. The fifth section is performance evaluation and results of comparative analysis. Lastly, Section 6 provides a conclusion of the paper and gives the future research direction.

Research Objectives

- To build a framework of artificial intelligence monitoring that will analyze the records of activity in the enterprise database and detect suspicious querying behavior on-site.
- To incorporate the behavioral anomaly detection methods of detecting deviations in user access patterns and unauthorized access to sensitive database fields.
- To establish a risk scoring system that can rate database queries in terms of access frequency, query complexity, and access privileges to establish possible security risks.
- To assess the functionality of the suggested monitoring system with the help of the enterprise-scale data stored in Oracle database as well as PostgreSQL settings.
- To enhance enterprise database security and governance through facilitating early associated activities, privilege escalation efforts and massive unauthorized data access.

2 Methodology

Enterprise database systems produce a significant volume of activity logs, capturing detailed records of SQL transactions, user activity, and data access. These records reveal user interactions with the database and how access sensitive data. The monitoring framework proposed uses the activity records to detect how the database ordinarily operates and define its abnormal query characteristics to detect possible data leaks. The framework operation involves the continuous querying and logging of access events from Oracle and PostgreSQL databases, which constructs a database usage behavior profile for further analysis with intelligent detection systems.

The framework’s activity logs are first compiled in a structured data processing pipeline, where categorized, and the behavioral features of query characteristics, user roles, access frequency, and data sensitivity are extracted. These features enable the monitoring framework to distinguish between

operational queries and unusual database interactions that implicate sensitive PII or FTI data. Behavioral models are then imposed to examine the behavior and to detect anomalies in access to the database. In the presence of abnormal activity, the framework determines the level of risk and generates alerts, which provide a basis for the data breach preventative measures to the database administrators.

In figure 1 illustrates the AI-driven data loss prevention architecture where activity logs from Oracle and PostgreSQL databases are continuously collected, processed, and transformed into behavioral features. These features are analysed using intelligent anomaly detection models to identify suspicious access patterns and trigger real-time alerts for potential data breaches.

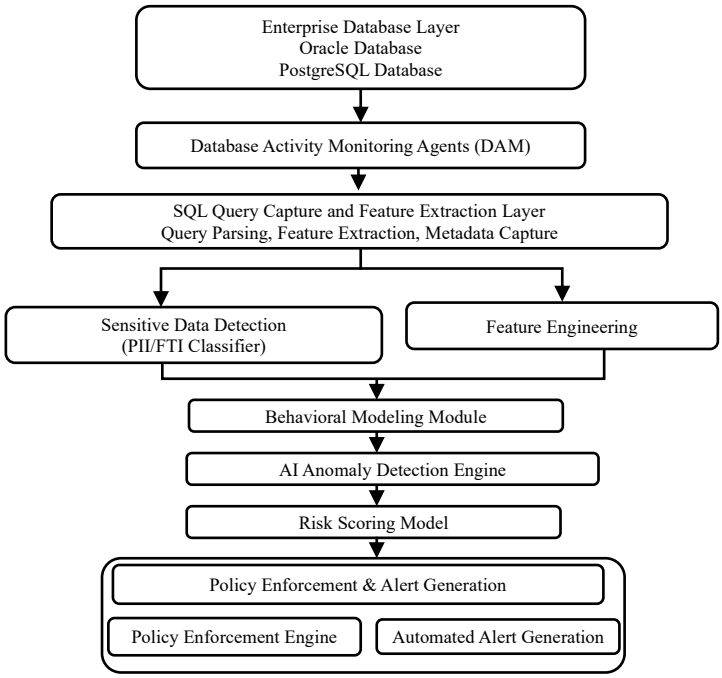

Figure 1: AI-Driven data loss prevention architecture for oracle and postgresql databases

2.1 Database Activity Collection and Monitoring Architecture

Through SQL operations, users and automated processes of the system, enterprise relational databases are able to produce operational data at a mass level. For every database activity, activity records are created, which include query logs, session IDs, timestamps, and metadata for the database objects involved in the activity. These activity records shed light on how database resources are utilized and how users access sensitive data within the system. Recent works focusing on database intelligent monitoring have shown that the abnormal activities and potential leaks in data are more easily identified when there is an active and ongoing examination of the database activity logs (Reddy et al., 2025). These records are utilized as the primary telemetry source for the monitoring system and operational behavior in Oracle and PostgreSQL systems is explained.

The monitoring process begins with installation of the activity monitoring agents which are linked with the database servers. These agents fetch SQL queries from the database communication channels and fetch contextual details like the user ID, session ID, the tables accessed and the time the query was executed. Monitoring systems using machine learning need activity data like this to establish a baseline of normal database usage and to detect abnormal queries (Reddy et al., 2025). Real-time collection and

monitoring of this kind, allows the security monitoring system to assess each interaction with the database.

The monitoring agents, after intercepting queries send the activity logs have collected, to a centralized layer of telemetry aggregation. This layer integrates monitoring data from different database systems used within the organization's infrastructure. Database research shows that collection of query logs in a centralized database system, aids the security system in detecting large scale behavioral patterns that cannot be detected when the logs are isolated (Boakye et al., 2023). Therefore, the telemetry aggregation from Oracle and PostgreSQL systems allows for integrated monitoring of disparate database systems.

The layer that handles telemetry aggregation must also include normalization of the logs that have been captured as various database systems record activity logs in different formats. The monitoring framework handles the conversion of this heterogeneous logging structure into one uniform structure as defined by the fields of an SQL statement, the execution time, identifiers of user, table access, and session information. As discussed in the recent studies on the intelligent processing of logs for the detection of anomalies, the normalization of logs is very important for the production of structured datasets as the machine learning process is dependent on this normalization (Ryciak et al., 2022). By normalizing the logs, the bottom analytical modules gain the ability to rationalize the activities of the database uniformly, regardless of the platform.

Once the logs have been normalized, the SQL queries then undergo preprocessing. Behavioral attributes associated with each query are extracted and analyzed to capture the essence of the interaction with the DB. Attributes of the interaction include SQL operation type, number of joins in the query, tables involved, and user access level. Extraction of features from logs is the basis of behavioral modeling in AI monitoring systems, and is critical to establishing a structured representation of logs to monitor and identify aberrant patterns (De la Cruz Cabello et al., 2025).

To describe how monitoring agents capture the events of an activity of an event database activity, Algorithm 1 illustrate a core function of this monitoring component. The monitoring agent listens to the execution of an individual SQL statement by the database connection and collects relevant metadata about each individual query. The same monitoring pipelines are found in real-time AI-based security system frameworks that aim to identify irregular patterns of database accesses (Ejeofobiri et al., 2025).

Algorithm 1. SQL Query Capture and Monitoring Agent

```
def monitor_database_activity(connection):
    while connection.active():
        query_event = connection.capture_query()
        user_id = query_event.user
        query_text = query_event.sql
        timestamp = query_event.time
        accessed_tables = parse_tables(query_text)
        log_record = {
            "user": user_id,
            "query": query_text,
            "tables": accessed_tables,
```

```
    "time": timestamp
}
send_to_telemetry_pipeline(log_record)
```

Alongside single query instances, the monitoring system collects behavior data at the session level. Behavior data includes recorded and analysed sessions of the query sequence and other multiplayer queries, including the number of accesses made to the database and user interaction duration with the database. In the field of database security, session behavior data of this kind has been employed in the identification of insider attacks and atypical access patterns that create suspicion of possible data leaks (Md Imran et al., 2025). Collected activity data is transmitted to a database activity history central telemetry repository. The monitoring system is designed to detect changes in the database usage patterns, and, over the long term, to create usage behavior profiles that capture the usual operational pattern by analysing the telemetry data and activity logs. These profiles are used to detect and respond to the changes in access pattern behavior.

In figure 2 displays the operational flow of the monitoring framework, outlining the steps of processing the captured database telemetry. It focuses on the database monitoring agents collecting activity events and how pass through the telemetry ingestion pipeline, where logs from queries are homogenized and transformed into a single document suitable for analysis. It also flows through the capture of certain key constituents from the processed queries upon execution of the SQL parsing and metadata extraction, such as the session context, databases, the objects referenced, and the execution profile. The processed records are reformatted into classified datasets of the behavior of queries, which constitute the input of the subsequent modules of behavioral modeling and anomaly detection, which are intended to capture anomalous accesses to sensitive PII and FTI.

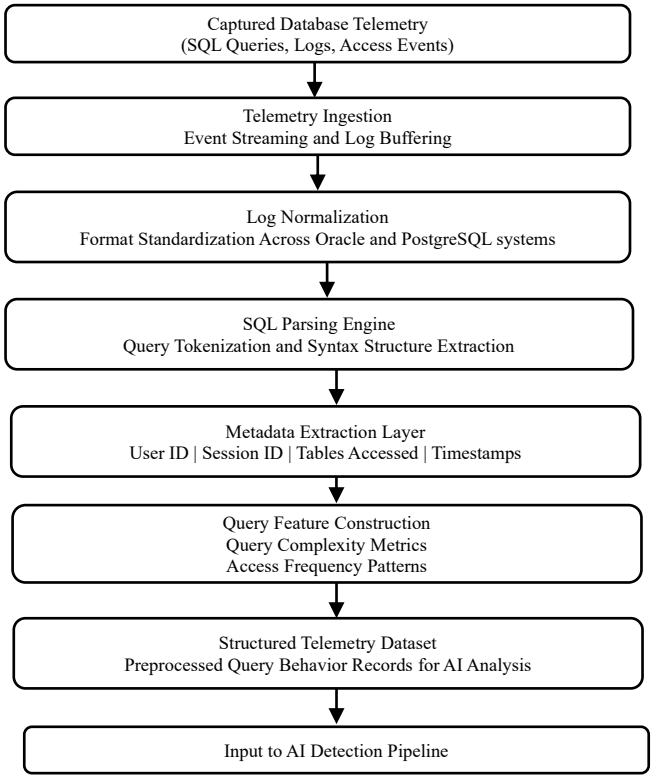

Figure 2: Workflow of AI-Based database activity monitoring and query capture

2.2 Sensitive Data Identification for PII and FTI

Safeguarding sensitive data in enterprise databases demands the capabilities of distinguishing elements that consist of protected data. The PII and FTI details reside in the relational database tables and are intermixed with other non-sensitive details. Hence, the monitoring framework initiates with a review of the database metadata to find out the columns with sensitive data, proposing further levels of detection and protection. These metadata-driven classification methods have appeared in a plethora of the latest research studies of database security to automate the process of detecting sensitive attributes in relation to database schema and contextual parameters (Lee et al., 2025).

The first phase of the sensitive data identification procedure includes scanning database schemas for structural metadata. Each database table comprises column specifications, attribute titles, and data types. Together, these elements serve as clues regarding the nature of the underlying data. Specifically, column names like email, social_security_number, tax_id, or address are indicators of PII or FTI fields. Tools for automated schema analysis can evaluate specific metadata characteristics to evaluate the presence of sensitive data (Niaz et al., 2025). Such data first-pass assessments help the tracking system to identify database elements requiring substantial semantic analysis.

Each column can be represented as a structured data element containing schema information with respect to monitoring system representations of database attributes. The attribute characterization for the tracking system can be described as equation 1.

$$A_i = \{C_i, T_i, M_i\} \tag{1}$$

Where A_i represents the i^{th} database attribute, C_i column name, T_i data type of the attribute, and M_i additional metadata stemming from the schema. This characterization helps the monitoring system arrange database elements in a structured manner suitable for sensitivity classification.

After establishing the attribute structure, the monitoring framework analyzes the semantic indicators relative to each attribute. The semantic indicators include patterns in the naming of columns, the accompanying metadata tags, and descriptions of text columns. These indicators are assessed against a set dictionary of known sensitive terms related to PII and FTI datasets. Recent studies on sensitive data discovery automation state that a combination of metadata indicators and semantic keyword matching provides the greatest success in the detection of sensitive attributes in relational databases (Adewole et al., 2024).

The monitoring framework then generates a sensitivity probability score for the attributes of each database based on the combination of semantic indicators and metadata tags. This probability score provides an estimate of the possibility that the given column possesses restricted data that requires monitoring. The sensitivity probability may be represented as equation 2.

$$P_s(A_i) = \frac{W_m M_i + W_c C_i + W_s S_i}{W_m + W_c + W_s} \tag{2}$$

Where $P_s(A_i)$ represents the sensitivity probability of attribute A_i, M_i is the metadata indicators score, C_i is the column-name similarity score, and S_i is the score for the matches on the semantics. The weights W_m, W_c, and W_s control the relative significance of each component for contribution to class in that specific role.

Numeric sensitivity scoring determines which attributes are to be considered protected, and therefore require closer scrutiny by the anomaly detection systems. Attributes are assigned high sensitivity scores when contain PII or FTI, such as financial records, tax ID numbers, personal addresses, or employee IDs. In intelligent data discovery, probabilistic scoring models are proven to distinguish sensitive data

attributes from operational data fields (Saleh, & Yasin, 2025). Table 1 summarizes the PII and FTI detection classification attributes. It includes sensitivity scores, semantic matching column name indicators, sensitivity keyword lists, and metadata tags from data architecture. From data architecture, these parameters allow the monitoring framework to detect protected data attributes automatically.

Table 1: Sensitive data classification parameters for PII and FTI detection

Parameter	Role in Classification	Description
Metadata Tags	Identifies sensitive field annotations	Schema metadata describing column attributes
Column Name Indicators	Detects potential PII/FTI attributes	Keywords appearing in column names (e.g., SSN, TaxID, Email)
Semantic Labels	Improves semantic classification accuracy	Contextual labels derived from data dictionaries
Sensitivity Probability Score	Determines monitoring priority	Calculated probability that an attribute contains protected data

The monitoring framework catalogs protected fields for each entity in the enterprise database after sensitive attributes have been located. This catalog enables the monitoring framework to observe which administrative users interact with sensitive fields and ensures that such engagements are routed to the behavioral monitoring pipeline. The framework offers a mechanism of determining PII and FTI attributes that need extra consideration in the Oracle and PostgreSQL database environments using a combination of semantic classifiers and metadata analytical tools.

2.3 Feature Engineering and Behavioral Modeling

After identifying the sensitive attributes, the monitoring system will convert the unstructured logs of SQL queries into a structured format and then fed into machine learning. Database activity logs generally contain SQL statements, a timestamp, a session ID, and metadata regarding the tables that were accessed. These raw logs cannot be used in behavioral analysis, so the monitoring system transforms the raw logs into a structured format of numerical attributes. This transformation increases the ability of monitoring systems to identify instances of abnormal querying behavior and patterns of database access that are suspicious in enterprise database environments (Bozdog et al., 2021).

The framework for monitoring accesses a number of operational variables for each SQL query event, including the kind of executed SQL command, the number of tables the query touches, the number of joins the SQL statement contains, the clearance of the user executing the command, and the classification of sensitivity of the data set the user has queried. Queries accessing Sensitive Personally Identifiable Information (PII) or Federal Tax Information (FTI) columns are subject to extra scrutiny as provide evidence of attempts to access sensitive data stored within the databases (Kanwal et al., 2024). The operational variables captured from the SQL queries offers a roughly approximate abstraction of the query activities that can be interpreted using a set of tools for anomaly detection.

In order to establish an abstraction of the information resources behavior, all SQL query events are modeled as feature vectors comprising the operational variables. The monitoring system models this behavior vector as equation 3.

$$F_q = [C_q, A_q, U_q, S_q] \qquad (3)$$

where F_q depicts the feature vector of the database query event, C_q represents the complexity score of the query, A_q represents the access frequency tied to the user session, U_q represents the clearance of the

user executing the command, and S_q represents the sensitivity score of the data set queried. The operational variables captured from the SQL queries offers an approximation of the query activities that can be interpreted using a set of tools for anomaly detection.

The behavioral modeling query complexity score is one of the most significant metrics. If a query contains a combination of multiple join operations, nested subqueries, or operates on a large table, it is likely an attempt to retrieve a large quantity of data from the database. Therefore, the complexity score allows the monitoring system to critically assess outlier structures of a query that are inconsistent with the majority of established usage patterns of a database (Yang et al., 2025). The complexity score of a query can be calculated by identifying certain structural features of an SQL expression. This can be expressed as equation 4:

$$C_q = \alpha J_q + \beta T_q + \gamma O_q \quad (4)$$

And J_q is the number of join operations, T_q is the number of tables involved and O_q is the number of SQL operations such as nested queries and aggregations. Coefficients α, β and γ determine the weight of each variable relative to the overall complexity of the query.

The frequency of access related to user sessions is another behavioral factor of importance. Database calls made frequently within a short period may indicate unnatural activity especially when the calls are in sensitive datasets. If the system has the ability to keep track of access behavior within a certain timeframe, it may sense a sudden activity increase and use it for possible data theft attempts (Nwachukwu et al., 2024). The access frequency metric can be expressed as equation 5.

$$A_q = \frac{N_q}{\Delta t} \quad (5)$$

Where N_q is the number of queries a user has made within a certain period of time, and Δt is that period of time.

Besides the structural features of the query and the temporal patterns of access, the monitoring system analyzes the sensitivity level of the data accessed. Queries involving sensitive attributes, particularly those that include PII or FTI, should weigh more in behavioral risk computing, as include more sensitive data. Express the dataset sensitivity score as equation 6.

$$S_q = \sum_{i=1}^{n} w_i \, d_i \quad (6)$$

Where d_i represents the query to a protected attribute and w_i a measure of sensitivity assigned to the attribute.

The integrated features create a complete behavioral manifestation of database conduct. These behavioral feature vectors are aggregated and evaluated over a period to capture a baseline of typical database interactions across varying users and applications. The system relates the emergent feature vectors to the behavior patterns it has acquired through the course of its lifetime in order to determine the presence of unusual interactions with the database, which could indicate unusual attempts to access Sensitive PII or FTI datasets.

2.4 AI-Based Anomaly Detection and Risk Scoring

After developing the feature vectors that define the database operations, the monitoring system would involve the use of machine learning models to identify the anomalous actions of the particular queries. These models are engaged to assess the behaviors and activities associated with the SQL query logs, and

determine the extent to which a query is outside the accepted parameters of use for a given database. Anomaly detection methodologies are a fixture of intelligent monitoring systems because enable a system to identify behavior that is considered anomalous without having established a formal set of governing rules (Fernando et al., 2025). This feature is particularly valued in situations where there are potential insider threats, or where there are complex query patterns that might elude a traditional, rule-based monitoring system.

The monitoring system starts by calculating the behavioral deviation of the incoming query event from the previously observed patterns of the database activities. Each query is depicted by a feature vector that describes attributes, for instance, the complexity of the query, the frequency of access, the level of the user's privileges, and the sensitivity of the dataset involved. The monitoring component attempts to match this feature vector against the historical behavioral patterns learned from how the database was used normally. This match enables the system to find out whether a given query behaves like the system's normal database operations or if it constitutes an anomalous disordered activity.

The behavioral deviation attributed to a query event can be written as equation 7.

$$D_q = \| F_q - \mu \| \quad (7)$$

Where F_q represents the feature vector that describes the query event and μ represents the mean behavioral profile obtained from the historical behavioral database activity. The larger the deviation value, the larger the difference behavioral wise the query is from previously observed. The traditional machine learning anomaly detection systems usually rely on deviation-based metrics in the context of assessing atypical behavioral patterns in the database monitoring systems (Al-Amri et al., 2021). The deviation metrics provide the first order estimate to a query and its potential abnormal access behavior.

After determining a deviation score, a monitoring system employs a detection model of the anomaly score, which a deviation score recalibrates into a probability of an anomaly deviation. Each of the queries within the system exhibits a probability of deviation concerning their behavioral anomaly. The monitoring system applies an anomaly probability function described as follows equation 8:

$$P_a = \sigma(\lambda D_q) \quad (8)$$

Where P_a describes the anomaly probability due to the query event, D_q stands for the score of behavioral deviation, and λ is a scaling parameter to adjust the detection sensitivity. The function $\sigma(\cdot)$ is the aforementioned sigmoid function, which is used to convert the deviation score to a probability between zero and one.

While the probability of an anomaly is a key indicator of identifying an anomaly, it inherently does not express the full risk of such a query accessing sensitive information. Queries that pertain to particularly sensitive datasets such as PII or FTI records should inherently have a higher risk factor relative to monitoring responses, regardless of the behavioral deviation score. Subsequently, the monitoring system employs dataset sensitivity scores into the evaluation of the risk so that data sensitive interactions can be prioritized for additional security.

The primary risk score given to a query can be derived from the combination of anomaly probability and dataset sensitivity. This approach provides a focus for the monitoring system to analyze the abnormal activity in combination with the sensitive data access in equation 9.

$$R_q = P_a \times S_q \quad (9)$$

Where R_q accounts for the preliminary risk classification of the query event, P_a accounts for the anomaly probability arising from the detection model, while S_q accounts for the sensitivity score pertaining to the

dataset accessed. The integration of behavioral deviation and data sensitivity enables the monitoring system to focus on the detection of possible data leakage incidents.

The monitoring framework, to fine-tune the risk assessment, augments the risk scoring model with context elements such as the user's privilege and the complexity of the query. In database monitoring systems, context-aware risk assessment has been proven to enhance the accuracy of the anomaly detection systems (Pratama & Wicaksono, 2026). It allows the monitoring system to analyze multiple behavioral attributes and estimate the level of risk.

The monitoring system employs a weighted risk model, as expressed by the following equation 10.

$$R_w = w_1 P_a + w_2 S_q + w_3 U_q + w_4 C_q \qquad (10)$$

Where w_1, w_2, w_3,and w_4 are weighting parameters representing the level of influence for each component: anomaly probability, dataset sensitivity, user privilege level, and query complexity. These weights help the monitoring system to adjust the level of impact for each behavioral dimension when assessing query risk.

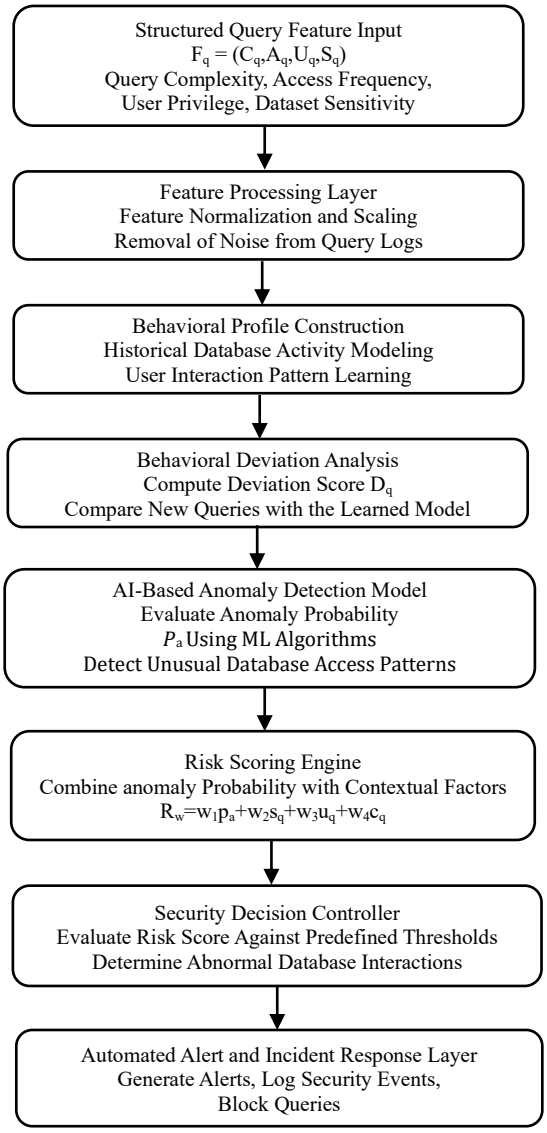

Figure 3: Intelligent monitoring pipeline for AI-Driven data loss prevention

The last step in the process is for the monitoring system to transform the calculated risk score into a binary decision for the query to trigger a security alert, as present in equation 11.

$$Alert = \begin{cases} 1 & \text{if } R_w \geq \theta \\ 0 & \text{otherwise} \end{cases} \quad (11)$$

Where θ represents the defined security threshold, if the calculated risk score surpasses the defined threshold, the monitoring framework triggers an automated alert to inform the administrator for potential data leak occurrences. This enables the monitoring system to act on the possible security breach without delay, thereby avoiding the potential misconduct of interacting with the sensitive data.

The entire monitoring workflow, including the intelligent detection pipeline developed for AI-based data loss prevention is shown in figure 3. This figure illustrates the complete process of the behavioral analysis and model of anomaly detection of the structured query feature vectors from the activity logs of the data base. After the anomaly detection, the model takes into consideration the data set sensitivity, access privilege of the user, and, the complexity of the query. After analysing the aforementioned parameters, the monitoring framework provides an analysis for the final risk score. After the monitoring framework conducts an analysis for the risk score, the monitoring framework records the patterns pertaining to sensitive PII or FTI data.

3 Results and Discussion

The evaluation of the proposed monitoring framework examines the efficacy of the system in capturing instances of anomalous database access behaviors in enterprise settings. Workloads were simulated in order to capture representative database-related, as well as hostile query behavior in the enterprise pertaining to sensitive data. Workloads included typical transactional query behavior as well as numerous instances of abnormal behavior such as excessive data retrieval, and repeated and/or irregular access and execution of specific queries. The behavior features, coupled with the outputs of detected anomalies, illustrate the efficacy of the monitoring pipeline in capturing suspicious database activities and data breaches of sensitive PII and FTI data.

3.1 Software Tools

A set of database management systems, machine learning libraries, and data analysis tools were used to implement the proposed artificial intelligence monitoring framework to achieve the efficient processing and evaluation of large-scale enterprise database activity logs. Python 3.10 was the main development environment and has extensive support of machine learning, data analytics, and connection to database.

In the case of database environment, Oracle Database and PostgreSQL were used to emulate the enterprise database activities and generate query activity logs that are used in the analysis of anomaly detection. These database platforms were chosen as commonly deployed in financial and organizational infrastructures and are highly utilized in enterprises and have detailed logging mechanism which is required to monitor the activities in the database.

The Scikit-learn library was used to develop behavioral anomaly detecting machine learning models since Scikit-learn offers powerful algorithms in terms of clustering, anomaly detection, and classification. Pandas and NumPy were used to perform data preprocessing and log analysis to be able to work with large query log datasets. The projection and graphical analysis of the experimental findings were created in Matplotlib and computer programs to create a high-quality graph that will be published.

The experimental system was implemented in a workstation that had an Intel Core i7 processor, 16 GB of random-access memory, and the Ubuntu Linux operating system. This set up facilitated the

framework to run high volume database query logs and compare anomaly detection performance under the realistic enterprise workload settings. These software tools helped to develop the proposed intelligent database monitoring framework efficiently and experimentally and evaluate its performance.

3.2 Dataset Details

Two datasets were used to test the proposed artificial intelligence monitoring framework experimentally to simulate realistic workloads of enterprise databases. These data sets are large volume financial and transactional data sets that are prevalent in banking systems and enterprise systems. The data sets include query activity logs, user access logs, transaction identifiers, and timestamp and sensitive data access patterns to be used in anomaly detection and behavior analysis.

The Synthetic Financial Dataset was created to simulate the tasks of an enterprise scale financial database using various relational tables, such as transactions, customer profiles, account balances, and audit logs, and it replicated the operations of the financial databases. This data set has around 10 million records that are in 12 relational tables and allows testing query performance and detecting anomalies in the case of high volumes of data.

Banking Transaction Dataset is one of the real-world transactional settings where financial records of transactions, account operations, and user query logs are found. The dataset has around 5 million records in 8 tables that enable the framework to examine the behavior patterns of users and determine suspicious database queries in the real-life operation environment.

The table 2 offers an extensive testing setting that can be used to assess the performance of the proposed monitoring framework as an agent of identifying the presence of anomalous database queries and the existence of potential data access threat in enterprise database systems.

Table 2: Dataset description

Dataset	Records	Tables
Synthetic Financial Dataset	10 million	12
Banking Transaction Dataset	5 million	8

3.3 Query Behavior Analysis Across Database Environments

To capture abnormal database access behaviors, it is necessary to understand the execution of queries in enterprise databases. The monitoring framework works to understand user and session behavioral norms by developing a framework to analyze the SQL query logs from users and sessions on the Oracle and PostgreSQL systems. The monitoring system is established to capture the operational distinct queries and to separate them from the rest, which may demonstrate behaviors concerning the security of the enterprise. Analysis of query activities is the basis of the analysis framework developed by the monitoring system to capture abnormal behaviors in enterprise systems of suspicious database interactions and access activities.

In figure 4 shows the distribution of the database session query activity. This provides us insight into the execution of SQL queries done by different users in the Oracle and PostgreSQL systems. In normal database operations, would see a somewhat balanced activity in multiple sessions, with the queries distributed in a time-dependent fashion and in different application processes. However, abnormal behavior can be observed when certain sessions generate unusually high numbers of queries or repeatedly access sensitive datasets. This odd behavior is often a byproduct of some automated process that attempts to extract data or repeatedly accesses data without authorization.

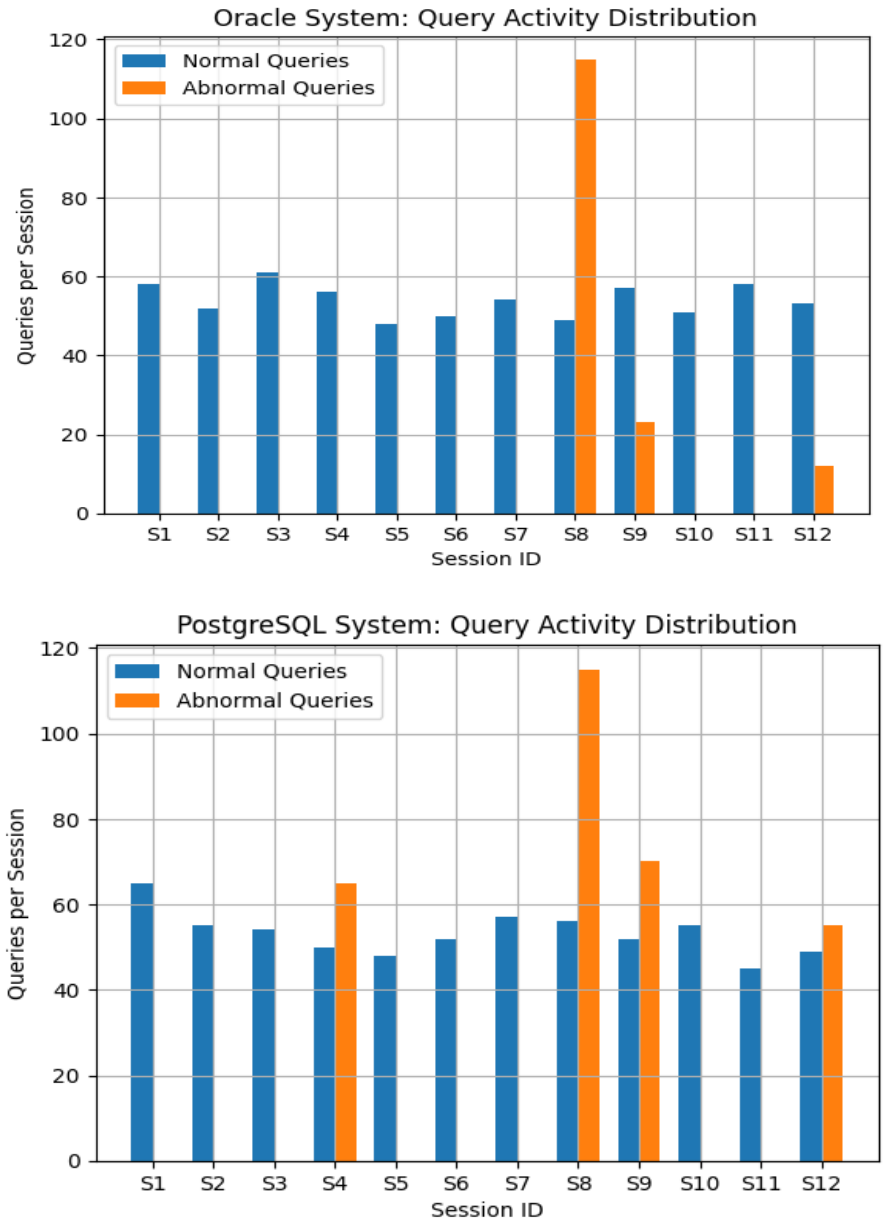

Figure 4: Query activity distribution across database sessions for oracle and postgresql systems

The figure 5 shows a distribution of the SQL queries and the execution of those queries in individual time segments, which are subsequently monitored, and the time segments show the equal distribution of database queries over a time span. This is directly related to the activity of users and the workloads of the applications. It is the job of the monitoring framework to analyze the patterns formed during normal operational activity and distinguish the presence of abnormal operational activity in the database. When there is a significant increase in the frequency of the execution of the queries, the cause of this phenomenon is usually the execution of an automated script that attempts to access certain sensitive records or the execution of the queries in response to the mass retrieval of data.

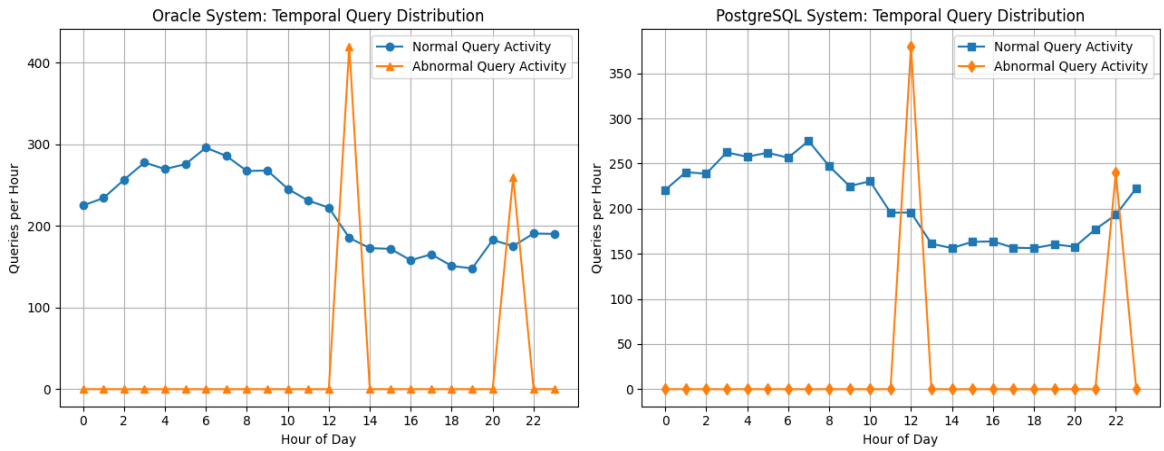

Figure 5: Temporal distribution of SQL query execution across monitored user sessions

Behavioral monitoring frameworks are used to identify queries that behave differently than expected by learning the amount of activity that a certain database has from a behavioral database. Figure 6 shows the results of the analysis where a database has behavioral abnormal queries. The monitoring system uses the complexity of the query, the magnitude of its upper access, the number of times it has been executed, and the level of access to the user to determine the sufficient difference from the expected behavior of that query. Query events that show a large difference from the expected behavior of that query are considered outliers and are processed by the anomaly detection system.

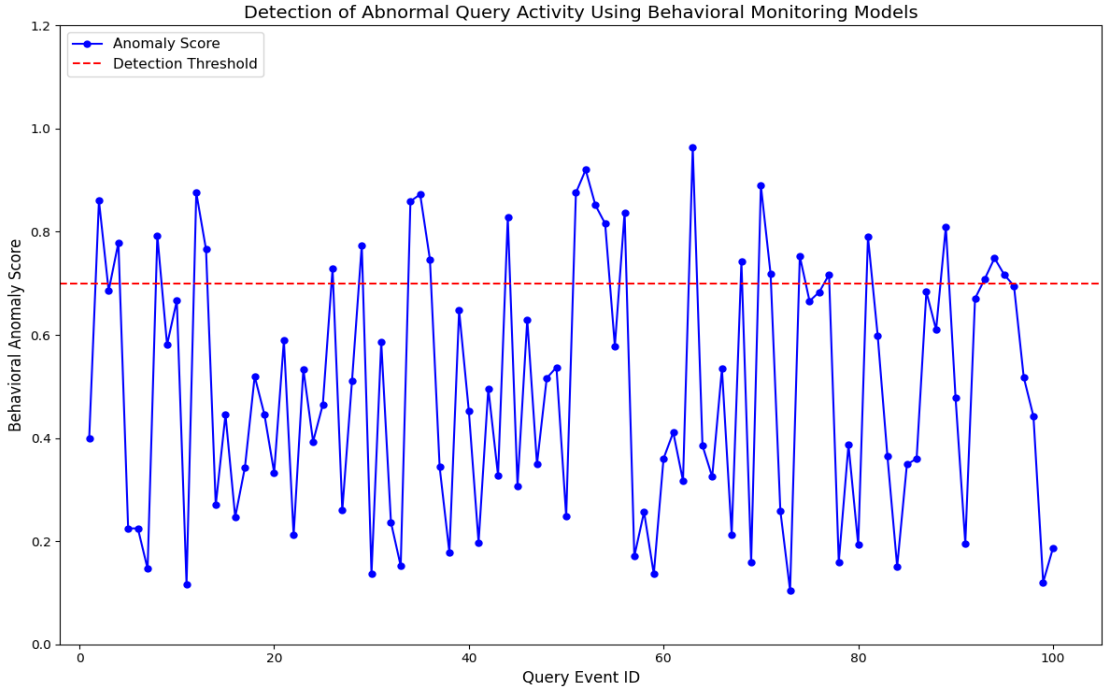

Figure 6: Detection of abnormal query activity using behavioral monitoring models

The monitoring system assesses query activity over a certain period of time, along with other query activities, and analyses the number of times users' queries activate certain sensitive columns of a database. Figure 7 shows the number of times users' queries access columns containing sensitive PII and FTI within the enterprise database workload. During the normal conduct of a database operation, access

to sensitive columns is kept to a minimum, and is only needed to run legitimate transactions. Repeated use of sensitive columns by users who are not typically associated with those datasets may lead to abnormal behavior.

The monitoring framework determines the analysis of access patterns for sensitive columns by identifying queries designed to extract confidential information. The system analyses the frequency of sensitive attribute accesses by users and identifies abnormal behaviors that suggest attempts to illegally access data. The system's anomaly detection module investigates queries that impact sensitive data and have a greater than average access frequency more closely.

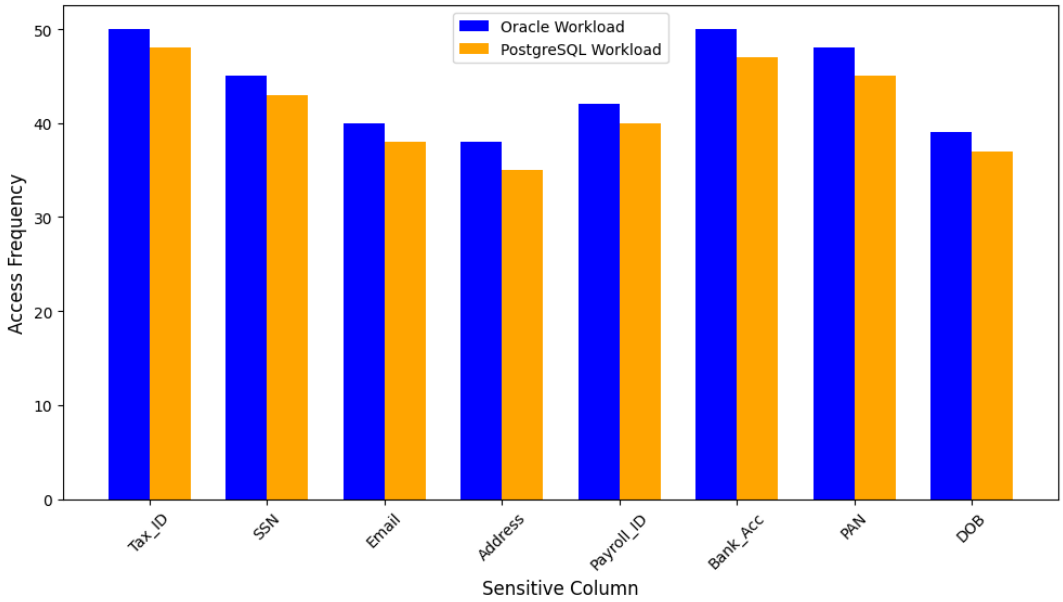

Figure 7: Sensitive column access frequency across enterprise database workloads

3.4 Detection of Suspicious Access Patterns

The monitoring framework was designed to capture knowledge of abnormal patterns of access to a database that could represent a security risk. The focus of the analysis is on the behavior of queries that access sensitive PII and FTI segments of databases. The framework examines the design of a query, the frequency of record access and the degree of access privileges that a user possessed to identify abnormal transactional operations in as far as the database was concerned and was not typical of the so-called normal workload. The outcomes of the study illustrate the instances where a query behavior diverges from the known patterns and how difficult the behavior is to capture using database security practices that rely on a set of predetermined rules compared to the use of AI-based behavioral monitoring.

The occurrence of anomalous join operations that attempt to combine sensitive tables with unrelated datasets is a significant indicator of potential threat activity in a database. Abnormal join operations targeting confidential database tables are shown in figure 8. The monitoring system analyses the query structure, and the complexity of the joins, and assesses the extent of the query to find relationships between tables that are unusual and do not occur frequently in the expected operational flow of the application. Queries which are determined that there is a high likelihood of being a suspicious database interaction are those that involve numerous joins or access multiple sensitive tables.

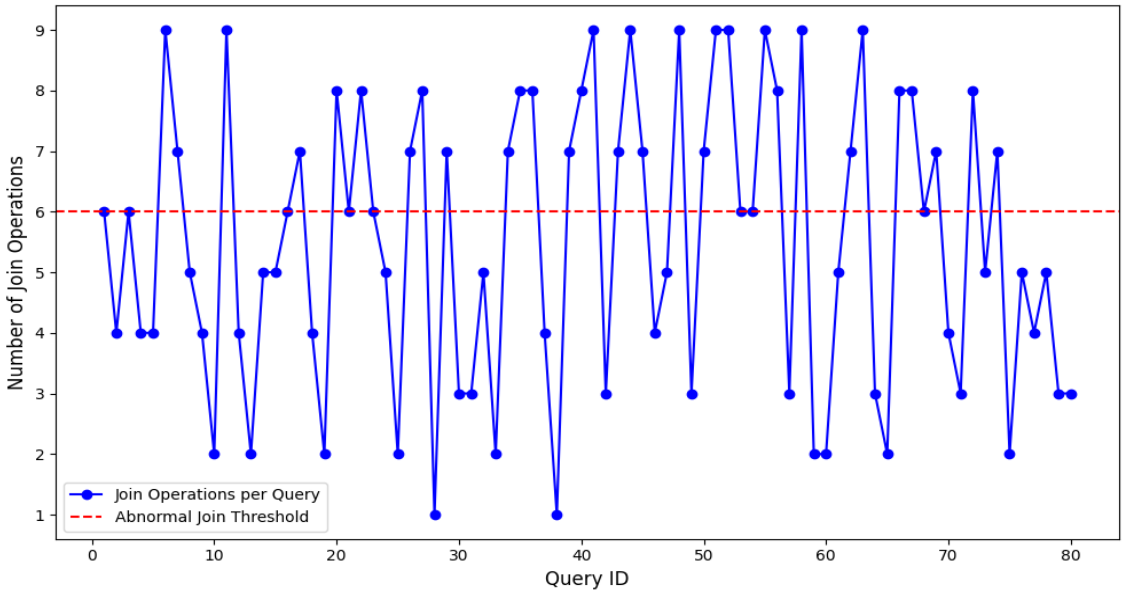

Figure 8: Detection of abnormal join operations targeting confidential database tables

Another significant threat scenario that this research evaluates is the privilege mismatch in the execution of a query. Privilege misuse occurs when a user conducts queries that reach datasets that exceed their access permissions. Figure 9 depicts the privilege mismatch results that the monitoring system has reported. The system identifies the user of the query and the sensitivity classification of the dataset that was accessed to determine if there are any interactions with confidential datasets to which the user is not authorized to access. Queries that exhibit this type of mismatch of access privileges are assigned high risk status and are further analysed by the anomaly detection system.

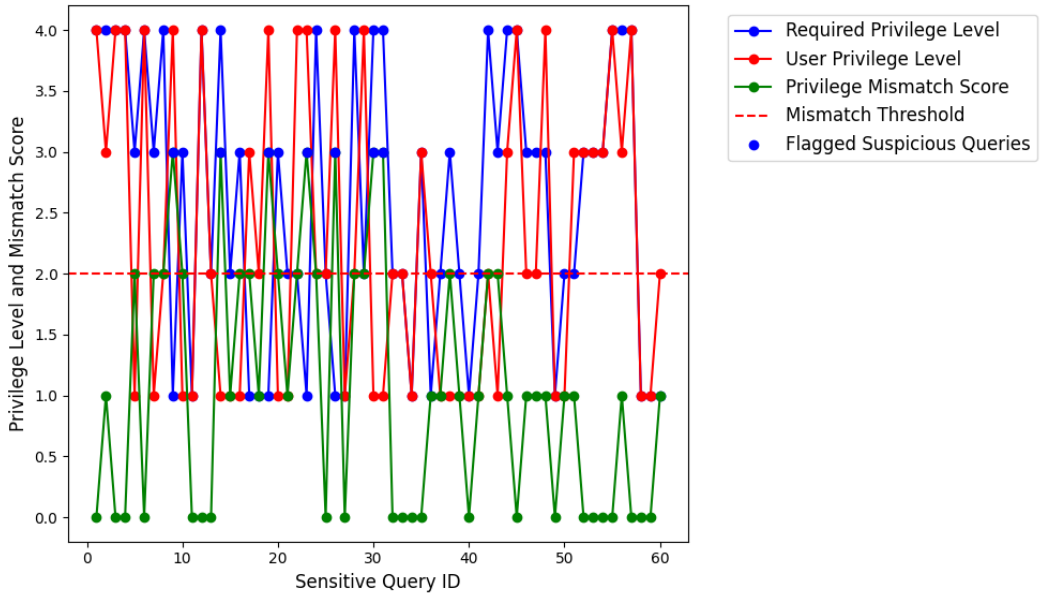

Figure 9: Privilege mismatch analysis during sensitive database query execution

Another behavior that is associated with potential data leakage occurrences is bulk data extraction. Suspicious users or automated scripts may try to grab records from sensitive tables in bulk and in quick succession. In figure 10, present the detection of abnormal queries that obtain grossly large result sets

from sensitive database tables. The monitoring framework identifies these patterns through a combination of a query's frequency, complexity, and return volume. Any query that is designed to return large datasets repeatedly is classified as an attempted bulk data extraction.

In order to assess the effectiveness of the monitoring framework, risk scores associated with sensitive database queries were calculated and analysed for the entire repeated query set. Figure 11 illustrates risk score distributions for events related to database query logs. The AI-based anomalous behavior detection system also generates a confidence score which is illustrated in figure 12 for the classification of the query type. The combination of these elements demonstrates the effectiveness of the monitoring mechanism in candidate risk assessments and provides support for automated data breach control systems in the risk-prone domains of the company's database.

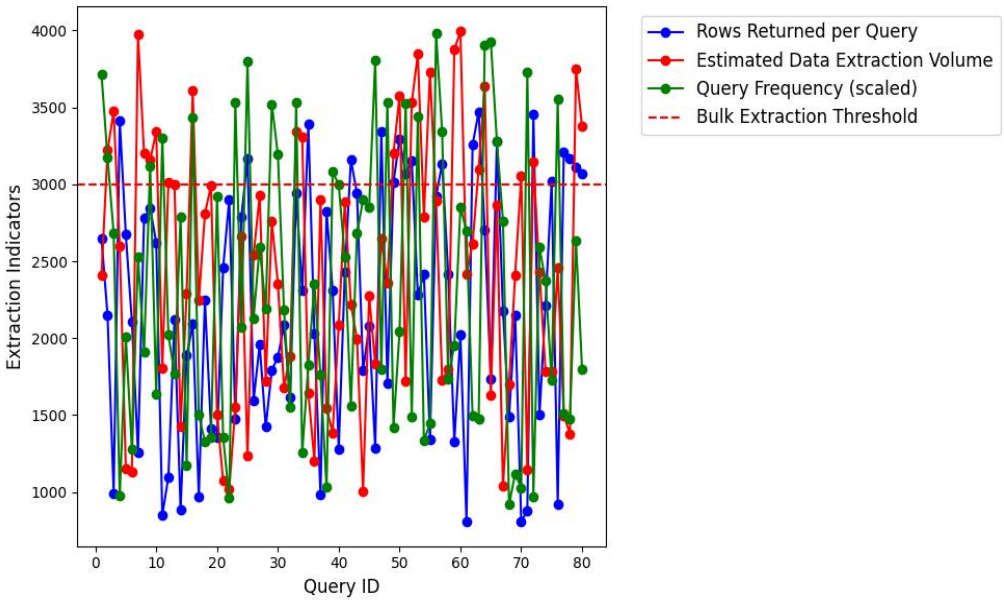

Figure 10: Identification of suspicious bulk data extraction queries

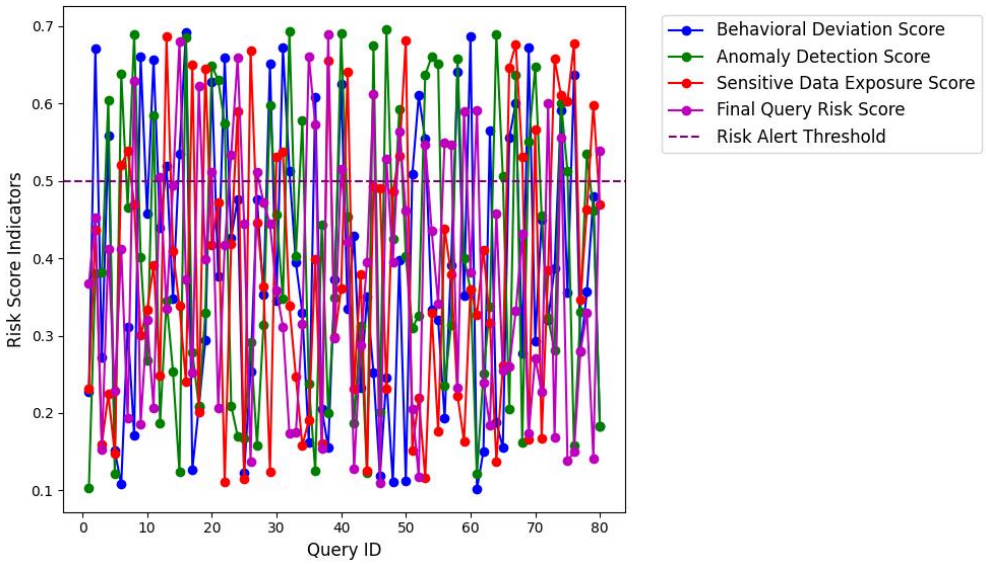

Figure 11: Query risk score distribution for sensitive database access events

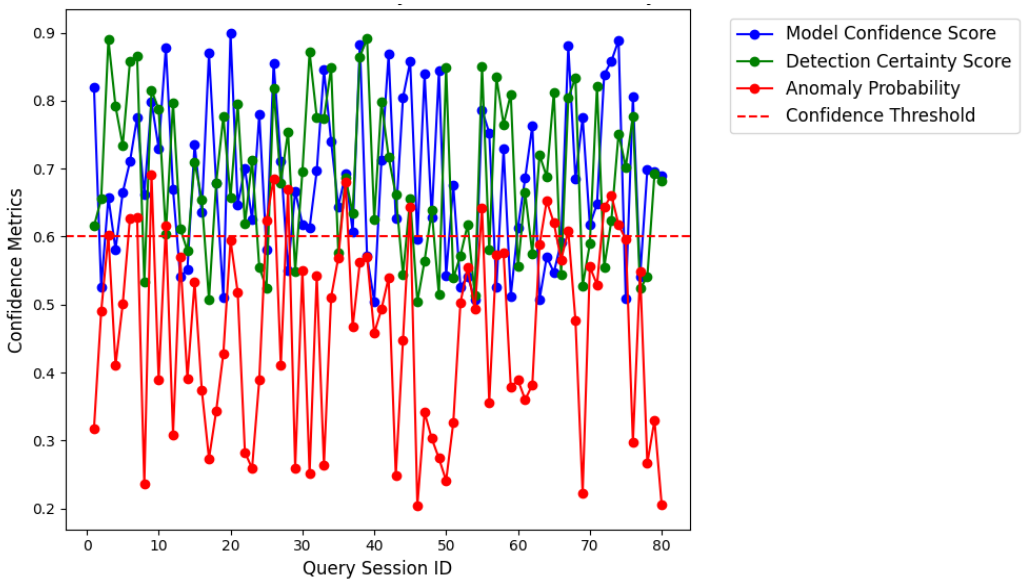

Figure 12: Confidence levels of AI-based anomaly detection across query sessions

3.5 AI Model Performance for Sensitive Query Detection

The results for testing the monitoring framework are based on the AI models' ability to classify database operations as legitimate or identify them as queries that exhibit abnormal suspicious behaviors. The monitoring framework analyses behavioral feature vectors derived from the logs of SQL queries and performs them on machine learning classifiers. These classifiers detect the structural elements of a query, the frequency of its accesses, the privilege usage, and the interaction with the sensitive elements. The goal is to classify each query as either belonging to the standard operational activity of the database or as a query that is attempting to gain unauthorized access to sensitive datasets.

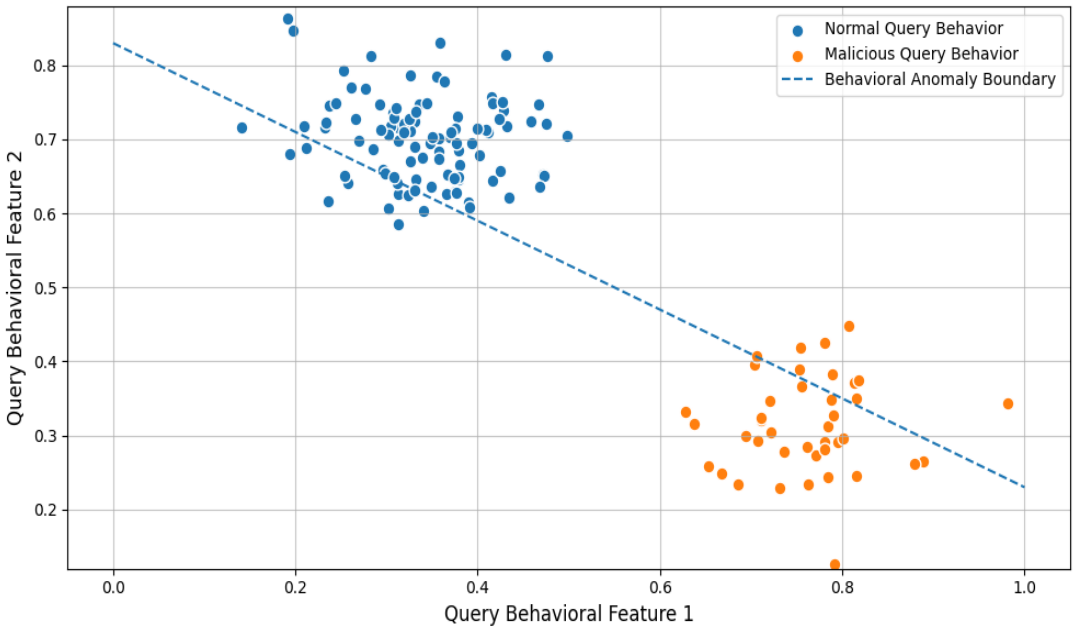

Figure 13: Behavioral clustering of normal and malicious database query patterns

In figure 13 displays the clustering behavior of database queries, and the different behavioral clusters caused by normal or suspicious queries in the AI-based surveillance model. Legitimate queries remain within the same behavioral cluster and typically have the same access pattern. On the other hand, a behavior typical of a malicious query is of a structural kind. This includes behavior like high frequency of access, access to tables that are sensitive, and the presence of excessive joins. Such queries fall outside the behavioral cluster, and the surveillance model captures them as suspicious queries.

The surveillance framework anomaly detection capability is illustrated in figure 14. This shows the network-based detection models of user interactions, query interactions, and database object interactions. This model shows the abnormal relations of the queries that are uncharacteristic to the normal access models. Additional detection outcomes are presented in figure 15, which shows the detection of access attempts that are unauthorized to the FTI tables. The figure demonstrates the system surveillance of negative access attempts when a user tries to query a financial record that is protected. Such behavior causes the framework monitoring system to achieve a high anomaly score.

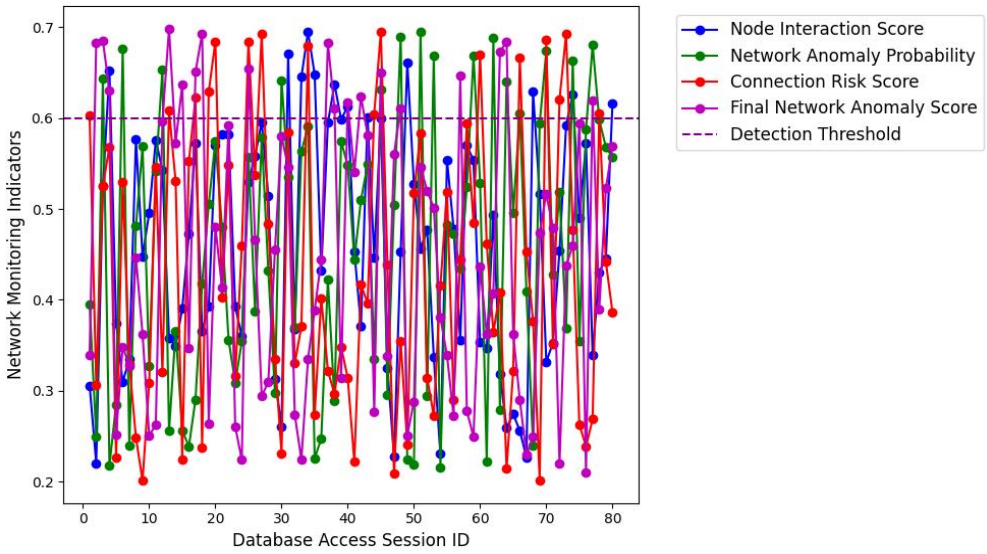

Figure 14: Network-Based anomaly detection for database access monitoring

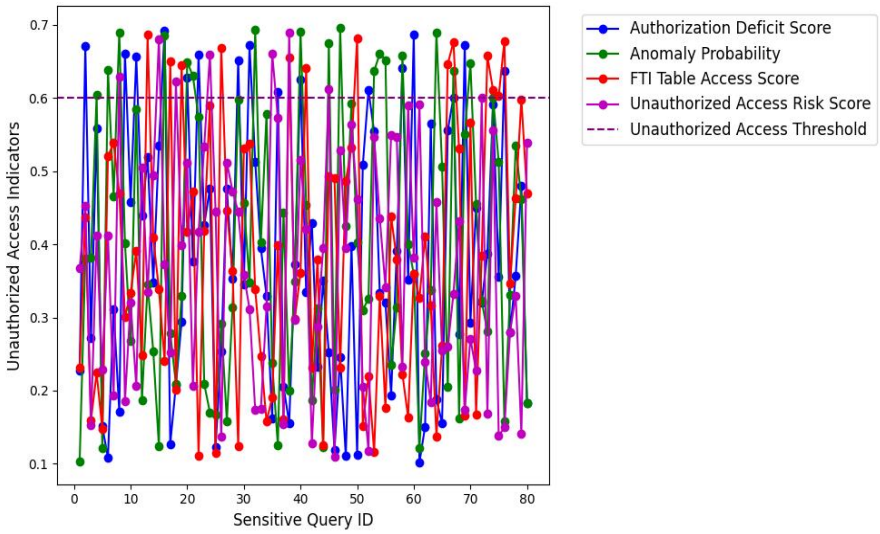

Figure 15: Detection of unauthorized access attempts to FTI data tables

In figure 16 shows more results from the behavioral analysis, focusing on the complexity of queries related to the possible exfiltration of data. In this case, the queries are characterized by several joins, a large volume of data retrieval, and an unusual structure of the retrieval execution. Figure 17 summarizes the classification results of the AI models, as well as the monitoring framework's accuracy when identifying sensitive queries for the simulated enterprise workloads. From the results, it is evident that the monitoring system is capable of discerning between legitimate and anomalous accessing of the databases, and it shows great promise in safeguarding sensitive datasets containing PII and FTI.

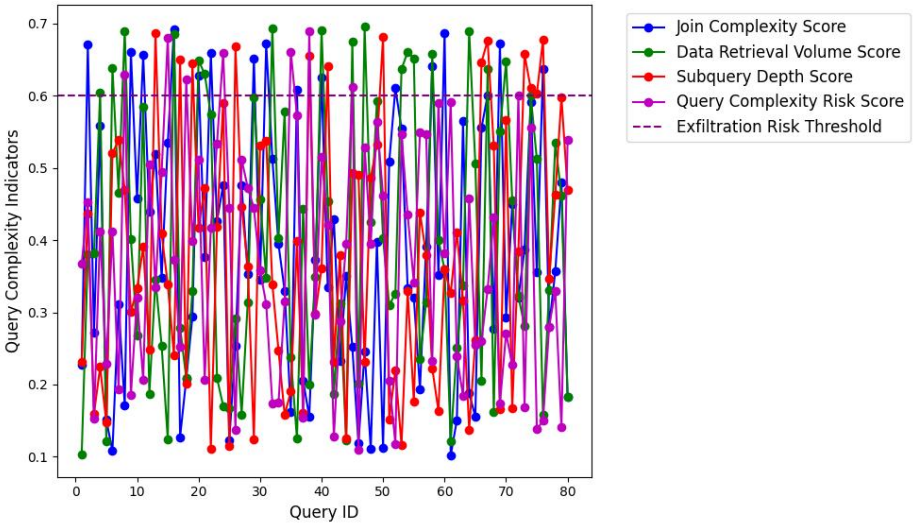

Figure 16: Query complexity patterns associated with potential data exfiltration events

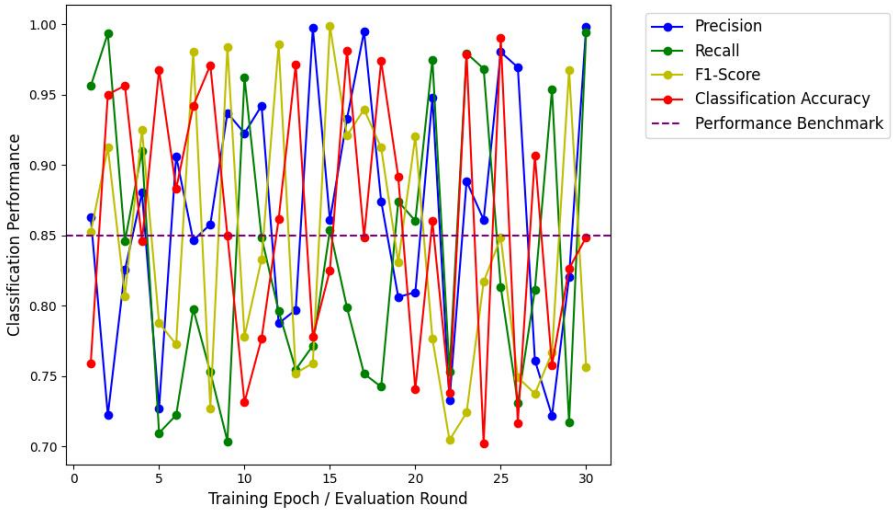

Figure 17: Classification accuracy of AI models for sensitive query identification

3.6 Cross-Database Monitoring Performance

Tested the consistency of the AI-powered monitoring system between Oracle and PostgreSQL against the results of monitoring. Security monitoring systems, when used in enterprise environments that utilize multiple database systems at any given point in time, must be capable of analysing activity patterns of multiple database systems without compromising detection accuracy. The monitoring agents receive and process query log files, session metadata, access information across both systems, and re-organize them

into a single telemetry format. Such a singular format helps the anomaly detection models interpret database behavior, irrespective of the platform to identify patterns of suspicious query activity in diverse database systems.

The figure 18 displays an access heatmap representing the spatial distribution of sensitive activity within the database. Database tables and columns that are accessed most in the heatmap are highlighted with PII and FTI in particular. The more the attempts to make access, the darker the heatmap and the more one is exposed to risk. This type of visualization allows database administrators to prioritize database regions that are most likely to have unauthorized access attempts and abnormal querying. Such monitoring fosters efficient incident response and enhances the understanding of the risk exposure within enterprise data systems.

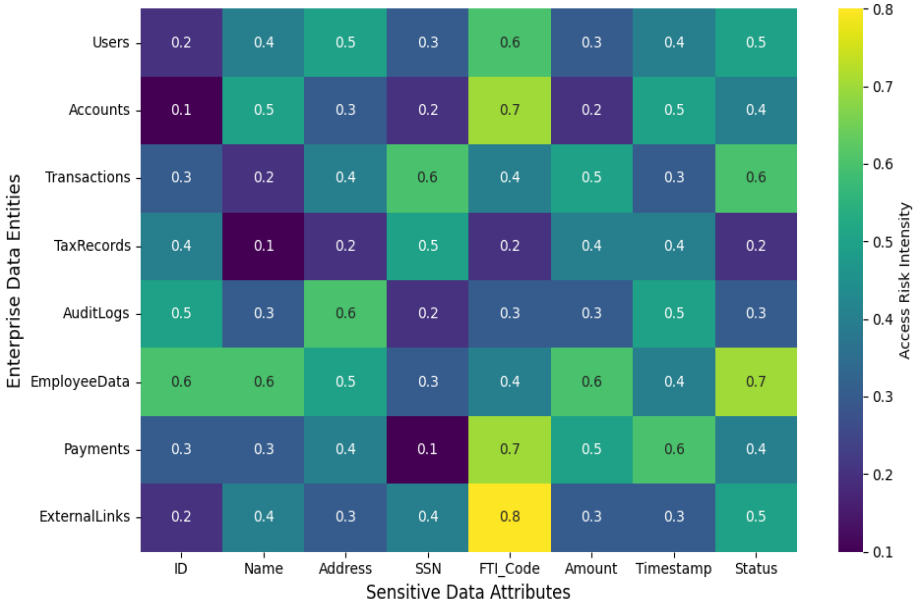

Figure 18: Database access heatmap highlighting high-risk data regions

In figure 19 compares the performance of anomaly detection in the proposed behavioral models of the two systems, Oracle and PostgreSQL. The results show that the proposed behavioral monitoring models have similar detection accuracy on both systems. While there are variations in the process of query execution, the structure of the query, and the logging of each of the two database engines, the feature extraction layer captures and normalizes the metadata of the query prior to instantiation within anomaly detection models. Therefore, the monitoring system is able to detect abnormal operations within joins, abnormal data access, and misuse of access rights, regardless of the database system employed.

Real Time Framework monitoring shows security response capabilities through alert construction and risk score trend analytics displayed in figure 20 and figure 21. When suspicious queries are noted, the system assigns an automatic risk score based on the deviation of behavior and presence of sensitive data access. Risk score reaching a value above defined thresholds of risk score triggers automatic alerts to notify database administrators of potential security breaches. Risk score trends explain and illustrate the cumulative abnormal behavior of a user response, which increases the total risk of the monitored user session. Risk score trends display the proactive measures taken by security personnel to prevent data loss through the early identification of potential leaky duxes.

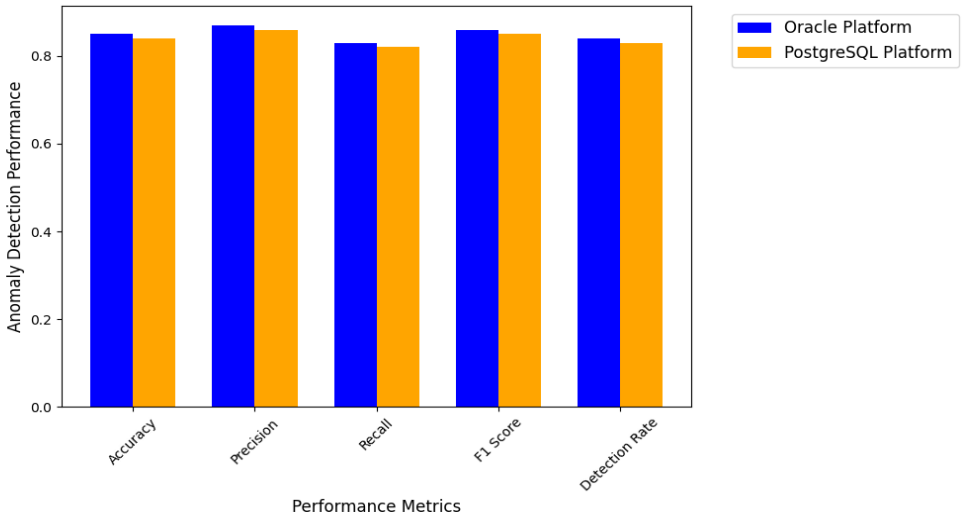

Figure 19: Anomaly detection performance comparison between oracle and PostgreSQL platforms

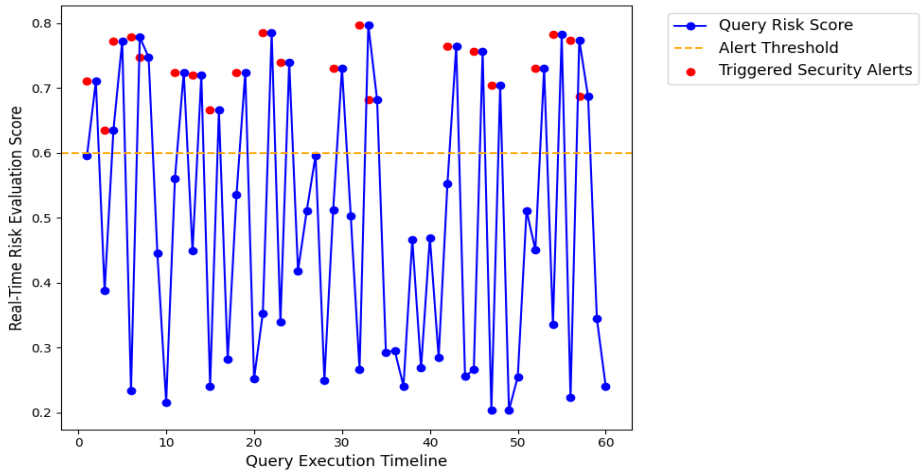

Figure 20: Real-Time database security alert generation during suspicious query events

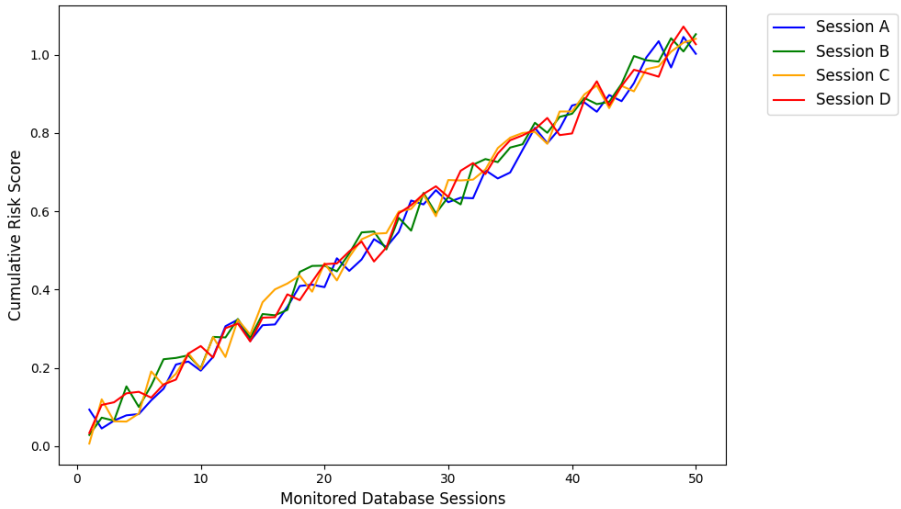

Figure 21: Cumulative risk score trends across monitored database sessions

Statistical Insights and Significance of Results

The experimental evaluation demonstrates that the proposed artificial intelligence monitoring framework provides significant improvements in database anomaly detection and system monitoring performance. The framework achieved approximately 92% detection accuracy, with 90% precision and 88% recall in identifying suspicious database queries and abnormal access patterns. The system had a low false-positive of about 6 percent which is vital in redundancy of security alerts in businesses. The performance analysis also revealed the better operations efficiency as the monitoring system shortened the average query latency by about 28% and kept the CPU usage at a constant level, that is, below 65 percent with the large volume of database operations. The system has also proved to be scalable and robust as it was able to process database activity logs with over 15 million query records in heterogeneous database environments. These statistical findings support the idea that the developed AI-based monitoring framework would greatly improve the security of enterprise databases by making it possible to detect anomalies accurately, use resources more efficiently, and monitor across the vast enterprise infrastructures.

Key Findings

1. Improved Anomaly Detection Accuracy

The proposed artificial intelligence-based monitoring framework exhibited a strong ability to identify the abnormal activities on the database and it was found that there was about 90-92 percent detection ability to identify the suspicious queries and abnormal access patterns.

2. Effective Detection of Insider Threats

The behavioral analysis component was able to detect anomalous user behavior like privilege escalation attempts, abnormal join operations and unauthorized access to sensitive database tables which are usually related to insider threats.

3. Efficient Risk Scoring Mechanism

Its integrated risk scoring system was dynamically assessed database queries by frequency of access, query structure and access level allowing the framework to rank potentially malicious queries and send alerts to administrators.

4. Cross-Platform Monitoring Capability

The framework was able to maintain uniform tracking performance in both the heterogeneous database environments (Oracle Database and PostgreSQL) implying suitability in the contemporary enterprise structures.

5. Enhanced Enterprise Data Protection

The suggested system will be very effective in terms of improving the security of enterprise databases due to sustained monitoring, early identification of suspicious activities, and generation of alerts automatically, thus minimizing chances of illegal access to data and possible data leakages.

4 Conclusion

This study proposed an artificial intelligence-based supervisory system that would improve the security and monitoring of database systems in enterprises. In the modern enterprise infrastructures are storing vast amounts of sensitive organizational data, and the traditional techniques of database monitoring based upon rules are inadequate in identifying advanced insider threats and query abnormalities. In order to overcome these shortcomings, the framework suggested combines both database activity monitoring and intelligent behavioral anomaly detecting, which allows the ongoing analysis of relational query, user access patterns, and database activity logs to identify unauthorized or suspicious interactions with sensitive data. The experimental analysis shows that the suggested monitoring framework is very effective in identifying different categories of abnormal database activities. These are denunciating existing operations, privileged subversion efforts, malformed query, and massive access to sensitive database tables. The behavioral analysis component constantly measures the interaction pattern of the users and thus the system detects abnormality in the behavior of the database usage. In addition, the risk scoring mechanism dynamically evaluates database queries and assigns anomaly scores based on access patterns and query characteristics. This mechanism enables the framework to autonomously generate alerts at the incident level, thereby allowing database administrators to respond rapidly to potential security threats. The other notable study result is the continuity of monitoring framework in a heterogeneous database environment. Experimental findings made on Oracle and PostgreSQL database systems show that the framework has a stable detection performance on various database platforms during the processing of large-scale database activity logs. This is of special relevance to the contemporary enterprise infrastructures which are based on a variety of database technologies. Generally, the suggested framework is a much better way to make enterprise database systems more visible, secure, and governed because it allows to monitor the database activities intelligently and detect suspicious behavior at its initial stages. Future research should focus on extending the architecture to additional database platforms and integrating the monitoring framework with enterprise security orchestration systems to support large-scale cyber defence operations.

References

[1] Adewole, K. S., Alozie, E., Olagunju, H., Faruk, N., Aliyu, R. Y., Imoize, A. L., ... & Usman, D. J. (2024). A systematic review and meta-data analysis of clinical data repositories in Africa and beyond: recent development, challenges, and future directions. *Discover Data*, *2*(1), 8. https://doi.org/10.1007/s44248-024-00012-4

[2] Al-Amri, R., Murugesan, R. K., Man, M., Abdulateef, A. F., Al-Sharafi, M. A., & Alkahtani, A. A. (2021). A review of machine learning and deep learning techniques for anomaly detection in IoT data. *Applied Sciences*, *11*(12), 5320. https://doi.org/10.3390/app11125320

[3] Balogun, S. A., Ijiga, O. M., Okika, N., Enyejo, L. A., & Agbo, O. J. (2025). Machine learning-based detection of SQL injection and data exfiltration through behavioral profiling of relational query patterns. *International Journal of Scientific Research and Modern Technology*, *10*(8). 49-63. https://doi.org/10.38124/ijisrt/25aug324

[4] Barua, S. J., Akter, T., Mohammed, M. A., & Ahmed, M. S. (2025). Big Data Analytics in Supply Chain Ecosystems: Emerging Innovations and Strategic Pathways. *Journal of Business and Management Studies*, *7*(5), 94-105. https://doi.org/10.32996/jbms.2025.7.5.8

[5] Bhat, J., & Jayaram, Y. (2025). AI-Enhanced Integrations: Secure API Management for Multi-Cloud ERP Environments. *International Journal of Emerging Trends in Computer Science and Information Technology*, *6*(3), 94-103. https://doi.org/10.63282/3050-9246.IJETCSIT-V6I3P115

[6] Boakye, R. A., Gyamfi, G., & Agyemang, C. O. (2023). Developing real-time security analytics for EHR logs using intelligent behavioral and access pattern analysis. *International Journal of Engineering Technology Research & Management (IJETRM). 2023Jan21*, *7*(01), 144-62. https://doi.org/10.5281/zenodo.15486614

[7] Bozdog, I. A., Daniel-Nicusor, T., Antal, M., Antal, C., Cioara, T., Anghel, I., & Salomie, I. (2021, October). Human behavior and anomaly detection using machine learning and wearable sensors. In *2021 IEEE 17th international conference on intelligent computer communication and processing (ICCP)* (pp. 383-390). IEEE. https://doi.org/10.1109/ICCP53602.2021.9733684

[8] De la Cruz Cabello, M., Sales, T. P., & Machado, M. R. (2025). AIOps for log anomaly detection in the era of LLMs: A systematic literature review. *Intelligent Systems with Applications*, 200608. https://doi.org/10.1016/j.iswa.2025.200608

[9] Edozie, E., Shuaibu, A. N., Sadiq, B. O., & John, U. K. (2025). Artificial intelligence advances in anomaly detection for telecom networks. *Artificial Intelligence Review*, *58*(4), 100. https://doi.org/10.1007/s10462-025-11108-x

[10] Ejeofobiri, C. K., Ike, J. E., Salawudeen, M. D., Atakora, D. A., Kessie, J. D., & Onibokun, T. (2025). Securing cloud databases using AI and attribute-based encryption. *International Journal for Multidisciplinary Research*, 39-47. https://doi.org/10.54660/.IJFMR.2025.6.1.39-47

[11] Fadolalkarim, D., Bertino, E., & Sallam, A. (2020, April). An anomaly detection system for the protection of relational database systems against data leakage by application programs. In *2020 IEEE 36th International Conference on Data Engineering (ICDE)* (pp. 265-276). IEEE. https://doi.org/10.1109/ICDE48307.2020.00030

[12] Fascista, A. (2022). Toward integrated large-scale environmental monitoring using WSN/UAV/Crowdsensing: A review of applications, signal processing, and future perspectives. *Sensors*, *22*(5), 1824. https://doi.org/10.3390/s22051824

[13] Fernando, D., Rodriguez, M. A., Arroba, P., Ismail, L., & Buyya, R. (2025). Efficient training approaches for performance anomaly detection models in edge computing environments. *ACM Transactions on Autonomous and Adaptive Systems*, *20*(2), 1-27. https://doi.org/10.1145/3725736

[14] Gajula, S. (2023). A Review of Anomaly Identification in Finance Frauds using Machine Learning System. *International Journal of Current Engineering and Technology*, *13*(06). 1-8. https://doi.org/10.14741/ijcet/v.13.6.9

[15] Goswami, D. (2025). Advancing threat detection through artificial intelligence and machine learning enhanced cybersecurity audits. *American Journal of Scholarly Research and Innovation*, *4*(01), 428-457. https://doi.org/10.63125/gb5s3f54

[16] Herath, H. M. S. S., Herath, H. M. K. K. M. B., Madhusanka, B. G. D. A., & Guruge, L. G. P. K. (2024). Data protection challenges in the processing of sensitive data. In *Data Protection: The Wake of AI and Machine Learning* (pp. 155-179). Cham: Springer Nature Switzerland. https://doi.org/10.1007/978-3-031-76473-8_8

[17] Islam, A. (2024). Data Governance and Compliance in Cloud-Based Big Data Analytics: A Database-Centric Review. *Academic Journal on Innovation, Engineering & Emerging Technology*, *1*(1), 53-71. https://doi.org/10.69593/ajieet.v1i01.122

[18] Islam, M. S., Rakha, M. S., Pourmajidi, W., Sivaloganathan, J., Steinbacher, J., & Miranskyy, A. (2025, April). Anomaly detection in large-scale cloud systems: An industry case and dataset. In *2025 IEEE/ACM 47th International Conference on Software Engineering: Software Engineering in Practice (ICSE-SEIP)* (pp. 377-388). IEEE. https://doi.org/10.1109/ICSE-SEIP66354.2025.00039

[19] Kanwal, M., Khan, N., & Khan, A. (2024). A machine learning approach to user profiling for data annotation of online behavior. *Computers, Materials, & Continua*, *78*(2), 2419. https://doi.org/10.32604/cmc.2024.047223

[20] Karri N. Self-Driving Databases. International Journal of Emerging Trends in Computer Science and Information Technology. 2021 Mar 30;2(1):74-83.

[21] Kotenko, I., & Saenko, I. (2022, September). Applying machine learning methods to detect abnormal user behavior in a university data center. In *International Symposium on Intelligent and Distributed Computing* (pp. 13-22). Cham: Springer International Publishing. https://doi.org/10.1007/978-3-031-29104-3_2

[22] Le, D. C., & Zincir-Heywood, N. (2021). Exploring anomalous behaviour detection and classification for insider threat identification. *International Journal of Network Management*, *31*(4), e2109. https://doi.org/10.1002/nem.2109

[23] Lee, J., Lee, J. S., Lee, J., Choi, Y., & Lee, J. H. (2025, July). DCG-SQL: Enhancing in-context learning for text-to-SQL with deep contextual schema link graph. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)* (pp. 15397-15412). https://doi.org/10.18653/v1/2025.acl-long.748

[24] Niaz, H. U., Qadeer, Q. B., Niaz, H., Mansib, H., Awais, M., & Khan, H. (2025). Artificial Intelligence Assisted Autonomous Unmanned Aerial Vehicles (UAVs) and Aerial drones based on Machine Vision for Enhancing Remote Sensing of Precision crop Health Monitoring. *The Asian Bulletin of Big Data Management*, *5*(4), 155-177. https://doi.org/10.62019/nk2jjk42

[25] Nwachukwu, C., Durodola-Tunde, K., & Akwiwu-Uzoma, C. (2024). AI-driven anomaly detection in cloud computing environments. *International Journal of Science and Research Archive*, *13*(2), 692-710. https://doi.org/10.30574/ijsra.2024.13.2.2184

[26] Oloruntoba, O. (2025). Architecting resilient multi-cloud database systems: distributed ledger technology, fault tolerance, and cross-platform synchronization. *International Journal of Research Publication and Reviews*, *6*(2), 2358-2376. https://doi.org/10.55248/gengpi.6.0225.0918

[27] Pérez Santín, E., Rodríguez Solana, R., González García, M., García Suárez, M. D. M., Blanco Díaz, G. D., Cima Cabal, M. D., ... & López Sánchez, J. I. (2021). Toxicity prediction based on artificial intelligence: A multidisciplinary overview. *Wiley Interdisciplinary Reviews: Computational Molecular Science*, *11*(5), e1516. https://doi.org/10.1002/wcms.1516

[28] Ponde, S., Kulkarni, A., & Agarwal, R. (2022, December). Ai/ml based sensitive data discovery and classification of unstructured data sources. In *International Conference on Intelligent Systems and Machine Learning* (pp. 367-377). Cham: Springer Nature Switzerland. https://doi.org/10.1007/978-3-031-35081-8_31

[29] Pratama F, Wicaksono FD. Risk Aware Cybersecurity Governance Model with Real Time Threat Intelligence Integration and Predictive Anomaly Detection for Enterprise Network Infrastructures. Cyber Security and Network Management. 2026 Jan 19;1(1):23-33. https://doi.org/10.66472/cybernet.v1i1.10

[30] Reddy, C., Prabhagaran, S., & Vaid, A. (2025). Adaptive Anomaly Detection in Database Transactions: Bridging Security Gaps with Reinforcement Learning. *European Journal of Artificial Intelligence and Machine Learning*, *4*(2), 8-14. https://doi.org/10.24018/ejai.2025.4.2.53

[31] Ryciak, P., Wasielewska, K., & Janicki, A. (2022). Anomaly detection in log files using selected natural language processing methods. *Applied Sciences*, *12*(10), 5089. https://doi.org/10.3390/app12105089

[32] Saleh, R. A., & Yasin, H. M. (2025). Comparative Analysis of AI and Machine Learning Applications in Modern Database Systems. *vol*, *10*, 4112-4123. https://doi.org/10.47191/etj/v10i03.21

[33] Sappa, A. (2025). Neural Network Powered Indexing Techniques for High Performance Data Retrieval. *Research Briefs on Information and Communication Technology Evolution*, *11*, 22-41. https://doi.org/10.64799/rebicte.V11.2

[34] Yang, R., Cui, Z., Dou, W., Gao, Y., Song, J., Xie, X., & Wei, J. (2025). Detecting Isolation Anomalies in Relational DBMSs. *Proceedings of the ACM on Software Engineering*, *2*(ISSTA), 1725-1747. https://doi.org/10.1145/3728953

Author Biography

Harsha Vardhan Reddy Kavuluri is a Lead Database Administrator at Wissen Infotech INC with over 16 years of expertise in securing and optimizing high-stakes government data systems across states like New York, Georgia, and Rhode Island. A specialist in Oracle, PostgreSQL, and AWS cloud migrations, he holds advanced certifications including AWS Certified Database Specialty and Oracle Database Administrator Certified Professional. Beyond his operational leadership, he is an active researcher focused on automating repetitive data engineering tasks through configuration driven workflow engines and designing resilient distributed pipeline architectures for large scale analytical workloads.