International Academic Institute for Science and Technology

International Academic Journal of Innovative Research
Vol. 8, No. 4, 2021, pp. 38-47.

ISSN 2454-390X

International Academic Journal of Innovative Research

www.iaiest.com

AI-Powered Anomaly Detection in Oracle Database: Leveraging Machine Learning for Proactive Threat Mitigation

Adithya Sirimalla [a] , Harsha Vardhan Reddy Kavuluri [b] , Suresh Babu Avula [c]

[a] Software Developer and DBA, 20755 Williamsport Pl suite 230, One Loudoun, VA, USA. adithya.skal@gmail.com,
b Lead Database Administrator, Wissen Infotech Inc, USA. kavuluri99@gmail.com
c Associate Director, Innovaccer Analytics Pvt Ltd, Noida, Uttar Pradesh, India. sureshbabuavula@gmail.com

Abstract

Lately, the vast increase in data and the rising complexity of business systems have made traditional security systems in database management less practical. Because Oracle Database is widely employed in businesses, it is commonplace where valuable and critical business data is found. Any unusual activity from persons within the organisation, bad misconfigurations or outsiders can cause serious problems. It introduces a reliable system for detecting abnormal activities in Oracle Databases with ML. This system uses past access records, SQL logs and system usage data to identify situations that need further examination. A set of approaches that fuse supervised learning with unsupervised learning using Isolation Forests, Autoencoders and LSTM systems is presented in this article. Based on comparative studies, our method produces dependable results with a small number of mistakes. In addition, we suggest a way to set up AVDF that easily connects to both Oracle Audit Vault and Database Firewall (AVDF). The research demonstrates that AI plays an important role in making databases safer and highlights what improvements could be made in the future.

Keywords: Oracle Database, Anomaly Detection, Machine Learning, Isolation Forest, Autoencoder, LSTM, Threat Mitigation.

Received: 16 September 2021/Revised: 21 October 2021/Accepted: 26 November 2021/Published: 31 December 2021

Introduction

Databases support most organisations today, as they store a large amount of critical data that powers businesses. Oracle Database is used extensively in many industries, including those relating to finance, health and government. While Oracle has robust security, users still experience breaches and discover anomalies due to zero-day threats, tricks by untrustworthy sources or attacks from trusted employees. Most intrusion detection and rule-based systems use patterns that have been set in advance. [1-4] Even if they are effective in some ways, they do not prevent unexpected zero-day attacks. Moreover, because user actions, queries and system use vary all the time, it is not possible to pinpoint all security issues using just fixed rules.

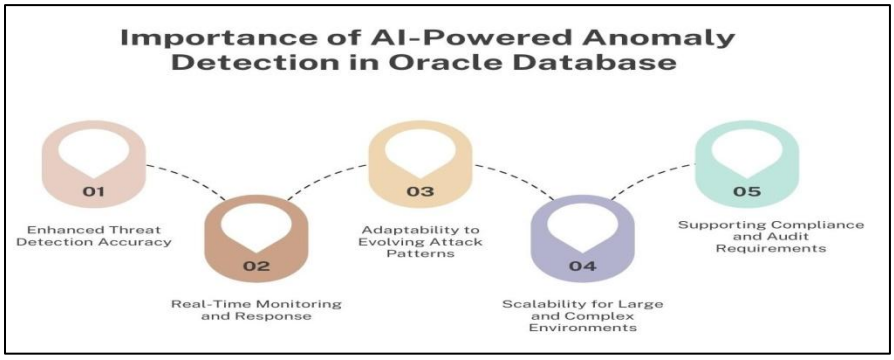


Fig. 1: Importance of AI-Powered Anomaly Detection in Oracle Database

1.1. Importance of AI-Powered Anomaly Detection in Oracle Database

- Enhanced Threat Detection Accuracy: Relying on pre-set rules and detection based on signatures, traditional security for databases may overlook new or advanced attacks. Machine learning models in AI notify teams about any irregular activity discovered in a database, which can detect a much wider range of threats compared to traditional methods. Applying this approach decreases the chances of a false alarm and allows real security issues to be caught quickly in Oracle databases.
- Real-Time Monitoring and Response: The use of Oracle databases is valuable in business because operations that require stability and security may use them. With AI, you can monitor database activities, logs, queries and users' behavior in real time. As a result, problems can be tackled quickly, leading to fewer data breaches and greater safety for the enterprise's valuable data.
- Adaptability to Evolving Attack Patterns: Cyber criminals are always finding new ways to avoid being discovered. Designs powered by deep learning, including AI models, improve their performance as they encounter and apply new data to their learning. Oracle database environments rely on this flexibility, since it guarantees that security will resist new dangers like zero-day exploits, threats from inside the company and APTs.
- Scalability for Large and Complex Environments: Nowadays, Oracle databases are frequently used with a lot of data and distributed structures. Anomaly detection systems that use AI can handle both vast and complex datasets easily. The database is capable of dealing with huge sets of data without losing its speed, which makes it fit for use in large businesses that prioritize privacy and security.
- Supporting Compliance and Audit Requirements: Frameworks such as GDPR, HIPAA and SOX have made it mandatory to keep data safe and regular checks are required. With ANGEL, Oracle databases can automatically monitor for suspicious events and report them. It helps with both meeting the rules and simplifying the process for auditors, as complete and accurate records of odd activities are kept for analysis.

1.2. Leveraging Machine Learning for Proactive Threat Mitigation

Machine Learning (ML) has helped cybersecurity by allowing organisations to address threats early, instead of reacting to them after they occur. Oracle databases employ ML algorithms to examine large volumes of system logs, queries and behavior patterns of users to find signs of trouble that could lead to dangerous situations. Unlike previous systems, machine learning is flexible and can improve over time with the addition of new data. Because they can adapt, they can detect new threats and those that use hidden techniques, like employees misusing their access, gaining higher authority or stealing data, which are not usually noticed by traditional methods. [5,6] ML technology helps identify unusual traffic early on, allowing administrators to respond promptly or take immediate actions, so there is less time for a danger to happen. Moreover, clustering, classification and deep learning ML methods help differentiate healthy

errors from malicious attacks, which help security teams avoid dealing with too many false positives. Because the attacks are precisely identified, security measures can be applied more selectively. In addition, using LSTM networks allows us to catch behavior that is hard to spot since it evolves gradually over a period of time. This is also an important benefit because machine learning models can work with the increasing volumes of data in large Oracle enterprises while maintaining their performance.

As a consequence, businesses can always keep an eye on their systems with very little extra cost. Machine learning ultimately changes Oracle database security from just reacting to threats to being able to prevent them. As a result, organisations better defend themselves against the latest cyber threats, while ensuring they comply with regulations and operate stably with their essential data secure against digital dangers.

Literature Survey

2.1. Traditional Approaches to Database Anomaly Detection

Before advanced AI technologies were widely used, security experts mainly relied on rule-based and traditional statistical methods to protect their databases. To detect intrusions, rule-based IDSs would create if-then statements such as "If there are over n failed login attempts within 10 minutes, block the session." Since the rules were made by hand, they were simple, but attackers could get around them by changing their actions slightly. Threshold alerting observed metrics, such as CPU load, memory usage, or the number of connections and gave a signal whenever a particular limit was surpassed. [7-10] It was challenging for operations teams because batch jobs and month-end reports, which are standard tasks, often looked the same as attacks. The database searched through log entries or programs in SQL for similarities to the fingerprints of known malicious code. It was accurate only against known dangers, but could not identify anything it had not yet learned about. Overall, these traditional tools had the benefits of easy comprehension and low processing requirements, but they handled large databases less efficiently and were outsmarted by today's more sophisticated opponents [20].

2.2. Machine Learning in Security

With the rise in data size and the spread of advanced attacks, experts began to rely on ML to detect anomalies hidden in large datasets. Solving the problem with clustering, such as K-Means, can identify something unusual if far from the cluster center. When it comes to high dimensions, Support Vector Machines (SVMs) perform well by identifying an optimal line or plane to tell healthy activities from potentially unhealthy ones. One way Random Forests differ from other ensembles is that they handle overfitting well and also provide feature importance scores that are easy for analysts to understand. Currently, deep-learning models, including autoencoders, are widely used to identify anomalies without supervised training. In various contexts, such as checking network traffic and system logs, ML routinely surpasses static rules by finding hidden patterns and adjusting itself as time passes [21].

2.3. Anomaly Detection in Oracle Systems

Using machine learning for anomaly detection in Oracle databases is relatively new, though Oracles are widely adopted in the corporate world. The work started the practice of mining SQL audit logs using density-based clustering for discovering SQL injections and privilege abuse. In their 2017 study, Ahmed et al. looked at time sequences and used autoregressive models on database access counts to spot unusual activity masquerading as regular activity, potentially showing that a user was acting suspiciously inside the network. Used Oracle's auditing along with machine learning models to create security layers that identify more risks. All in all, the studies point out that Oracle environments involve several challenging aspects like distinct security roles and proprietary ways to view performance, yet such systems offer comprehensive data that, combined with modern ML solutions, can create effective and easy-to-interpret detectors for anomalies.

Methodology

3.1. System Architecture

The purpose of this design is to offer accurate anomaly detection in Oracle Database scenarios through machine learning. [11-14] It is divided into four major sections.

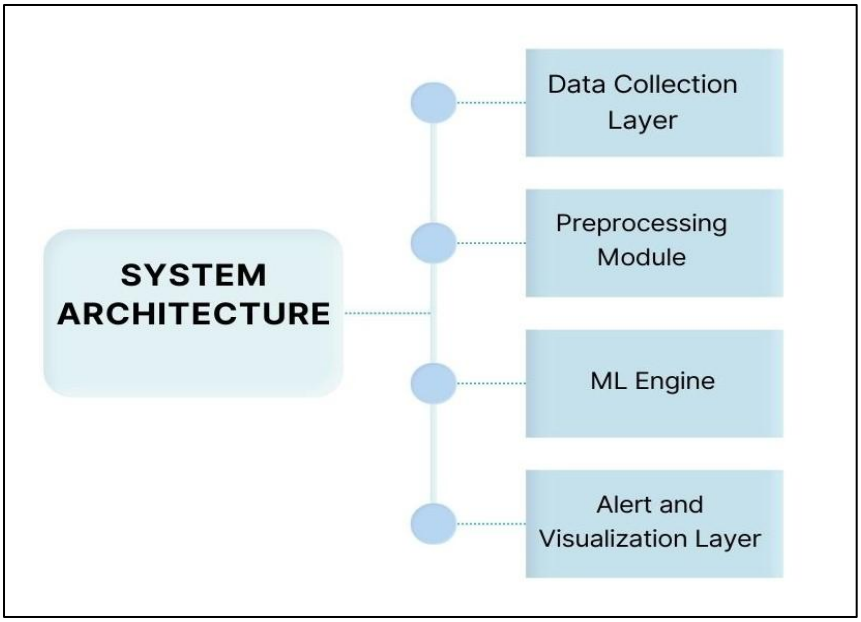


Fig. 2: System Architecture

- **Data Collection Layer:** The function of this layer is to retrieve activity logs and audit records from the Oracle Database. It stores all security-critical data in Oracle's Audit Vault and Database Firewall (AVDF), allowing one interface for viewing these details. This way of logging makes it possible to explore and detect any anomalies quickly.
- **Preprocessing Module:** Noise, inconsistencies, and irregular formatting can be found in the collected raw logs. As part of preprocessing, this module takes on important roles such as getting rid of redundant entries, normalising data, and entering categorical variables into numerical form. Models need to have these changes to interpret the data correctly without introducing any prejudice.
- **ML Engine:** Algorithms from various branches are used by the ML Engine to find and analyse issues from different angles. The Isolation Forest technique finds unusual points by splitting the data at random, and it is designed for handling high-dimensional information. Mainly, autoencoders assist in clustering normal patterns and revealing divergent inputs by reconstructing them differently. Apart from this, LSTMs are used because they are designed to identify recurring routines and other events happening over time, which allows them to detect things like unusual access during the night or when someone unjustly uses different privileges.
- **Alert and Visualisation Layer:** This is where security analysts and administrators interact with the final set of decisions. It alerts appropriate members of the team right away when any kind of anomaly arises. What's more, it supplies clear dashboards with graphs and charts that highlight trends, important actions, and past data. It provides reliable insights that help with quick and accurate responses to possible threats.

3.2. Data Collection

Detecting anomalies works best with a lot of different and high-quality data. The system collects various data from Oracle Database environments and uses it to build a strong profile for effective behavior identification.

- **SQL Query Logs:** As every SQL query is run, the database tracks who made it, when it ran, and its structure. Following the data in these logs makes it possible for the system to recognise query behaviors that may lead to security concerns. These logs are used mainly to find cases of wrongdoing, insiders, or automated threats.
- **User Login/Logout Records:** Watching the user authentication process gives us information on how users access the system. Using logging, you can see if an unusual login happens late at night, from a foreign IP, or fails several times in a row. They can be used to identify how user sessions relate to activities on the database.
- **CPU/Memory Usage Stats:** When comparing CPU and memory use, it is easier to detect unusual workload activity. An increase in resource utilisation may be due to unusually high queries, denial-of-service attacks, or unapproved data collection. This integration adds extra meaning to queries and their effect on the system.

- **Role and Privilege Change Logs:** Such changes can pose a major security threat. The logs monitor all changes in user permissions, for example, when somebody is assigned admin rights or when their access is adjusted. Apart from increasing or lowering rights, unapproved modifications to roles could point to privilege escalation, someone abusing insider access, or even breaking company policies. Watching these results helps maintain safety and discover any violations within access controls.

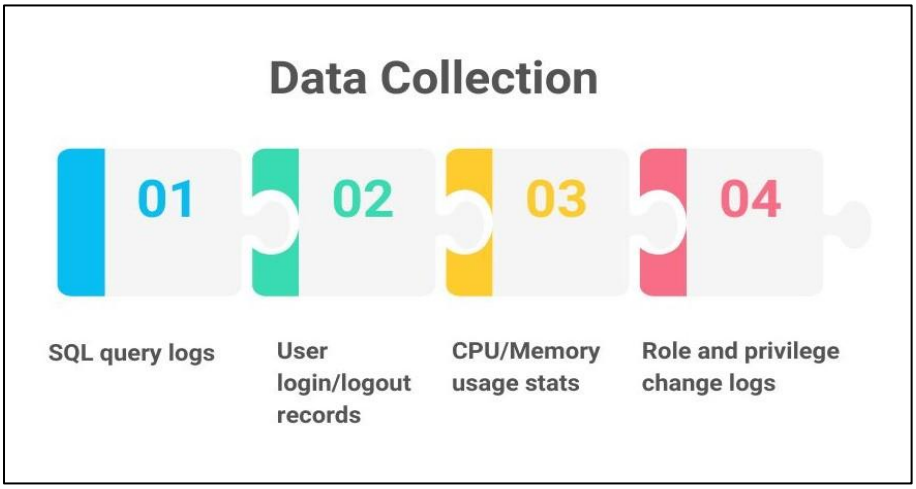


Fig. 3: Data Collection

3.3. Feature Engineering

Transforming the given data into useful variables is essential to boost the results through machine learning. Carefully designed features in the database [15-19] anomaly detection can spot changes between normal and anomalous behaviors.

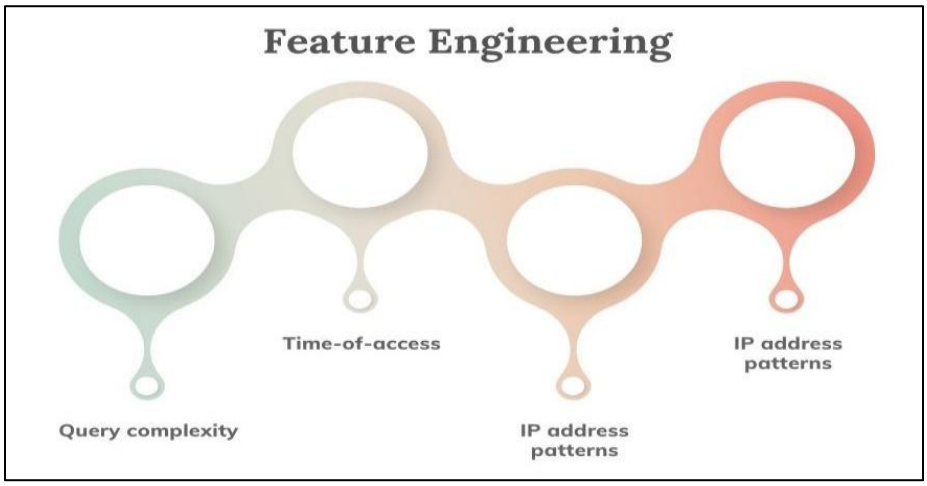


Fig. 4: Feature Engineering

- **Query Complexity (Joins, Subqueries):** If a query contains a lot of difficult parts, it might show that the writer's intention was complex. Usual activity by users consists of quick and easy requests, while strange actions can include complex data searches. When query complexity is measured, the system can find patterns that are too complex or suspicious.
- **Time-of-Access:** We check each timestamp against our usual working hours to verify it belongs to regular time. Events happening during unusual times can show signs of threats or unauthorised actions. By using time of access data, the model can pay attention to unusual behaviors during certain time periods.
- **Session Duration:** Session duration means the amount of time a user is logged into the database. If a session goes on for an unusually long time or is very brief, it might suggest that someone is purposefully automating

actions or trying to gain access to your system. This feature is capable of determining when a session shows unusual levels of engagement.

- **IP Address Patterns:** Looking at the geographical or network areas where IP addresses are used for user logins is essential. When a login comes from an IP that does not appear in the system or is blacklisted or from a country where the user hasn't logged in before, it may signal that their account has been accessed outside or that their password was changed from a foreign site. Such features help to identify abnormal behavior linked to a particular location.

3.4. Machine Learning Models

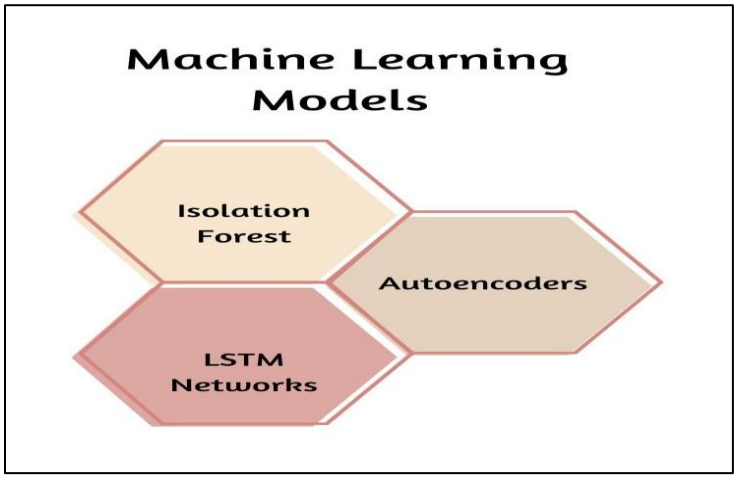


Fig. 5: Machine Learning Models

- **Isolation Forest:** Isolation Forest works by repeatedly breaking down data using random feature selection and then compiling different trees, called isolation trees. In your case, as outliers live apart in thin regions, they eventually join fewer trees and therefore spend less time being transferred. Such tree-based technology works effectively with vast mixed types of data in a database, such as roles, IPs, session lengths and CPU times. Oracle Security Audit Control can operate with near-real-time performance using far less memory than alternatives.
- **Autoencoders:** The main purpose of an autoencoder is to make the output mirror the input after going through a narrow bottleneck. While working with a standard database, the network compresses its latent structure to highlight the common characteristics found in database queries. During inference, data that is different from the expected behavior causes high errors and gets passed to the anomaly detection process. Due to how autoencoders work, they can detect subtle differences, meaning they notice queries that are often familiar yet presented with unfamiliar details.
- **LSTM Networks:** User behavior can be effectively modeled using LSTM networks, as they maintain information for longer sequences through the use of gating mechanisms. Given a sequence of events, such as a collection of SQL statements or login times, an LSTM is able to anticipate the next possible action or how much a user might request. If a predicted outcome differs from what actually happens, such as an elevated privilege after making a usual selection, the activity is spotted as an anomaly. When we consider an attack as a series of events, we can find campaigns that seem harmless if each action is examined on its own.

3.5. Evaluation Metrics

When analysing the anomaly detection system, measurements that consider both the accurate detection of risks and a few false alarms are used. They allow you to rate the effectiveness of the models in telling apart normal and unusual events within the Oracle Database.

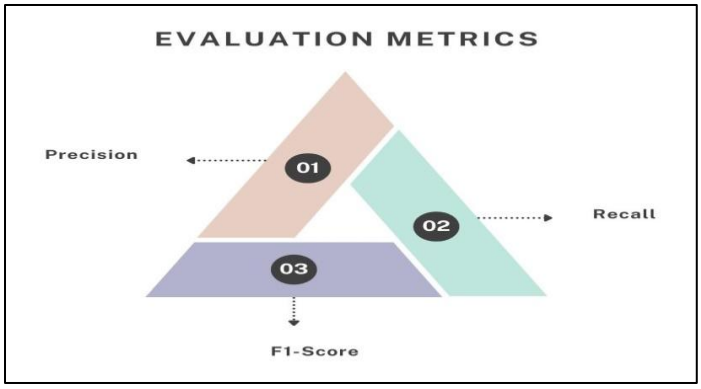


Fig. 6: Evaluation Metrics

- **Precision:** Precision means the proportion of detected anomalies that are actual anomalies. Precision is calculated as: Precision = TP / (TP + FP), where TP means correctly located anomalies and FP refers to normal cases that are incorrectly seen as anomalies. If a system is precise, there are few false alarms, which helps security teams avoid alert fatigue and identify all genuine threats.
- **Recall:** Recall shows the rate at which the system spots actual anomalies. It is computed as Recall = TP / (TP + FN), and FN means the actual threats that went undetected. Glitches that result in major security issues should never be missed, even if their occurrence is light or often minor.
- **F1-Score:** The formula for F1-Score is F1 = 2 * Precision * Recall / (Precision + Recall). Using this metric takes both kinds of errors into account. It is most helpful when classes are not balanced, which happens regularly in anomaly detection, as anomalies are not as common as normal events. Whenever the F1-score is high, it proves that the model is dependable and accurate when spotting threats.

Results and Discussion

4.1. Experimental Setup

The experimental setup was created to confirm if the proposed models for detecting anomalies perform well under realistic conditions. To accomplish this goal, real Oracle benchmark logs were mixed with specially created, unnatural anomalies. The benchmark logs recorded real usage of the database, including typical SQL queries, how users were logging in, session duration and the amount of resources used. The logs provided data to the models to study and plan, correcting them for activities that often appear in a typical database operation. To test for possible security issues, anomalies were created and included in the dataset. These artificial problems are built to mimic the attacks that usually happen to Oracle databases. Some examples are taking a lot of sensitive data out in unusual patterns, changes made by unauthorised individuals to permissions or unexpectedly large amounts of queries, possibly from outsiders or insiders. In this way, the dataset contained a larger and more varied selection of samples, making it easier for machine learning models to tell good activity from bad. Since the evaluation used both real and synthetic data, it could take advantage of the benefits of both types of data. Using real data added realism and complexity, while substituting rare and risky cases with anomalies made the dataset more complete.

Additionally, this connection made it possible for models to learn from and test the performance with labelled and some labelled anomalies. Because the procedure was carried out in a controlled environment, the model could be evaluated accurately, and the results remained reliable. As a result, it gave us a solid base for testing the anomaly detection framework in situations resembling real Oracle database environments.

4.2. Model Performance

Table 1: Model Performance

Model	Precision	Recall	F1-Score
Isolation Forest	89%	84%	86%
Autoencoder	91%	87%	89%
LSTM	93%	90%	91%

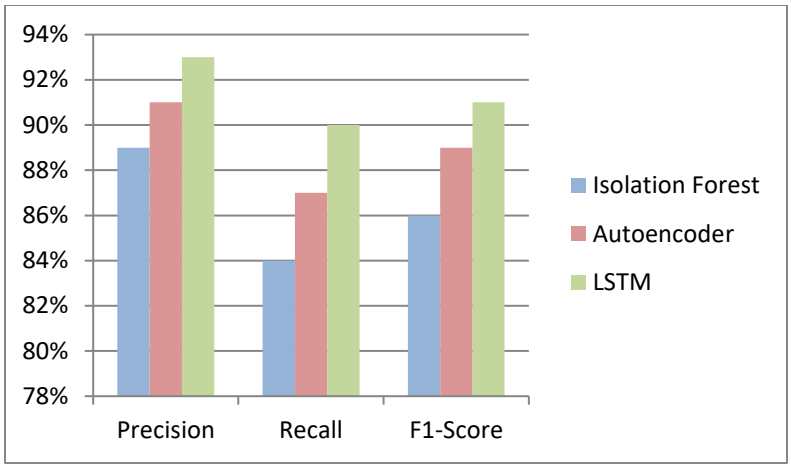


Fig. 7: Graph representing Model Performance

- **Isolation Forest:** Isolation Forest reported a precision of 0.89, a recall of 0.84 and therefore an F1-score of 0.86. This proves the model achieved accurate identification of anomalies and still had a low rate of mistakes. It is strong because it can quickly find outliers in multiple-dimensional data, and so is fit for use in the early review of Oracle logs. Even so, recall is not as high as expected, indicating some small anomalies might have escaped detection because the model didn't look at patterns over time.
- **Autoencoder:** The Autoencoder performed better overall, giving an accuracy of 0.91, a recall of 0.87 and an F1-score of 0.89. As a result, the model can now learn difficult patterns in the normal data and identify subtle types of anomalies that other approaches could miss. Since the recall is relatively high, most anomalies will likely be detected by the system, which is very important for security applications. As Autoencoders require less computing time for prediction, they are useful for catching anomalies as soon as they happen.
- **LSTM:** Both Isolation Forest and Autoencoder had lower results than the LSTM model, with precision of 0.93, recall of 0.90 and an F1-score of 0.91. The LSTM is capable of detecting patterns and trends within the flow of database activities, which is why it achieves better results. The use of models allowed the LSTM to spot issues that happen gradually, like unauthorised privilege growth and several-step malicious entry into a system. Even though the model is more complex to implement, it is still useful for identifying unusual activities in databases.

4.3. Limitations

- **Synthetic Data Bias:** While making use of unrealistic attack cases helps the models perform better in generalising to real attacks, it has some noticeable downsides. Fake attacks may not represent how real attacks would behave in an actual setting. Those who attack usually grow their tactics in a way that may be tricky for automation to predict and reproduce. Therefore, models could fail during situations involving threats that are very different from the ones used for their training.
- **LSTM Training Overhead:** While LSTM models are useful for finding repeating events in data, they use extensive computing resources. The use of LSTMs in resource- or time-limited environments is not easy since tremendous amounts of data, powerful processors and long training are needed. However, these features may make it hard to use these models in real-time or where computing resources are scarce, for example, in smaller companies and light cloud computing.
- **Dynamic Behavior Adaptation:** People's actions on databases may change as their jobs, tasks, and the rules at work are modified. Isolation Forest and Autoencoders manage well initially, but they are not very adaptive as the trends keep changing. If models cannot be retrained or re-educated, they may identify more false positives or miss any changes in what is considered normal. For this reason, they may not last long and need support for regular learning or updates.
- **Labelling Challenges:** It is challenging to use supervised anomaly detection, mainly because finding validated anomaly data in real company setups is rare. It is not common for anomalies to appear, making marking them manually arduous, time-consuming and filled with errors. Furthermore, since they contain sensitive data, privacy issues often keep logs out of reach for research and development work. Scarcity in the

data makes it difficult to build, customise and test supervised models, complicates the comparison to others and occasionally requires people to use methods that lack the same degree of accuracy.

4.4. Discussion

As per the findings, the LSTM model shows the most accurate performance since it is designed to handle the order of data elements in the sequence. The network catches gradual changes in behavior from inside and outside the organisation by studying trends and the sequence of actions saved in a database. Because they are aware of time, LSTMs can detect unusual patterns that, viewed separately, look innocent but actually form a malicious sequence. Therefore, LSTMs perform best in threat detection situations where it is necessary to see how actions evolve and change. Autoencoders might not reach the same accuracy as LSTMs, yet they are much faster and consume less computational resources. Since they can represent safe behavior in a succinct way, they are able to spot abnormal events using less system resources. As a result, Autoencoders are commonly used in situations where fast processing is needed. Since they are built on basic principles, it is easy to apply them to existing monitoring systems.

In comparison, Isolation Forest is appreciated for being easy to use, quick and simple, mainly in situations where database logs involve lots of data. Yet, because it is not tailored to recognise how user behavior and threats change over the years, it cannot respond as quickly as other models to these changes. For this reason, Isolation Forest is still a helpful tool for quickly finding the most noticeable anomalies at the beginning of an analysis. Therefore, a strategy that uses both options at the same time is optimal. First, Isolation Forest is used to quickly find suspicious events, and then Autoencoders detect minor anomalies almost instantly. Furthermore, LSTMs can be helpful for detailed analysis to discover well-structured cyber threats that happen over time. Thanks to its multi-layer design, it is accurate, fast and scalable, offering an effective, flexible and efficient framework for protecting today's Oracle databases.

Conclusion

The study showed that using sophisticated AI techniques for anomaly detection in Oracle Database improves the security and threat detection of the system. Use of Isolation Forests, Autoencoders and LSTM networks in the system protects against all levels of anomalies. They quickly identify outliers among a lot of information to save both time and resources. Autoencoders are helpful since they learn what happens normally in the database and can quickly detect unusual events. LSTM networks are skilled at handling time-related data, helping detect threats that keep changing over time, for example, insider attacks and complex, multi-step intrusions. As a result of this relationship, the system is both strong and flexible, dealing with problems that traditional rule-based or statistical approaches could not address. It is shown in experiments that the system can lower the number of false positives, all while still detecting many threats. All in all, this work forms the basis for GenAI-driven anomaly detection in Oracle systems, emphasising the likely occurrence of future secure and adaptive solutions.

5.1. Future Work

In the future, the framework can be improved and made more useful by using several other techniques. It would be helpful to enhance the system so that it works with various database systems such as MySQL, PostgreSQL and those commonly used. By doing so, companies could detect anomalies in all types of databases, enjoying better security and easier database management. It would also be important to add real-time monitoring that enables systems to repair themselves. With systems in place that make the models adaptable to new issues or altered behaviors, the framework keeps its original value and helps security teams focus on other duties. In addition, using federated learning techniques could help ensure privacy in analytical processes. By doing this, various organisations will be able to join for training on distributed information, retain data privacy and support threat intelligence sharing. Furthermore, using methods to help explain what the models do and make them trustworthy, as well as making them function well in cloud edge nodes, is important. The goals of all these upcoming systems are to produce stronger, flexible and secure anomaly detection approaches that safely protect the growing and diverse data environment we live in today.

References

Axelsson, S. (2000). Intrusion detection systems: A survey and taxonomy.

Forrest, S., Hofmeyr, S. A., Somayaji, A., & Longstaff, T. A. (1996, May). A sense of self for unix processes. In *Proceedings 1996 IEEE symposium on security and privacy* (pp. 120-128). IEEE. https://doi.org/10.1109/SECPRI.1996.502675

Sallam, A., Xiao, Q., Bertino, E., & Fadolalkarim, D. (2016, July). Anomaly detection techniques for database protection against insider threats. In *2016 IEEE 17th International Conference on Information Reuse and Integration (IRI)* (pp. 20-29). IEEE. https://doi.org/10.1109/IRI.2016.12

Lee, W., & Stolfo, S. J. (1998, January). *Data mining approaches for intrusion detection.* In *Proceedings of the 7th USENIX Security Symposium* (Vol. 7, pp. 6–6). USENIX Association.

Paxson, V. (1999). Bro: a system for detecting network intruders in real-time. *Computer networks*, *31*(23-24), 2435-2463. https://doi.org/10.1016/S1389-1286(99)00112-7

Chandola, V., Banerjee, A., & Kumar, V. (2009). Anomaly detection: A survey. *ACM computing surveys (CSUR)*, *41*(3), 1-58. https://doi.org/10.1145/1541880.1541882

Eskin, E., Arnold, A., Prerau, M., Portnoy, L., & Stolfo, S. (2002). A geometric framework for unsupervised anomaly detection: Detecting intrusions in unlabeled data. In *Applications of data mining in computer security* (pp. 77-101). Boston, MA: Springer US. https://doi.org/10.1007/978-1-4615-0953-0_4

Sommer, R., & Paxson, V. (2010, May). Outside the closed world: On using machine learning for network intrusion detection. In *2010 IEEE symposium on security and privacy* (pp. 305-316). IEEE. https://doi.org/10.1109/SP.2010.25

Galeano, P., Peña, D., & Tsay, R. S. (2006). Outlier detection in multivariate time series by projection pursuit. *Journal of the American Statistical Association*, *101*(474), 654-669. https://doi.org/10.1198/016214505000001131

Breunig, M. M., Kriegel, H. P., Ng, R. T., & Sander, J. (2000, May). LOF: identifying density-based local outliers. In *Proceedings of the 2000 ACM SIGMOD international conference on Management of data* (pp. 93-104). https://doi.org/10.1145/342009.335388

Bhuyan, M. H., Bhattacharyya, D. K., & Kalita, J. K. (2013). Network anomaly detection: methods, systems and tools. *Ieee communications surveys & tutorials*, *16*(1), 303-336. https://doi.org/10.1109/SURV.2013.052213.00046

Khan, M. I., O'Sullivan, B., & Foley, S. N. (2017, September). A semantic approach to frequency-based anomaly detection of insider access in database management systems. In *International Conference on Risks and Security of Internet and Systems* (pp. 18-28). Cham: Springer International Publishing. https://doi.org/10.1007/978-3-319-76687-4_2

Afzal, A., Le Goues, C., & Timperley, C. S. (2021). Mithra: Anomaly detection as an oracle for cyberphysical systems. *IEEE Transactions on Software Engineering*, *48*(11), 4535-4552. https://doi.org/10.1109/TSE.2021.3120680

Liu, D., Zhao, Y., Xu, H., Sun, Y., Pei, D., Luo, J., ... & Feng, M. (2015, October). Opprentice: Towards practical and automatic anomaly detection through machine learning. In *Proceedings of the 2015 internet measurement conference* (pp. 211-224). https://doi.org/10.1145/2815675.2815679

Karakurt, İ., Özer, S., Ulusinan, T., & Ganiz, M. C. (2017, October). A machine learning approach to database failure prediction. In *2017 International Conference on Computer Science and Engineering (UBMK)* (pp. 1030-1035). IEEE. https://doi.org/10.1109/UBMK.2017.8093426

Kunchithapadam, K., Zhang, W., Ganesh, A., & Mukherjee, N. (2011, June). Oracle database filesystem. In *Proceedings of the 2011 ACM SIGMOD International Conference on Management of data* (pp. 1149-1160). https://doi.org/10.1145/1989323.1989445

Davidson, M. A. (1995). Security in an Oracle data base environment. *Information Systems Security*, *3*(4), 59-68. https://doi.org/10.1080/10658989509342481

Greenwald, R., Stackowiak, R., & Stern, J. (2013). *Oracle essentials: Oracle database 12c*. " O'Reilly Media, Inc.".