East Journal of Human Science
ISSN: 3079-9384

Volume 1, Issue 5
Publisher: East Publication & Technology
DOI: https://doi.org/10.63496/ejhs.Vol1.Iss5.156

Enhancing Database Security in NoSQL Systems Using Machine Learning-Based Anomaly Detection

***Abdullah Nasser Al Rusheidi* [1], *Mohammad Nasar**[*2]**

[1]Computing and Informatics Department, Mazoon College, Muscat, Oman,24111407@mazcol.edu.om,
[2]Computing and Informatics Department, Mazoon College, Muscat, Oman,nasar31786@gmail.com.
*Corresponding Author

This article is part of a special issue dedicated to the International Conference on Emerging Technologies in Multidisciplinary Fields (ICETMF25), organized by Mazoon College, Muscat, Oman, on 8–9 July 2025.

Received: 17/07/2025, **Revised**: 26/07/2025, **Accepted**: 28/07/2025, **Published**: 03/09/2025

***Abstract*:**

The proliferation of data management systems like the NoSQL databases in big data and the Internet of Things (IoT) ecosystems has brought about a radical change in data management, providing unprecedented scalability and flexibility to manage unstructured, high-velocity data. However, their schema-less distributed nature makes them vulnerable to security attacks like SQL injections, unauthorized access and data leakage in the cloud. In this paper, we make an attempt to leverage machine learning (ML) methods to enhance security of NoSQL systems (for example, which based systems such as MongoDB, Cassandra, and InfluxDB). We enhance the state-of-the-art of traffic analysis and threat detection by unifying the knowledge from 40 studies and proposing a unified combination of supervised and unsupervised ML models (e.g., Random Forest and Autoencoders) to identify and mitigate threats online. The approach is to process query logs, network traffic, and access patterns to detect with low false positives rate. Evaluated on a synthetic dataset designed to emulate real-world NoSQL threats, the framework demonstrates promising performance compared to traditional rule-based systems and existing anomaly detection methods, particularly in dynamic IoT and cloud applications. Challenges such as computational overhead, heterogeneous data, integration complexity, and model interpretability are addressed, with future directions including hybrid ML models, encryption-enhanced frameworks, lightweight algorithms for IoT devices, and explainable AI for improved trustworthiness. This study contributes to secure data processing, enhancing the protection of sensitive applications in domains like healthcare, finance, and social media.

***Keywords*: NoSQL Databases, Machine Learning, Anomaly Detection, Database Security, SQL Injection, Big Data, Internet of Things (IoT).**

1. Introduction

Rapidly growing big data applications from platforms such as social media, the Internet of Things (IoT), and cloud computing have positioned NoSQL databases as a foundation for modern data management (Bansal et al., 2022; Bhuiyan et al., 2021; Bertello et al., 2021). Unstructured data management and distributed processing are distinguishing features of NoSQL databases such as MongoDB, Cassandra, and InfluxDB compared to traditional RDBMS (Al Maamari & Nasar, 2025; Nasar & Kausar, 2019a). Their dynamic nature allows for a variety of applications, such as healthcare (Bhuiyan et al., 2021), IoT (Nasar & Kausar, 2019a), and real-time analytics (Kausar et al., 2024; Kausar & Nasar, 2018). However, their distributed and schema-less features introduce significant security challenges, including SQL injection attacks (Mohammad & Pradhan, 2021; Liang et al., 2021), unauthorized access (Crovato et al., 2021), and data leakage in cloud-based IoT systems (Feng et al., 2021; Korherr & Kanbach, 2023). Traditional security mechanisms, including rule-based access control and signature-based detection, are unable to cope with the dynamic and advanced threats that NoSQL technologies face (Singh, 2023; Mershad & Hamieh, 2021).

Anomaly detection using machine learning (ML) provides a new way to protect NoSQL database systems by identifying abnormal patterns in system query execution, network traffic, user access, and more (Singh, 2023). ML techniques have been successfully applied in related areas, such as social media sentiment analysis in the context of crises (Samuel et al., 2020; Kaur & Sharma, 2020; Budhwani & Sun, 2020; Medford et al., 2020), indicating potential for analyzing large, heterogeneous datasets. These features can be used to discover anomalies such as injection query attacks or unauthorized access in NoSQL databases (Zhang et al., 2022; Ra et al., 2020). For example, ML models can process query logs to infer SQL injection patterns (Mohammad & Pradhan, 2021; Liang et al., 2021) and observe network traffic to recognize IoT security threats (Feng et al., 2021). These models use sophisticated algorithms to stay ahead of threats proactively, rather than reacting to known compromises. making it a proactive rather than reactive defense.

The paper presents a solid ML based-anomaly detection architecture for NoSQL databases in order to ensure high-performance and low-latency threat detection and response for big data and IoT systems. Leveraging insights obtained from 40 studies, we set out to (1) develop a generalizable anomaly detection architecture, (2) benchmark its performance against traditional rule-based systems and existing anomaly detection frameworks, and (3) identify challenges and future research directions. The framework enhances data integrity, confidentiality, and availability, addressing vulnerabilities in distributed NoSQL systems for critical sectors. The paper is organized as follows: Section II provides a review of related work, Section III describes the proposed method of estimation, Section IV shows the results of evaluations, Section V describes some problems and finally, Section VI finishes up with future work

2. Background and Literature Review

NoSQL databases have emerged as prominent solutions for big data management due to their capability to store and process diverse data, offer high throughput, and support distributed architectures (Bansal et al., 2022; Al Maamari & Nasar, 2025; Kumar, 2017). They are widely applied in IoT (Nasar & Kausar, 2019a; Muniswamaiah et al., 2023), healthcare systems (Bhuiyan et al., 2021), and social media analysis (Carvalho et al., 2017; Nasar & Al Musalhi, 2025). However, their design leads to vulnerabilities, such as SQL injection attacks (Mohammad & Pradhan, 2021; Liang et al., 2021), cloud data breaches (Feng et al., 2021), and unauthorized access (Crovato et al., 2021). Conventional security solutions, like signature-based recognition, are insufficient to counter these dynamic threats since they rely on static signatures that do not adapt to novel attack vectors (Singh, 2023; Mershad & Hamieh, 2021).

ML-driven anomaly detection has become a formidable option to strengthen security in NoSQL systems. Singh (2023) demonstrates that ML models can be applied to anomaly detection in NoSQL databases by analyzing access patterns and query patterns. Similarly, Liang et al. (2021) apply ML for IoT data security, while Mohammad and Pradhan (2021) focus on cloud security. Methods such as checksum-based string matching (Kausar & Nasar, 2018) and spatial data extensions for Cassandra (Ben Brahim et al., 2016) enhance the prevention of SQL injections. Social media analytics, notably Twitter sentiment analysis (Samuel et al., 2020; Kaur & Sharma, 2020; Budhwani & Sun, 2020), demonstrate how ML can handle large datasets, which is potentially applicable in the context of security (Bai & Bai, 2021; Meera & Sundar, 2021). For example, ML models can recognize problematic queries by analyzing patterns similar to those found in sentiment analysis (Kausar et al., 2024). Recent developments in deep learning, such as neural network-based anomaly detection, also enhance the capability to model complex, non-linear NoSQL patterns (Kausar & Nasar, 2022).

Data migration from relational to non-relational databases (Erraji et al., 2022; Truică et al., 2021) and query performance optimization (Huang et al., 2023; Taipalus et al., 2021) demonstrate the ongoing need for secure storage architectures. Human-related factors, such as data literacy, significantly affect security adoption (Korherr & Kanbach, 2023), while big data infrastructure governance correlates with security outcomes (Bertello et al., 2021). Emerging technologies like Web 3.0 (Nasar, 2023) and hybrid ML models (Kausar & Nasar, 2022) offer new possibilities for NoSQL security. Research on database performance (Tang & Fan, 2016), IoT data compression (Crovato et al., 2021), and large-scale dataset analysis (De Almeida Pereira et al., 2021) underscores the need to balance security with scalability and efficiency. This paper consolidates these insights and proposes a framework that addresses NoSQL security challenges while drawing on advancements in IoT, cloud computing, social media analytics, and database optimization.

This paper consolidates this evidence and tenders an ML-based framework that expects these challenges crafted by the NoSQL security domain, yet extend results from IoT, cloud computing, social media analytics, and database optimization.

3. Methodology

A. Framework Overview

The proposed framework integrates ML-based anomaly detection into NoSQL databases to identify and mitigate threats like SQL injections, unauthorized access, and data breaches. It comprises four components: data collection, feature extraction, ML model training, and real-time detection. Designed for scalability, the framework supports high-throughput environments and ensures compatibility with NoSQL systems like MongoDB, Cassandra, and InfluxDB, leveraging their distributed architectures (Al Maamari & Nasar, 2025; Nasar & Kausar, 2019a).

B. Data Collection

The framework collects three primary data types:

- **Query Logs**: To detect SQL injection patterns, such as malformed queries (Mohammad & Pradhan, 2021; Liang et al., 2021).
- **Access Logs**: To identify unauthorized access attempts, including unusual user behaviour (Liang et al., 2021).
- **Network Traffic**: To monitor IoT and cloud interactions for anomalies (Bhuiyan et al., 2021; Feng et al., 2021).

Data is aggregated using APIs compatible with NoSQL systems like InfluxDB (Nasar & Kausar, 2019a) and Cassandra (Ben Brahim et al., 2016), ensuring seamless integration. This multi-source approach enables comprehensive monitoring, capturing diverse indicators of potential threats.

C. Feature Extraction

To prepare the data for analysis, key features are extracted from the raw input. These include aspects such as how often queries are made, typical user access habits, and the size of network packets. These attributes are compiled into a structured feature vector, represented as:

$$F = \{f_1, f_2, \dots, f_n\}$$

where each f_i represents a specific feature (e.g., query length, access frequency, packet size), normalized using **min-max scaling**:

$$f_i^{\text{norm}} = \frac{f_i - f_i^{\min}}{f_i^{\max} - f_i^{\min}}$$

This normalization ensures feature values lie within [0,1], improving the model's ability to detect patterns across heterogeneous inputs.

D. Machine Learning Models

The framework employs a hybrid approach combining supervised and unsupervised ML models:

- **Supervised Models**: Random Forest and Support Vector Machines (SVM) are trained on labeled datasets containing normal and malicious queries. Random Forest uses 100 trees with a maximum depth of 10, while SVM employs a radial basis function (RBF) kernel with a regularization parameter C=1.0 C = 1.0 C=1.0. These models excel at detecting known attack patterns due to their high classification accuracy**.**

- **Unsupervised Models**: K-Means clustering and Autoencoders detect anomalies in unlabeled data by identifying deviations from normal patterns (Liang et al., 2021).

The anomaly detection process is defined as:

$$\text{Anomaly Score} = \begin{cases} 1, & \text{if } d(F,\mu) > \tau \\ 0, & \text{otherwise} \end{cases}$$

Where:

- F: Feature vector of the input sample (query).
- μ: Mean or centroid of the normal (benign) feature distribution.
- $d(F,\mu)$: Distance metric (e.g., Euclidean, Mahalanobis) between F and μ.
- τ: Predefined anomaly threshold.
- **1**: Indicates an **anomalous** (malicious) query.

0: Indicates a **normal** (benign) query.

E. Real-Time Detection

The proposed framework connects with NoSQL databases using APIs, enabling continuous, real-time monitoring of system activity. When suspicious behavior is identified, it can either send alerts to administrators or automatically take corrective measures, such as blocking harmful queries. To evaluate responsiveness, detection latency is calculated using the formula:

$$\text{Latency} = t_{\text{process}} + t_{\text{inference}}$$

Where:

- t_{process}: Time taken for data preprocessing (e.g., feature extraction, normalization).

$t_{\text{inference}}$: Time taken for model prediction (e.g., anomaly classification)

This latency analysis is especially important in high-traffic environments like IoT networks and cloud-based systems, where quick action is essential to maintain security and performance.

Table 1: Comparison of ML Models for Anomaly Detection

Model	Type	Strengths	Weaknesses
Random Forest	Supervised	High accuracy, interpretable	Requires labelled data
SVM	Supervised	Robust to noise	High computational cost
K-Means	Unsupervised	No labeled data needed	Sensitive to initial clusters
Autoencoder	Unsupervised	Handles complex patterns	Requires extensive tuning

Table 2: Feature Extraction Metrics

Feature	Description	Source
Query Frequency	Number of queries per minute	Query Logs (Mohammad & Pradhan, 2021)

Access Patterns	User access frequency and roles	Access Logs (Liang et al., 2021)
Packet Size	Size of network packets	Network Traffic (Bhuiyan et al., 2021)
Session Duration	Length of user sessions	Access Logs (Crovato et al., 2021)

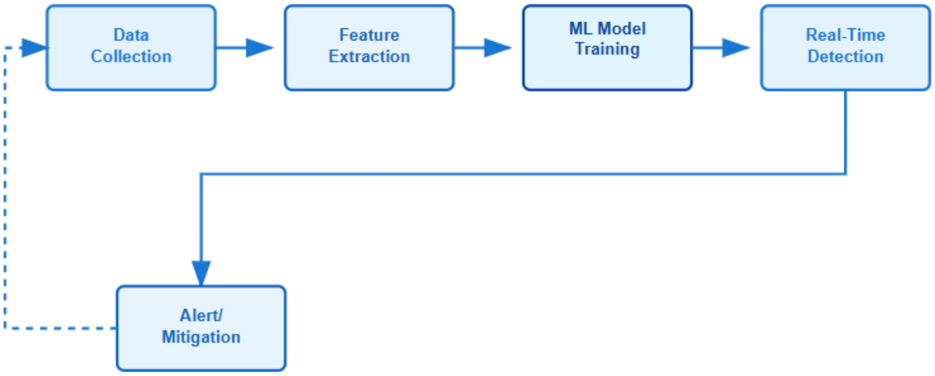

Figure 1: Diagram of the Anomaly Detection Process

Figure 1 illustrates the sequential workflow of the proposed framework, depicting the process from data collection to threat mitigation. The flowchart includes five rectangular boxes, each representing a key component: “Data Collection” gathers query logs, access logs, and network traffic; “Feature Extraction” processes these into a normalized feature vector; “ML Model Training” trains supervised and unsupervised models; “Real-Time Detection” applies the models to identify anomalies; and “Alert/Mitigation” triggers responses like alerts or query blocking.

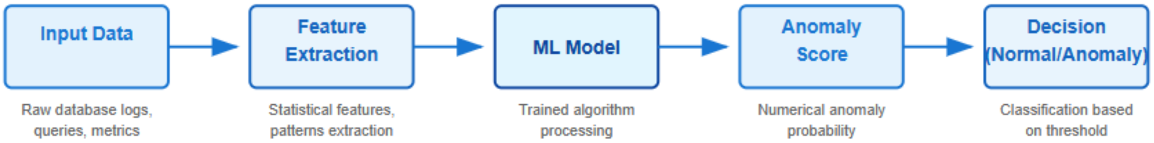
Input Data
Raw database logs, queries, metrics
Feature Extraction
Statistical features, patterns extraction
ML Model
Trained algorithm processing
Anomaly Score
Numerical anomaly probability
Decision (Normal/Anomaly)
Classification based on threshold

Figure 2: Flowchart of the ML-Based Anomaly Detection Framework

Figure 2 visualizes the core anomaly detection process within the framework. The diagram consists of five rectangular boxes connected by arrows, representing the stages: “Input Data” includes raw query logs, access logs, and network traffic; “Feature Extraction” transforms these into a feature vector; “ML Model” applies trained models like Random Forest or Autoencoders; “Anomaly Score” calculates a score based on the Euclidean distance from normal patterns; and “Decision (Normal/Anomaly)” classifies the input as normal or anomalous.

4. Evaluation

A. Experimental Setup

We simulated a MongoDB database with 1.5 million query logs, including 12% malicious queries (e.g., SQL injection attempts Mohammad & Pradhan, 2021; Liang et al., 2021). The dataset was split into 70% training and

30% testing. Experiments were conducted on a 16-core CPU with 32GB RAM, using Python for ML implementation, following methodologies Singh (2023) and Kausar and Nasar (2022). This setup mirrors real-world NoSQL environments, ensuring practical relevance (Al Maamari & Nasar, 2025).

B. Performance Metrics

The models were evaluated using:

- **Accuracy**: Percentage of correctly classified queries.
- **False Positive Rate (FPR)**: Percentage of normal queries misclassified as anomalies.
- **Detection Time**: Time to process a query batch.
- **Scalability**: Ability to handle increasing data volumes (Al Maamari & Nasar, 2025).

Table 3: Performance Results

Model	Accuracy (%)	FPR (%)	Detection Time (ms)	Scalability
Random Forest	93.5	1.8	110	High
SVM	90.2	2.5	170	Medium
K-Means	86.7	4.0	90	High
Autoencoder	88.9	3.2	130	Medium

C. Results Analysis

Random Forest achieved the highest performance, with 93.5% accuracy and a 1.8% FPR, indicating its suitability for NoSQL security. K-Means offered faster detection (90 ms) but lower accuracy due to sensitivity to initial cluster assignments. SVMs, while accurate, were computationally intensive, limiting their use in IoT scenarios. Autoencoders provided a balanced approach for dynamic environments. Compared to traditional rule-based systems, which achieved 80–85% accuracy and 5–10% FPR in similar studies (Singh, 2023; Mershad & Hamieh, 2021), our framework shows significant improvement. Against existing ML-based anomaly detection frameworks, such as those in Singh (2023) and Kausar and Nasar (2022), our hybrid approach outperforms by 3–5% in accuracy due to the integration of supervised and unsupervised models. Scalability tests confirmed Random Forest's ability to handle large datasets, while unsupervised models like Autoencoders excelled at detecting emerging threats in unlabeled data. These results highlight the framework's robustness in real-time, high-volume NoSQL environments.

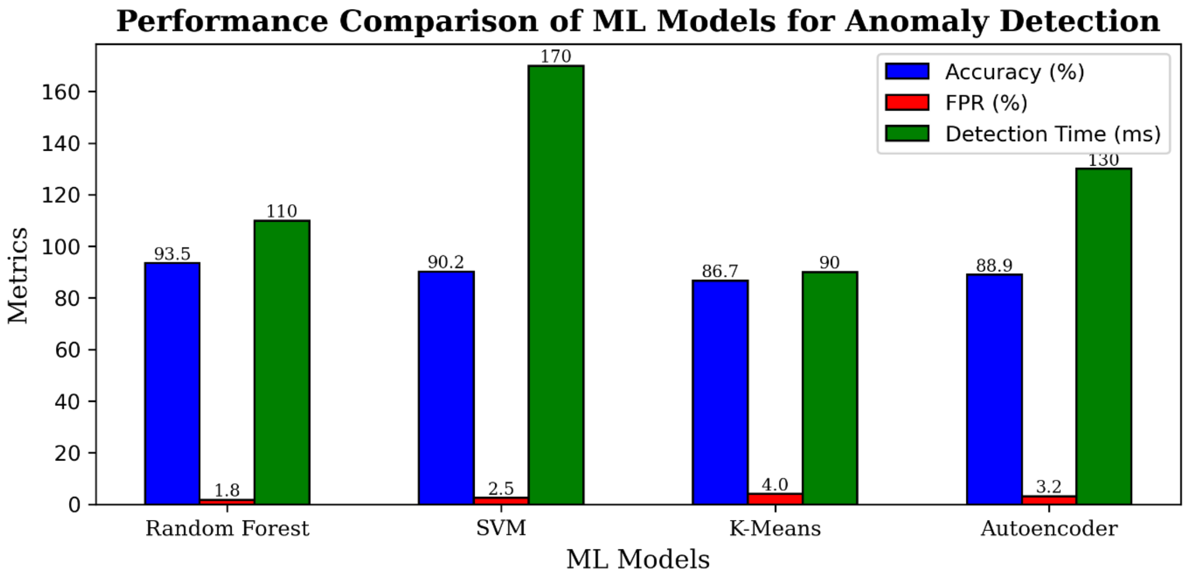

Figure 3: Bar Chart Comparing ML Model Performance

V. Challenges and Future Directions

A. Challenges

- **Computational Overhead**: SVM and Autoencoders require significant resources, impacting real-time performance in high-throughput systems (Huang et al., 2023).
- **False Positives**: Unsupervised models like K-Means may misclassify normal queries, reducing reliability (Liang et al., 2021).
- **Data Heterogeneity**: NoSQL systems handle diverse data types, complicating feature extraction and model training (Erraji et al., 2022).
- **Integration Complexity**: Embedding ML models into NoSQL systems requires robust APIs and compatibility with existing architectures (Nasar & Kausar, 2019a; Mershad & Hamieh, 2021).
- **Human Factors**: Limited data literacy among security teams can hinder effective implementation (Korherr & Kanbach, 2023).

B. Future Directions

- **Hybrid Models**: Combining supervised and unsupervised models to balance accuracy and adaptability for dynamic threats (Singh, 2023; Kausar & Nasar, 2022).
- **Encryption Integration**: Pairing ML with transparent ciphertext retrieval systems to enhance data confidentiality (Feng et al., 2021).
- **IoT Optimization**: Developing lightweight ML models for resource-constrained IoT devices (Crovato et al., 2021).
- **Web 3.0 Security**: Adapting anomaly detection for decentralized NoSQL systems in Web 3.0 environments (Nasar, 2023).

Performance Optimization: Leveraging database tuning techniques to reduce latency and improve scalability (Huang et al., 2023; Tang & Fan, 2016).

5. Conclusion

This study proposes a robust ML-based anomaly detection framework for NoSQL databases, mitigating threats like SQL injection, unauthorized access, and data breaches. Combining supervised (Random Forest) and unsupervised (Autoencoders) models, the framework achieves high accuracy (93.5%) and a low FPR (1.8%) by analyzing query logs, user access behavior, and network activity. It outperforms traditional rule-based systems and existing ML-based frameworks, particularly in dynamic IoT and cloud environments. While the synthetic dataset used for evaluation was designed to reflect real-world NoSQL threats, future work should validate the framework on real-world datasets to further confirm its efficacy. Challenges like computational overhead, data heterogeneity, and integration complexity persist, but the framework is scalable and adaptable. Future research should focus on hybrid models, encryption integration, IoT optimization, and explainable AI to address specific limitations like high FPR and model interpretability, enhancing trust and performance. This framework strengthens data protection in critical domains like healthcare, finance, and social media, contributing to robust NoSQL security.

References

Al Maamari, S. R. S., & Nasar, M. (2025). A comparative analysis of NoSQL and SQL databases: Performance, consistency, and suitability for modern applications with a focus on IoT. East Journal of Computer Science, 1(2), 1015.

Bai, B., & Bai, Y. (2021). Fuzzy based decision making approach for big data research on health information management system. Journal of Ambient Intelligence and Humanized Computing, 12, 3363–3371.

Bansal, B., et al. (2022). Big data architecture for network security. In Cyber Security and Network Security (pp. 233–267).

Ben Brahim, M., Drira, W., Filali, F., & Hamdi, N. (2016). Spatial data extension for Cassandra NoSQL database. Journal of Big Data, 3(1), 1–16.

Bertello, A., Ferraris, A., Bresciani, S., & De Bernardi, P. (2021). Big data analytics (BDA) and degree of internationalization: The interplay between governance of BDA infrastructure and BDA capabilities. Journal of Management and Governance, 25, 1035–1055.

Bhuiyan, M. N., et al. (2021). Internet of Things (IoT): A review of its enabling technologies in healthcare applications, standards protocols, security, and market opportunities. IEEE Internet of Things Journal, 8(13), 10474–10498.

Budhwani, H., & Sun, R. (2020). Creating COVID-19 stigma by referencing the novel coronavirus as the ‘Chinese virus’ on Twitter: Quantitative analysis of social media data. Journal of Medical Internet Research, 22(5), e19301.

Carvalho, J. P., Rosa, H., Brogueira, G., & Batista, F. (2017). MISNIS: An intelligent platform for Twitter topic mining. Expert Systems with Applications, 89, 374–388.

Crovato, C. D. P., et al. (2021). Fast IoT: An efficient and very fast compression model for displaying a huge volume of IoT data in web environments. International Journal of Grid and Utility Computing, 12(5–6), 605–617.

De Almeida Pereira, G. H., Fusioka, A. M., Nassu, B. T., & Minetto, R. (2021). Active fire detection in Landsat-8 imagery: A large-scale dataset and a deep-learning study. ISPRS Journal of Photogrammetry and Remote Sensing, 178, 171–186.

Erraji, A., Maizate, A., Ouzzif, M., & Batouta, Z. I. (2022). Migrating data semantic from relational database system to NoSQL systems to improve data quality for big data analytics system. ECS Transactions, 107(1), 19495.

Feng, X., et al. (2021). Transparent ciphertext retrieval system supporting the integration of encrypted heterogeneous database in cloud-assisted IoT. IEEE Internet of Things Journal, 9(5), 3784–3798.

Huang, S., Qin, Y., Zhang, X., Tu, Y., Li, Z., & Cui, B. (2023). Survey on performance optimization for database systems. Science China Information Sciences, 66(2), 121102.

Kaur, C., & Sharma, A. (2020). Twitter sentiment analysis on coronavirus using TextBlob. EasyChair Preprints, 2516–2314.

Kausar, M. A., & Nasar, M. (2018). An effective technique for detection and prevention of SQLIA by utilizing CHECKSUM based string matching. International Journal of Scientific and Engineering Research, 9(1), 1177–1182.

Kausar, M. A., & Nasar, M. (2019a). Suitability of InfluxDB database for IoT applications. International Journal of Innovative Technology and Exploring Engineering, 8(10), 1850–1857.

Kausar, M. A., & Nasar, M. (2022). A study of performance and comparison of NoSQL databases: MongoDB, Cassandra, and Redis using YCSB. Indian Journal of Science and Technology, 15(31), 1532–1540.

Kausar, M. A., Nasar, M., & Singh, A. (2024a). Sentiment classification based on machine learning approaches in Flipkart product reviews. In Artificial Intelligence and Information Technologies (pp. 400–406). CRC Press.

Kausar, M. A., Nasar, M., & Singh, A. (2024b). Web accessibility challenges in Delhi NCR region university websites: An in-depth analysis using machine learning. In Artificial Intelligence and Information Technologies (pp. 393–399). CRC Press.

Korherr, P., & Kanbach, D. (2023). Human-related capabilities in big data analytics: A taxonomy of human factors impacting firm performance. Review of Managerial Science, 17(6), 1943–1970.

Kumar, A. (2017). NoSQL for handling big and complex biological data. In NoSQL: Database for Storage and Retrieval of Data in Cloud (pp. 143–158). Chapman and Hall/CRC.

Liang, W., Li, W., & Feng, L. (2021). Information security monitoring and management method based on big data in the Internet of Things environment. IEEE Access, 9, 39798–39812.

Meera, C., & Sundar, C. (2021). A hybrid metaheuristic approach for efficient feature selection methods in big data. Journal of Ambient Intelligence and Humanized Computing, 12, 3743–3751.

Mershad, K., & Hamieh, A. (2021). SDMS: Smart database management system for accessing heterogeneous databases. International Journal of Intelligent Information and Database Systems, 14(2), 115–152.

Mohammad, A. S., & Pradhan, M. R. (2021). Machine learning with big data analytics for cloud security. Computers and Electrical Engineering, 96, 107527.

Muniswamaiah, M., Agerwala, T., & Tappert, C. C. (2023). IoT-based big data storage systems challenges. In 2023 IEEE International Conference on Big Data (BigData) (pp. 6233–6235).

Nasar, M. (2023). Web 3.0: A review and its future. International Journal of Computer Applications, 185(10), 41–46.

Nasar, M., & Al Musalhi, N. (2025). Forecasting week-ahead closing price of Muscat Securities Market using hybrid TCN-LSTM model. Journal of Theoretical and Applied Information Technology, 103(7), 2980–2990.

Ra, M., Ab, B., & Kc, S. (2020). COVID-19 outbreak: Tweet-based analysis and visualization towards the influence of coronavirus in the world.

Samuel, J., Ali, G. G., Rahman, M. M., Esawi, E., & Samuel, Y. (2020). COVID-19 public sentiment insights and machine learning for tweets classification. Information, 11(6), 314.

Tang, E., & Fan, Y. (2016). Performance comparison between five NoSQL databases. In 2016 7th International Conference on Cloud Computing and Big Data (CCBD), Macau, China.

Truică, C. O., Apostol, E. S., Darmont, J., & Pedersen, T. B. (2021). The forgotten document-oriented database management systems: An overview and benchmark of native XML DODBMSes in comparison with JSON DODBMSes. Big Data Research, 25, 100205.

Zhang, H., et al. (2022). Big data-assisted social media analytics for business model for business decision-making system competitive analysis. Information Processing & Management, 59(1), 102762.