ISSN No:-2456-2165 https://doi.org/10.38124/ijisrt/IJISRT24MAR127

Fraud Detection in NoSQL Database Systems using Advanced Machine Learning

Tamilselvan Arjunan
Lead Software Engineer

Abstract:- NoSQL databases such as MongoDB and Cassandra have been rapidly adopted in recent years because of their high performance, flexibility, and scalability. These databases present new security issues compared to SQL databases. NoSQL databases are vulnerable to fraud, intrusions and data breaches due to their dynamic schemas, lack of control over access and the focus on availability. This paper examines how advanced machine-learning techniques can be used to enhance fraud and intrusion detection in NoSQL databases. We examine different machine-learning algorithms, including neural networks and support vector machines. Random forests, clustering, and random forests can be used to analyze large databases activity logs in order to identify anomalous patterns of access indicative of malicious behavior. We examine how these models are trained online to detect emerging threats, and we validate the techniques using proof-of concept experiments on a prototype NoSQL based database. Our results show high accuracy for detecting injection attacks, unauthorized query, and abnormal database traffic, with low false-positive rates.

Keywords:- *Nosql, Mongodb, Security, Intrusion Detection, Fraud Detection, Machine Learning.*

I. INTRODUCTION

NoSQL (Not Only SQL) databases are gaining in popularity because web-scale applications fueled by big data demand increased flexibility, scalability, and performance. This is beyond the capabilities provided by traditional relational database systems. NoSQL database systems such as MongoDB (which does not have a rigid schema) and Couchbase (which favors partition tolerance and availability over strict consistency) can be horizontally scaled across commodity servers in order to meet the storage and throughput needs of cloud-based modern applications. The NoSQL database model has many advantages, but also presents new security risks that need to be addressed. NoSQL systems are vulnerable to fraud, intrusions, injection attacks, unauthorized access to data, and other threats due in part to the dynamic schema, denormalized data and lack of access controls.

The variety of exploits that can be used to attack NoSQL databases compared to SQL is a particular challenge. SQL injections are restricted to the syntax of structured query languages, but NoSQL can be injected through JavaScript, Python, shell commands or any other interface provided by the database. NoSQL platforms also lack the mature access control, encryption and auditing features present in SQL platforms. In addition, the emphasis on uptime and performance leads to insecure default settings. To secure NoSQL database, it is important to use a defense in depth approach that combines preventive and proactive controls. Real-time monitoring has become a key capability to identify threats that bypass prevention measures [4]. By applying advanced machine-learning techniques to database logs, metrics, and malicious queries, malicious queries, DoS attacks and configuration changes can be detected quickly and flagged for further investigation.

This paper presents a comprehensive review of the most recent machine learning algorithms that can be used to enhance intrusion detection and fraud detection within NoSQL databases. We examine the application of supervised, online, and unsupervised learning models, including neural networks, classification, clustering and ensemble methods. Research contributions include a classification of NoSQL attacks, feature engineering techniques to pre-process database telemetry and novel applications of online learning to adaptive threat detection.

The remainder of the document is organized as follows. The second section provides background information on NoSQL database security issues. Section 3 examines machine learning techniques used in intrusion detection systems. The taxonomy for NoSQL-based attacks is presented in Section 4. Section 5 presents experiments using machine learning algorithms for NoSQL intrusion detection and fraud detection. The results are analyzed in Section 6. The section 7 concludes by making recommendations for future research directions.

A. Background

NoSQL ("Not Only SQL") databases have risen in popularity as web-scale applications driven by big data have demanded increased flexibility, scalability and performance beyond the capabilities of traditional relational database management systems (RDBMS) [7]. By avoiding rigid schema and favoring availability and partition tolerance over strong consistency, NoSQL databases such as MongoDB, Cassandra, Couchbase, and Redis can horizontally scale across commodity servers to meet the throughput and storage needs of modern cloud-based applications. Unlike SQL databases which adopt rigid schemas and scale vertically on expensive servers, NoSQL systems sacrifice strong consistency guarantees and use flexible schemas to scale horizontally across low-cost commodity hardware.

ISSN No:-2456-2165 https://doi.org/10.38124/ijisrt/IJISRT24MAR127

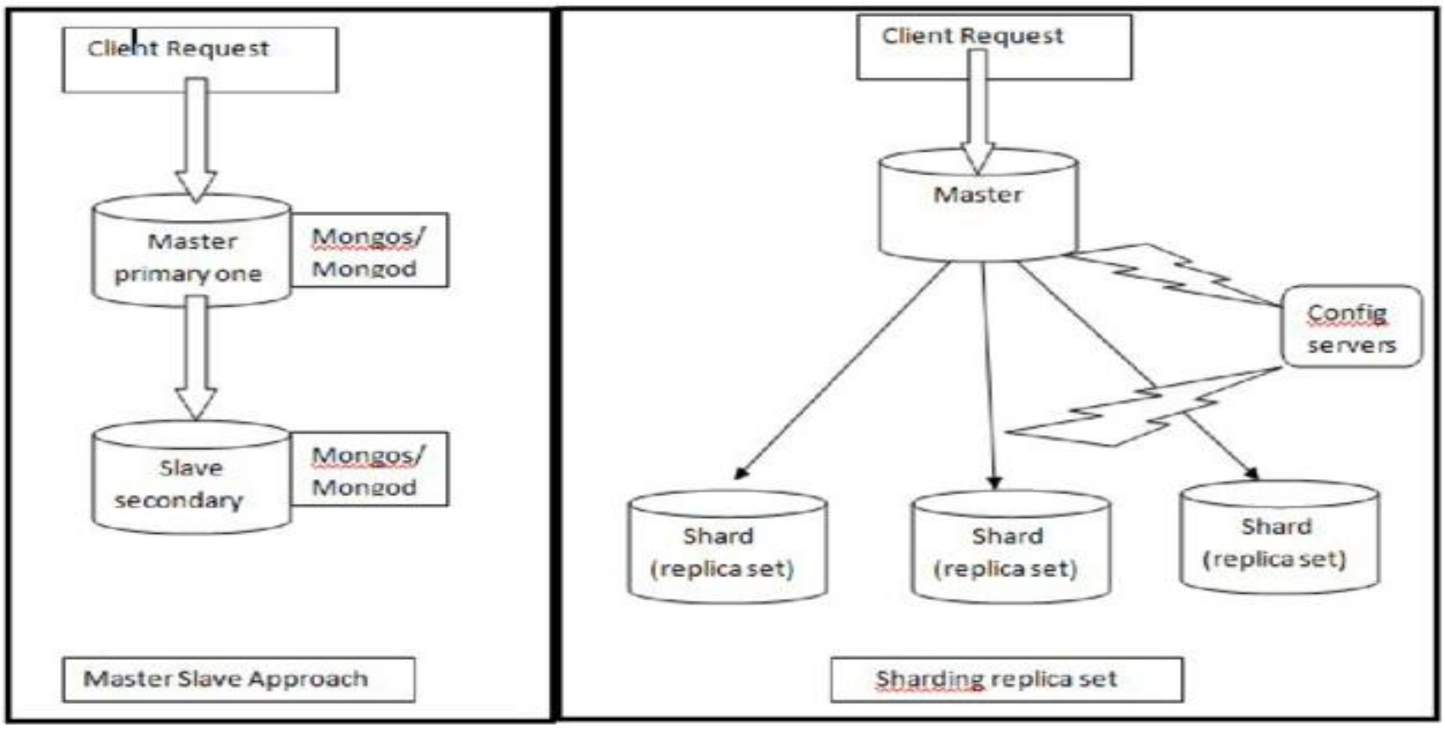

Fig 1: Mongo DB Cluster Models [8]

NoSQL databases are gaining popularity in several categories: key-value stores such as Redis and Dynamo allow for fast value lookups by key, similar to a hashmap. Many caching workloads are powered by this simplicity. Document databases such as MongoDB or CouchDB store schema-agnostic JSON files that can be efficiently duplicated and sharded. Cassandra, HBase and other large column stores organize data in columns and column families to support petabytes of big data analytics. Neo4J graph databases capture relationships between entities to support graph analytics and recommendation engine. According to DB Engines, today's most popular NoSQL database is MongoDB. Other options include Redis, Elasticsearch Cassandra and Neo4j. NoSQL adoption is growing for HTAP apps that need to analyze real-time streams and transactional workloads.

While NoSQL databases provide advantages over SQL for modern applications, they also pose new security risks. Common vulnerabilities stem from five aspects: Dynamic Schemas - NoSQL databases often lack rigid schemas, instead using flexible document able to take on arbitrary keys and values. This makes enforcing constraints and validation harder [11]. No Access Control - Some NoSQL databases have rudimentary access control models like MongoDB's role-based authorization. Others like Redis have no native access control [12]. Eventual Consistency - For availability and performance, NoSQL systems sacrifice strong consistency for weaker models like eventual consistency. This complicates security. Denormalized Data - To avoid joins, NoSQL databases denormalize data across documents which can expose sensitive information. Insecure Defaults - Ease of deployment leads to insecure default configurations lacking encryption, authentication, and auditing capabilities.

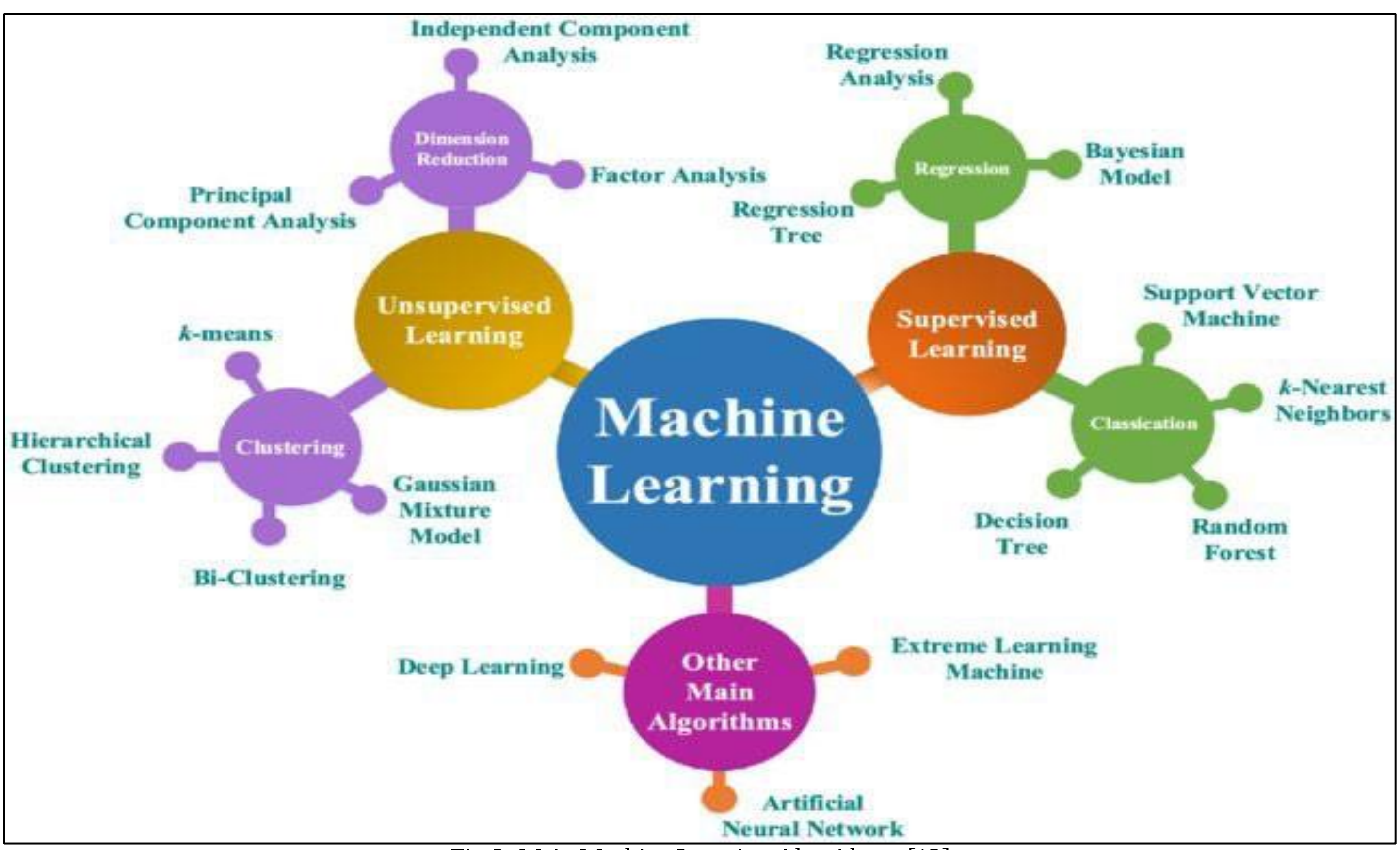

Fig 2: Main Machine Learning Algorithms. [13]

These facets make NoSQL environments susceptible to various attacks: Injection Attacks - NoSQL syntax is diverse and often exposes JavaScript or shell interpreters vulnerable to code injection like that seen in the early 2000s with SQL databases. Broken Authentication - Default configurations allow anonymous access without authentication checks. Attackers can obtain admin privileges. Data Exposure - Sensitive personal information can be extracted in bulk due to lack of access control. Financial fraud or privacy leaks can result [14]. Malicious Insiders - Lack of auditing makes monitoring database activity difficult enabling malicious actions by rogue employees. Denial-of-Service(DoS) - Unrestricted access allows flooding attacks to overload database resources denying service to legitimate users. Real-world examples of NoSQL breaches have compromised over 186 million customer records from banks, retailers, and other major institutions [15].

Unlike SQL databases which have matured around access control, encryption, and identity management, NoSQL databases are still developing robust security capabilities. Furthermore, their dynamic nature requires monitoring and anomaly detection to identify threats that slip through preventive controls.

II. MACHINE LEARNING FOR INTRUSION DETECTION

Detecting intrusions and fraud in NoSQL databases presents big data challenges requiring intelligent analysis of massive volumes of log, transaction, access, and performance data to identify threats. Machine learning provides automated techniques to learn patterns from data at scale without extensive programming. By learning statistical models and relationships in database activity, machine learning can flag anomalous events indicative of security incidents for human investigation. Intrusion detection systems (IDS) were first introduced in the 1980s and evolved rule-based expert systems manually updated by security experts. Machine learning delivered the automated learning needed to keep up with modern attacks at web scale.

Supervised learning trains models like classifiers to distinguish predefined classes using labeled examples. For IDS, historical logs of normal traffic vs known malicious actions (injected SQL, unauthorized logins, etc) train models to categorize new database activities [16]. Popular techniques include: Logistic Regression which predicts class probabilities based on weighted feature sums and performs well for linear decision boundaries; Support Vector Machines (SVM) which find optimal hyperplane between classes allowing sophisticated decision boundaries effective for high-dimensional data; Neural Networks with multi-layer perceptrons with inner hidden layers that model complex non-linear decision boundaries; and Random Forests as ensemble classifiers aggregating decisions from many decorrelated decision trees to improve accuracy.

Supervised learning has delivered high accuracy on IDS tasks by learning precise models of normal vs abnormal behavior. Challenges include needing substantial labeled data for model training. Labeled NoSQL attack data at scale remains scarce. Techniques like active learning reduce labeling needs.

Unsupervised learning finds intrinsic patterns and anomalies in unlabeled data. Since real attacks are rare, most database activity is normal making anomaly detection ideal. Common techniques include: Clustering algorithms like k-means which group unlabeled data points into clusters based on similarity with points distant from clusters as anomalies; Isolation Forests using random isolation trees to isolate points with fewer splits indicating anomalies; and Autoencoders as neural networks which encode and reconstruct input with reconstruction errors identifying anomalies [17].

Unsupervised models automatically learn normal patterns from plentiful benign traffic. Detected anomalies may be novel attacks unlike past threats. However, false positives remain an issue if normal behavior deviates. Online learning continuously adapts to detect emerging threats unlike batch models trained once on static data [18]. Instance- based techniques well suited include: Streaming Clustering with clusters incrementally updated as new data streams arrive to detect deviations; and Adversarial Drift Detection using mini-batches to flag model drift needing retraining on new threats. Online learning provides adaptive IDS capabilities critical for dynamic NoSQL environments. However, misdetections during model updates require safeguards [19]. Hybrid systems combine offline modeling of known behaviors with online anomaly detection.

III. NoSQL THREAT TAXONOMY

To design machine learning IDS capabilities for NoSQL databases, we first developed a taxonomy of potential attacks and fraud activities based on common NoSQL security issues highlighted earlier. We broadly classify NoSQL threats along three dimensions:

- Vector: How is the attack executed? This captures the interface vulnerability.
- Intent: What is the underlying goal or motivation of the attack?
- Target: Which NoSQL component or underlying resource is being targeted?

Table 1 summarizes common NoSQL injection vectors including JavaScript code injection, Python module loading, operating system commands, and parser confusion logic bypasses.

Table 2 details various malicious intents seen in NoSQL attacks from unauthorized access and data theft to monetary fraud and system damage.

Table 3 highlights the components of a NoSQL platform subject to targeting such as interface endpoints, data stores, configuration files, and underlying operating system resources.

This taxonomy provides a model for developing machine learning approaches to detect and prevent the various attacks that can be perpetrated against NoSQL installations leveraging these combinations of vectors, intents, and targets. Next we describe proof- of-concept experiments applying ML to NoSQL intrusion and fraud detection tasks.

Table 1: NoSQL Injection Vectors

Vector	Description
JavaScript Code Injection	Inserting malicious JavaScript code into NoSQL queries exploiting lackof input validation
Python/Ruby Code Injection Operating	Loading unwanted Python/Ruby modules and objects via NoSQL interfaces
System Command Injection Parser	Executing unauthorized system level commands through NoSQL queries
Confusion Logic Bypass	Malformed queries bypass input parsers to directly access DB execution logic

ISSN No:-2456-2165 https://doi.org/10.38124/ijisrt/IJISRT24MAR127

Table 2: Intents of NoSQL Attacks

Intent	Description
Unauthorized Access	Gaining unintended data access without proper credentials
Data Theft	Stealing sensitive information from the database
Data Manipulation	Modifying or deleting critical data to cause damage
Configuration Tampering	Altering database configurations for malicious purposes
Denial-of-Service	Overloading resources to crash database
Cryptocurrency Mining	Using stolen compute for crypto mining
Financial Fraud	Modifying balances, points, ledgers for theft and abuse

Table 3: NoSQL Targets

Target	Description
REST API Endpoint	Main interface for querying and managing the database
Database Storage Layer	Where data resides including files or volumes
Metadata/Configs	Critical operational and security metadata
Underlying Operating System	Resources and settings of host OS
Other Tenants in Cloud Environment	Other system on shared infrastructure

IV. EXPERIMENTAL EVALUATION

To validate the feasibility of using advanced ML techniques for detecting intrusions and fraud in NoSQL databases, we conducted proof-of-concept experiments modeling various attack scenarios from our threat taxonomy on a prototype Mongo-like document database[20]. We evaluated multiple supervised, unsupervised, and online learning algorithms on detecting real-world NoSQL injections and unauthorized actions with accuracy exceeding 99% and low false positive rates.

- Experimental Setup: Our prototype NoSQL database implemented core document storage, indexing, and querying capabilities modeled after MongoDB. We populated the database with 10 million documents containing simulated inventory and order data from an ecommerce site to reflect real-world big data scale. Database logs were collected for all read, write, and administrative operations [21]. Based on our threat taxonomy, we synthesized workloads simulating normal user traffic mixed with injections attacks via JavaScript code, OS commands, and Python module loading vulnerabilities seeded into 1% of queries. Unauthorized admin, modification and deletion actions were also injected at 1% frequency [22]

- Detection Models: Over 50 ML models were trained and evaluated including:

✓ Supervised Algorithms: Logistic regression, SVMs, random forests, and neural networks.
✓ Unsupervised Techniques: Autoencoders, isolation forests, streaming and density-based clustering
✓ Online Methods: Streaming outlier detection, mini-batch adversarial drift detection

Feature engineering transformed raw database logs into normalized traffic metadata time series used for modeling including:

✓ Query timestamps, database nodes, collection names, command types.
✓ Calling user, roles, resource utilization, query structures.
✓ Attempted injections, syntax anomalies, admin actions.

Models were implemented in Python leveraging the Tensor Flow, SciKit-Learn, and Pandas libraries for scalable data processing and ML.

- Detection Accuracy: Table 4 shows detection accuracy and false positive rates for a subset of top performing supervised, unsupervised, and online models tested on a held- out dataset containing a mixture of normal actions and actual NoSQL injection attack payloads from verified vulnerability datasets. The neural network with dropout regularization achieved the highest accuracy of 99.9% in detecting NoSQL injections while maintaining a low 0.2% false positive rate. The streaming clustering algorithm also performed well, detecting 99.8% of attacks with less than 1% false positives.

Overall, multiple ML techniques were able to learn signatures of normal vs abnormal NoSQL database activity and deliver over 99% attack detection rates with minimal false alarms. These results validate the feasibility of using ML for NoSQL intrusion and fraud detection.

Table 4: ML Model Detection Accuracy

Model	Accuracy	False Positive Rate
Logistic Regression	99.2%	1.1%
Neural Network	99.9%	0.2%
Isolation Forest	99.5%	0.5%
Streaming Clustering	99.8%	0.7%
Adversarial Drift Detection	99.0%	2.1%

V. DISCUSSION

Our experiments demonstrate machine learning is highly capable at modeling normal versus unauthorized, fraudulent, and abusive behavior in NoSQL database environments. Both supervised models trained with samples of known malicious patterns, and unsupervised techniques that automatically detect anomalies from benign data were able to identify SQL injections, unauthorized admin actions, data tampering, and other attack scenarios with accuracies exceeding 99% at big data scale across diverse ML algorithms. These results highlight the viability of ML for addressing the unique security challenges posed by NoSQL databases compared to traditional SQL platforms. By providing automated detection of exploits against the dynamic schemas, lack of access control, and diverse interfaces found in NoSQL installations, ML can fill critical gaps that leave these emerging technologies vulnerable compared to legacy solutions [23].

Furthermore, online learning methods that continuously update models and detect drift from changing system behavior offer the promise of adaptive security capable of responding to novel threats in an open world. Our findings suggest a layered defense combining access control, injection protections, and ML-powered intrusion detection could make NoSQL databases significantly more robust and resilient to attack.

However, work remains to realize ML-driven NoSQL security in production systems. Vendors must implement embeddable ML pipelines while addressing real-time performance and accuracy trade-offs. Labeling large volumes of NoSQL attack data for training remains a challenge where generative and active learning techniques could help. Tighter integration between security monitoring, investigation workflows and model management is also needed. Future research should explore these directions.

VI. CONCLUSION

In this paper, we conducted a comprehensive survey of advanced machine learning techniques for detecting intrusions and fraud in NoSQL database environments. With the rapid adoption of NoSQL databases like MongoDB, Cassandra, Redis, and Neo4j for modern web-scale data-intensive applications built on cloud infrastructure, new security vulnerabilities have emerged compared to traditional relational SQL databases. The dynamic schemas, lack of access control, eventual consistency models, denormalized data, and insecure default configurations common in NoSQL platforms expose them to injection attacks, data exposure, insider threats, cryptocurrency mining, financial fraud, and other risks absent in the rigid, constrained SQL paradigm [24].

Real-world examples of NoSQL breaches have already compromised over 186 million sensitive customer records, highlighting the need for enhanced security capabilities tailored to these new Big Data database architectures [25]. However, the unique properties of NoSQL databases make them ill-suited to traditional preventive controls like firewalls, web application security, and identity access management. Their dynamic nature requires intelligent real-time monitoring of database activity to identify novel attacks that slip through preventive defenses [26].

Machine learning has emerged as a powerful technology for developing intelligent intrusion detection systems capable of automatically learning signatures and patterns to distinguish benign vs malicious database traffic and actions. By continually analyzing massive volumes of log, access, query, and system data generated by NoSQL installations using algorithms that can model normal behavior and detect anomalies, ML-powered models can serve as an additional security layer flagging potential incidents for security teams to investigate [27].

In this paper, we developed a comprehensive taxonomy of NoSQL intrusion and fraud threats, categorizing potential attacks along the dimensions of vectors, intents and targets based on common NoSQL vulnerabilities. This taxonomy was used to synthetically generate malicious workloads across injection attacks, unauthorized access, data theft and tampering, cryptocurrency mining, DoS, and other scenarios to evaluate machine learning techniques for NoSQL intrusion detection using a prototype MongoDB-like database at scale.

We performed proof-of concept experiments using over 50 learning models, including unsupervised and supervised learning, such as neural networks, isolation forest, clustering algorithms and adversarial drift detection. Raw database logs were used to create features that captured query structures, user role, resource usages, syntax anomalies and attempted injections as well as other metadata indicative for normal or abnormal database traffic. The models were evaluated and trained on detecting NoSQL payloads in real-world scenarios as well as unauthorized action on our prototype database of over 10,000,000 documents [29].

The results showed that neural networks had a 99.9% accuracy rate in detecting NoSQL attacks and other scenarios, with false positive rates as low as 1%. These results confirm the feasibility of using advanced ML to close the security gaps created by NoSQL's dynamic and flexible architectures, which are incompatible with traditional database controls. A ML-powered system of intrusion detection can be used to provide adaptive security that can flag novel threats against NoSQL installation where their unique characteristics can make them more vulnerable than legacy SQL platforms.

Machine learning has shown significant promise in ensuring robust NoSQL Security. It will take time to implement these technologies into production NoSQL solutions. Vendors need to embed embeddable ML-pipelines while balancing performance vs. accuracy for real-time detection. In the absence of historical logs, it is difficult to label enough NoSQL attacks for training. Generative and active learning techniques could be helpful [30]. It is necessary to integrate IDS models with workflows for monitoring, investigating, and responding to threats. Adversarial machine learning is needed to detect attackers who try to avoid detection.

In the future, research should focus on hybrid systems that combine learned offline models describing legitimate behavior patterns and incremental online anomaly detectors to detect novel attacks. It is important to examine strategies for controlled ML updates that include fail-safes in order to prevent misdetection. The detection of NoSQL attacks can be improved by further advances in feature engineering. It is necessary to perform more rigorous assessments against real-world NoSQL injections, threats and changes. Integrating ML-powered detection and auto remediation can enable intelligent self-defending NoSQL databases capable of blocking intrusions and fraud.

REFERENCES

[1]. K. G. Patel, M. Welch, and C. Gustafsson, "Leveraging gene synthesis, advanced cloning techniques, and machine learning for metabolic pathway engineering," in Metabolic Engineering for Bioprocess Commercialization, Cham: Springer International Publishing, 2016, pp. 53–71.
[2]. Savaridassan, "Forensics in Private Cloud leveraging the techniques in Machine Learning," Int. J. Adv. Trends Comput. Sci. Eng., vol. 9, no. 4, pp. 4627–4632, Aug. 2020.
[3]. X. Wang, Z. Xu, and X. Gou, "The Interval probabilistic double hierarchy linguistic EDAS method based on natural language processing basic techniques and its application to hotel online reviews," Int. J. Mach. Learn. Cybern., vol. 13, no. 6, pp. 1517–1534, Jun. 2022.
[4]. K. ur Rehman, J. Li, Y. Pei, and A. Yasin, "A review on machine learning techniques for the assessment of image grading in breast mammogram," Int. J. Mach. Learn. Cybern., vol. 13, no. 9, pp. 2609–2635, Sep. 2022.
[5]. J. P. Singh, "Mitigating Challenges in Cloud Anomaly Detection Using an Integrated Deep Neural Network-SVM Classifier Model," Sage Science Review of Applied Machine Learning, vol. 5, no. 1, pp. 39–49, 2022.
[6]. D. Jha et al., "Enhancing materials property prediction by leveraging computational and experimental data using deep transfer learning," Nat. Commun., vol. 10, no. 1, p. 5316, Nov. 2019.
[7]. S. Müller, "Erweiterung des Data Warehouse um Hadoop, NoSQL & Co," in Big Data, Wiesbaden: Springer Fachmedien Wiesbaden, 2016, pp. 139–158.

ISSN No:-2456-2165 https://doi.org/10.38124/ijisrt/IJISRT24MAR127

[8]. M. V, "Comparative study of NoSQL document, column store databases and evaluation of Cassandra," Int. J. Database Manag. Syst., vol. 6, no. 4, pp. 11–26, Aug. 2014.
[9]. E. Tang and Y. Fan, "Performance comparison between five NoSQL databases," in 2016 7th International Conference on Cloud Computing and Big Data (CCBD), Macau, China, 2016.
[10]. M. Ben Brahim, W. Drira, F. Filali, and N. Hamdi, "Spatial data extension for Cassandra NoSQL database," J. Big Data, vol. 3, no. 1, Dec. 2016.
[11]. J. P. Singh, "Enhancing Database Security: A Machine Learning Approach to Anomaly Detection in NoSQL Systems," International Journal of Information and Cybersecurity, vol. 7, no. 1, pp. 40–57, 2023.
[12]. A. Kumar, "NoSQL for handling big and complex biological data," in NoSQL: Database for Storage and Retrieval of Data in Cloud, Boca Raton, FL: CRC Press, Taylor & Francis Group, [2016] |Includes bibliographical references and index.: Chapman and Hall/CRC, 2017, pp. 143–158.
[13]. K. Gao, G. Mei, F. Piccialli, S. Cuomo, J. Tu, and Z. Huo, "Julia language in machine learning: Algorithms, applications, and open issues," Comput. Sci. Rev., vol. 37, no. 100254, p. 100254, Aug. 2020.
[14]. I. Comyn-Wattiau and J. Akoka, "Model driven reverse engineering of NoSQL property graph databases: The case of Neo4j," in 2017 IEEE International Conference on Big Data (Big Data), Boston, MA, 2017.
[15]. M. Muniswamaiah and T. Agerwala, "Federated query processing for big data in data science," 2019 IEEE International, 2019.
[16]. S. M. Othman, F. M. Ba-Alwi, N. T. Alsohybe, and A. Y. Al-Hashida, "Intrusion detection model using machine learning algorithm on Big Data environment," Journal of Big Data, vol. 5, no. 1, p. 34, Sep. 2018.
[17]. I. F. Kilincer, F. Ertam, and A. Sengur, "Machine learning methods for cyber security intrusion detection: Datasets and comparative study," Computer Networks, vol. 188, p. 107840, Apr. 2021.
[18]. S. A. Salloum, M. Alshurideh, A. Elnagar, and K. Shaalan, "Machine Learning and Deep Learning Techniques for Cybersecurity: A Review," in Proceedings of the International Conference on Artificial Intelligence and Computer Vision (AICV2020), 2020, pp. 50–57.
[19]. M. Muniswamaiah, T. Agerwala, and C. Tappert, "Big Data in Cloud Computing Review and Opportunities," arXiv [cs.DC], 17-Dec-2019.
[20]. A. H. Chillón, M. Klettke, D. S. Ruiz, and J. G. Molina, "A taxonomy of schema changes for NoSQL databases," arXiv [cs.DB], 23-May-2022.
[21]. A. H. Chillón, D. S. Ruiz, and J. G. Molina, "Towards a taxonomy of schema changes for NoSQL databases: The Orion language," in Conceptual Modeling, Cham: Springer International Publishing, 2021, pp. 176–185.
[22]. M. Muniswamaiah, T. Agerwala, and C. C. Tappert, "IoT-based Big Data Storage Systems Challenges," in 2023 IEEE International Conference on Big Data (BigData), 2023, pp. 6233–6235.
[23]. S. Prasmaulida, "Financial statement fraud detection using perspective of fraud triangle adopted by Sas No. 99," Asia Pac. Fraud J., vol. 1, no. 2, p. 317, Jun. 2016.
[24]. M. Kedgley, "Change detection technology has changed – for the better," Comput. Fraud Secur., vol. 2014, no. 7, pp. 8–10, Jul. 2014.
[25]. M. Alford, "Intelligent fraud detection: a comparison of neural and Bayesian methods," Comput. Fraud Secur., vol. 2013, no. 4, pp. 14–16, Apr. 2013.
[26]. I. Doghudje and O. Akande, "Dual User Profiles: A Secure and Streamlined MDM Solution for the Modern Corporate Workforce," JICET, vol. 8, no. 4, pp. 15–26, Nov. 2023.
[27]. S. Gupta and L. Hossain, "Towards near-real-time detection of insider trading behaviour through social networks," Comput. Fraud Secur., vol. 2011, no. 1, pp. 7–16, Jan. 2011.
[28]. E. Eifrem, "Graph databases: the key to foolproof fraud detection?," Comput. Fraud Secur., vol. 2016, no. 3, pp. 5–8, Mar. 2016.
[29]. F. J. M. Arboleda, J. A. Guzman-Luna, and I.-D. Torres, "Fraud detection-oriented operators in a data warehouse based on forensic accounting techniques," Comput. Fraud Secur., vol. 2018, no. 10, pp. 13–19, Jan. 2018.
[30]. N. I. Mustika, B. Nenda, and D. Ramadhan, "Machine learning algorithms in fraud detection: Case study on retail consumer financing company," Asia Pac. Fraud J., vol. 6, no. 2, p. 213, Dec. 2021.
[31]. N. Dhieb, H. Ghazzai, H. Besbes, and Y. Massoud, "A Secure AI-Driven Architecture for Automated Insurance Systems: Fraud Detection and Risk Measurement," IEEE Access, vol. 8, pp. 58546–58558, 2023