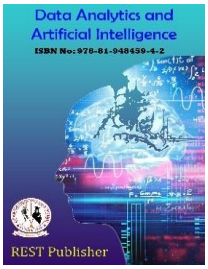
Data Analytics and Artificial Intelligence

Vol: 5(2), 2025

REST Publisher; ISBN: 978-81-948459-4-2

Website: http://restpublisher.com/book-series/daai/

DOI: https://doi.org/10.46632/daai/5/2/9



Performance Improvement of Data Vault Processes Using the Random Forest Regression Method: An Empirical Analysis of ETL Load Parameters

***Sapthagiri Padmanabham**

Vice President, JP Morgan Chase

*Corresponding author email: Sapthagiripadmanabham@gmail.com

Abstract: *This study investigates the operational performance and optimization challenges of data warehouse implementations in modern data warehousing environments. As organizations face increasing data volumes and complexity, Data Vault 2.0 has emerged as an adaptive architectural approach addressing the limitations of traditional data warehousing methods. Through the analysis of 350 operational observations, this research explores the relationships between source system load time, the number of records loaded, transformation complexity score, and load performance index. The study employs random forest regression to model these relationships and predict performance outcomes. Descriptive statistics reveal significant variability in operational parameters, with source system load times ranging from 11 to 96 seconds and transformation complexity scores varying from near zero to 123.64. Correlation analysis demonstrates a strong positive relationship (r = 0.84) between load time and performance index, while the number of records shows only a moderate impact on overall performance. The random forest model achieves high accuracy with R^2 values of 0.9832 for training data and 0.8200 for test data, confirming its effectiveness in capturing non-linear performance patterns. The results indicate that transformation complexity and system load time, rather than just data volume, are the primary drivers of Data Vault performance. These findings provide practical insights for organizations implementing data vault architectures, emphasizing the importance of optimizing transformation processes and managing system load times to achieve efficient data warehousing operations.*

Key words: *Data Vault, Data Warehouse, ETL Performance, Random Forest Regression, Load Performance Optimization.*

1. INTRODUCTION

A data warehouse serves as a repository of information designed to facilitate decision-making at the management level. It can serve as a tool for maintaining long-term records and for improving information governance systems to ensure regulatory compliance and better data oversight. In contrast to enterprise data warehouses (EDWs), data marts represent more focused, specialized versions that contain only a portion of the data in a broader warehouse. [2] There is a common misconception that data warehouses remain unchanged after they are built. However, this is not accurate, as organizational needs evolve and expand over time, and operational practices undergo regular changes. These changes create the need for new types of information requests that require different types of data. [3] As reliance on data increases, its volume, velocity, and diversity have increased significantly in recent decades. Thus, adaptations in data modeling approaches are required to accommodate these changes. For example, given the sheer volume and diversity of data, relying solely on structured formats is becoming increasingly challenging. This is prompting organizations to embrace semi-structured and unstructured data formats as well. [4] Conventional methods typically provide limited capabilities for processing non-standard data types within databases. Data Vault is built within the framework of Monet DB and uses its scientific array-query language, Sci QL. [5] Recently, however, a third technique called “data vault” has emerged and is experiencing rapid proliferation as it improves the adaptability, growth potential, and operational efficiency of data warehouses. [6] Since the fuzzy vault scheme only retains the transformed

Copyright@ REST Publisher

shape of the template, the primary implementation challenge lies in aligning (registering) the query fingerprint with the original template. In our approach, we store the high-curvature points extracted from the orientation field of the template as auxiliary data to facilitate the alignment process. [7] Block chain has fundamental properties that ensure data integrity and prevent unauthorized changes. Interestingly, data vaults adopt comparable principles. However, implementing data storage through block chain is relatively complex, and understanding the overall process is challenging. [8] Through the use of Data Vault 2.0, the study aims to demonstrate its effectiveness in addressing the unique challenges faced by higher education institutions - such as data inconsistency between systems, the need for real-time data updates, and the need to enable analytics-based decision-making. [9] Real-time analytics and large-scale data integration, its architectural design, and automated features emphasize how organizations can gain immediate insights, scalable functionality, and reliable data oversight. By leveraging both architectural best practices and self-managed systems, businesses can maximize the benefits derived from their data resources while maintaining adaptability in ever-changing operational environments [10]. This study aims to explore the challenges of modeling large and complex structures and combining data modeling with different real-time engineering techniques. The primary focus of these research papers is to provide an overview of data vaulting or to evaluate it in comparison to traditional data warehousing methods. This research examines in detail the challenges of modeling complex data vaulting projects. In particular, it examines the difficulties of managing the large number of tables to be modeled, dynamic changes to existing models, unique modifications required for data vaulting, and the potential for human errors in manual model creation.

2. MATERIALS AND METHOD

Materials

Source System Load Time: This variable represents the total time, in seconds, required to extract data from the source system and load it into the staging area. It reflects the performance of the data ingestion pipeline and can be affected by network latency, the size of the source system, and parallelization. Long load times may indicate high system utilization, complex data extraction, or performance bottlenecks within the data storage loading process.

Number Records Loaded: This feature measures the total number of data records successfully loaded during each ETL batch. It captures the amount of data being processed and helps assess performance and scalability. A high number of records typically indicates large batch loads, which require optimal indexing, parallel data ingestion, and robust resource management. Monitoring this variable ensures balanced system performance, identifying whether the data storage infrastructure can efficiently handle growing data volumes.

Transform Complexity Score: This variable measures the complexity of the data transformations performed before loading into the data storage. This score reflects the number and complexity of joins, calculations, lookups, and validation steps within the ETL process. Higher scores indicate more sophisticated transformation logic, which can increase processing time and computational requirements. Analyzing this metric helps identify optimization opportunities in the ETL design and ensures manageable transformation workloads for system stability.

Load Performance Index: This output variable represents a composite metric that evaluates the overall performance and efficiency of the Data Vault loading process. It combines factors such as system load time, data size, and transformation complexity to provide a holistic performance measure. A higher index indicates a more efficient and optimized load process, while lower values indicate bottlenecks or inefficiencies. This is essential for monitoring and improving Data Vault operational performance.

Optimization Techniques

Random Forest Regression: Is a powerful supervised machine learning method used for predictive modelling. This method involves training several decision trees on various dataset subsets and their outputs are averaged to improve the prediction accuracy of the method, not only improving performance but also reducing the computational burden associated with training, storing, and predicting with many individual models. Due to their efficiency, random forests are extremely helpful for jobs involving regression, where continuous values are usually predicted. A "forest" of several independently built decision trees is created using the random forest technique, using the ultimate forecast derived by averaging each tree's outputs. By exposing each tree to slightly different data, this approach helps to reduce variance and increase the over fitting, ultimately improving the generalizability of the model.

Copyright@ REST Publisher

3. ANALYSIS AND DISCUSSION

The dataset represents the operational performance of the Data Vault implementation processes, focusing on how system load time, data size, and transformation complexity affect the overall load performance index. Source system load time (approximately 11 to 96 seconds) represents the time it takes to transfer source system data to the stage or source tier. Longer load times are often associated with higher performance index values, such as 96.23 seconds, which corresponds to a load performance index of 219.32, which illustrates a nonlinear relationship where processing overhead increases proportionally for larger and more complex loads. number records loaded varies widely - from a few thousand to nearly 100,000 - reflecting varying data block sizes across different runs. Higher record counts do not always equate to higher performance indices, indicating that load performance is more affected by complexity and system time than sheer data size. For example, even with over 90,000 records, some runs show modest performance indices below 100, indicating well-optimized load processes. The transformation complexity score, which indicates the amount of data transformation, ranges from near zero (minimum processing) to 120 (very complex transformations). As transformation complexity increases especially above 40 the performance index rises sharply, reflecting additional computation, dependency resolution, and resource contention. The load performance index (3 to over 220) captures the combined impact of these factors. The results emphasize that while data volume affects performance, transformation complexity and system performance at load time dominate in determining the performance effects of data presence load.

TABLE 1. Descriptive Statistics

	Source System Load time	Number records loaded	Transform Complexity score	Load Performance index
Count	350	350	350	350
Mean	49.998097	50381.42871	20.083256	107.048463
Std	11.472996	28754.22691	20.348248	35.126389
Min	11.104792	2124.010832	0.092856	3.667304
25%	41.922343	25000.0676	5.259115	85.449419
50%	50.622683	50803.81422	13.676573	105.598141
75%	57.503366	75061.61417	28.227733	129.040491
Max	96.232778	99972.04966	123.643888	219.326609

Table 1 provides descriptive statistics for the dataset used in analyzing the performance of the Data Vault implementation, based on 350 observations. The parameters include the source system load time, the number of records loaded, the transformation complexity score, and the load performance index. The results highlight considerable variability across the dataset, indicating different operational conditions and performance effects. The source system load time averages 50 seconds, with values ranging from 11.10 to 96.23 seconds. This suggests that while most data loading processes complete within a minute, some outliers experience significantly longer times, perhaps due to network latency or large data sizes. The number of records loaded shows a wide range, from over 2,000 records to almost 100,000 records, with an average of 50,381. This spread reflects the varying batch sizes processed by different systems or at different times. The transformation complexity score, which measures the amount of data processing or transformation required, has a mean of 20.08 and a relatively high standard deviation (20.35), indicating that transformation requests fluctuate greatly between runs. Complexity values range from almost zero to 123.64, indicating that some operations are minimal and others are significantly more complex. The load performance index, a composite measure of overall system performance, has a mean of 107.05 with values between 3.67 and 219.33. This variability indicates that system performance is highly sensitive to input conditions particularly transformation complexity and load time reinforcing the need for adaptive optimization in data vault architectures.

Copyright@ REST Publisher

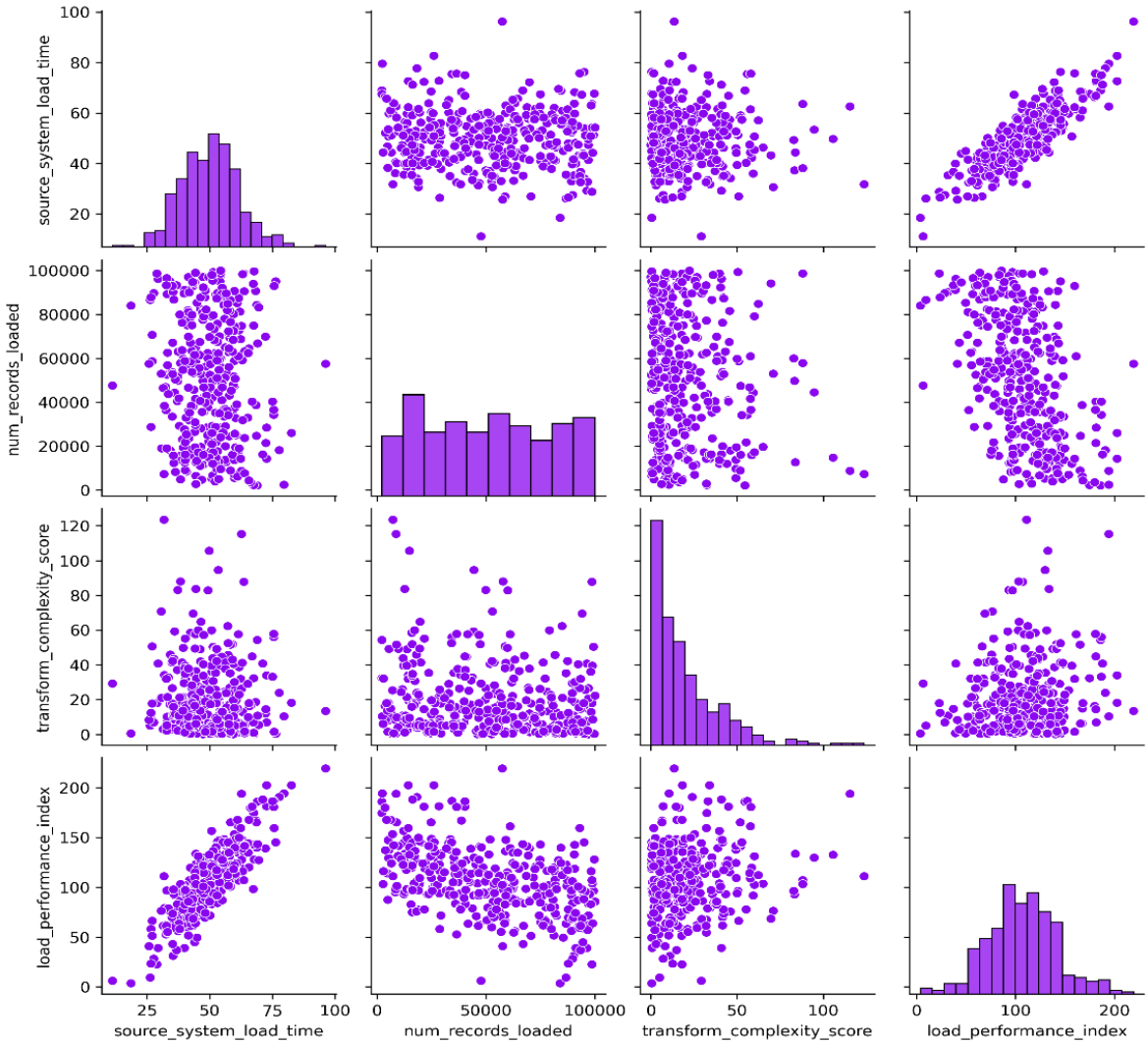


FIGURE 1. Scatter diagram showing the relationships between Data Vault implementation parameters

Figure 1 presents a scatter plot that illustrates the pairwise relationships between four key parameters that affect Data Vault implementation: source system load time, number of records loaded, transformation complexity score, and load performance index. The diagonal plots show the distributions of each variable, while the off-diagonal scatter plots reveal their correlations. The distribution of source system load time appears approximately normal, centered at 50 seconds, which indicates a stable data load duration for most operations. In contrast, the number of records loaded shows a nearly uniform distribution, indicating varying data block sizes with no dominant range. The transformation complexity score shows a right-skewed distribution, indicating that while most data transformations are relatively simple, a few involve high computational complexity. A strong positive relationship is evident between source system load time and load performance index, showing that longer load times tend to increase performance index values—perhaps due to more intensive data manipulation or complex transformations. Meanwhile, the transformation complexity score also demonstrates a moderate positive correlation with the performance index, indicating that higher complexity contributes to increased load performance overhead. However, the number of loaded records shows a weak correlation with the performance index, indicating that data size alone is not the dominant performance driver.

Copyright@ REST Publisher

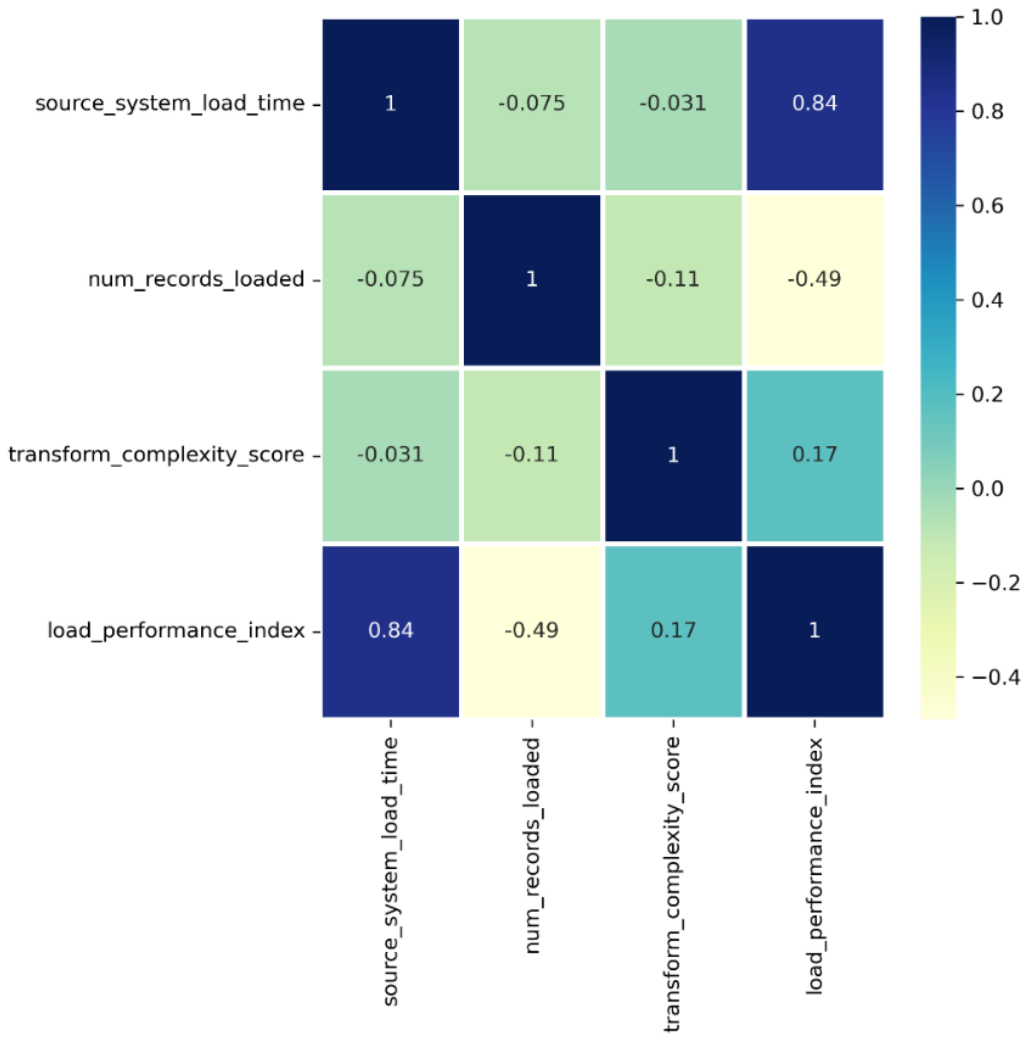


FIGURE 2. Heat Map of the Relationship Between Process Parameters and Responses

Figure 2 illustrates a correlation heat map that depicts the strength and direction of relationships between key Data Vault implementation parameters: source system load time, number of records loaded, transformation complexity score, and load performance index. The color scale ranges from dark blue (strong positive correlation) to light yellow (weak or negative correlation), enabling a visual understanding of the interdependencies between the variables. There is a significant strong positive correlation (r = 0.84) between source system load time and load performance index, indicating that longer data load times are directly related to higher performance index values. This suggests that system performance increases when more time is spent processing or validating data—which may reflect higher computational effort or higher transformation overhead. Conversely, the number of records loaded shows a modest negative correlation with the load performance index (r = -0.49), indicating that as data size increases, system performance may decrease slightly. This trend may be due to increased data transfer times or processing bottlenecks when dealing with large datasets. The Transform Complexity Score exhibits a weak positive correlation (r = 0.17) with the performance index, indicating that complexity contributes moderately to performance variations but is not the dominant factor.

Copyright@ REST Publisher

Random Forest Regression



FIGURE 3. Random Forest Regression (Training data)

Figure 3 (Training Data) shows a strong agreement between the actual and predicted load performance index values using the Random Forest Regression model. Most data points are very close to the 45° reference line, indicating high prediction accuracy and minimal error during training. The tight clustering suggests that the model effectively learned the underlying non-linear relationships in the training data, with low bias and variance in the training set.

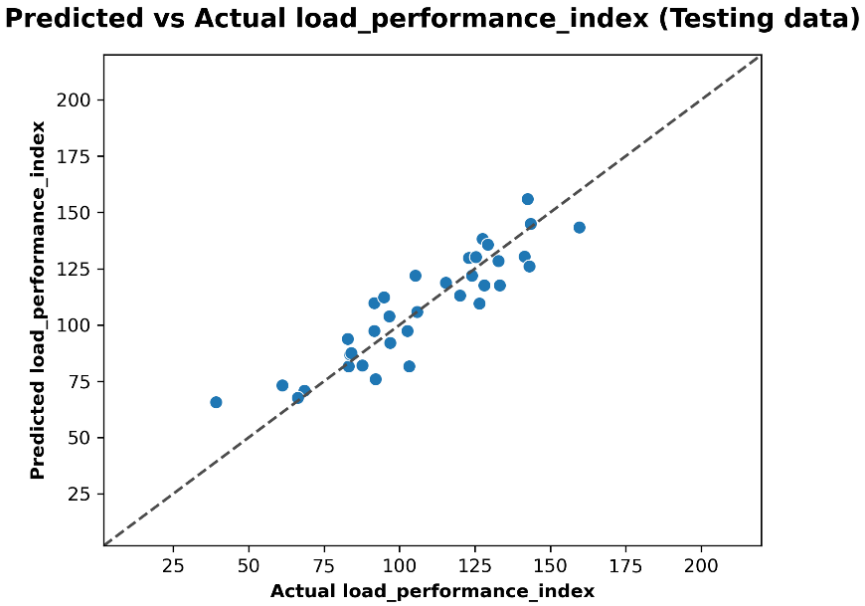


FIGURE 4. Random Forest Regression (Testing data)

Figure 6 (Test Data) demonstrates that the model maintains good predictive performance on unseen data. Although the scatter is slightly wider than in the training set, most points still follow the diagonal trend, indicating good generalization ability and limited overfitting.

Copyright@ REST Publisher

TABLE 2. Performance Metrics of Random Forest Regression (Training, Testing Data)

Data	R2	EVS	MSE	RMSE	MAE	MaxError	MSLE	MedAE
Train	0.9832	0.9832	21.5715	4.6445	3.6034	21.2618	0.0090	2.7591
Test	0.8200	0.8204	131.5308	11.4687	9.3740	26.5449	0.0171	6.9332

Table 2 presents the performance metrics of the Random Forest Regression model for both the training and testing datasets. The training results show excellent model fit, with a high $R2$ and EVS value of 0.9832, along with low error metrics (MSE = 21.57, RMSE = 4.64, MAE = 3.60), indicating strong learning capability. In the testing phase, the model achieves a satisfactory $R2$ value of 0.82, confirming good generalization to unseen data. Although the error values increased in the test set, the relatively low MSLE and MedAE values suggest stable predictive behavior without severe overfitting.

4. CONCLUSION

This research provides detailed insights into the performance characteristics and optimization opportunities within Data Vault implementations. Analysis of 350 operational observations reveals important relationships between system parameters and workload performance, demonstrating that Data Vault performance is primarily dependent on transformation complexity and source system load time, rather than data size alone. The strong positive correlation between load time and performance index (r = 0.84) highlights the significant impact of data processing time on overall system performance, while the modest negative correlation with record count indicates that increasing data size does not disproportionately reduce performance when properly optimized. The use of random forest regression proved to be very effective in modeling these complex, nonlinear relationships, achieving an R^2 of 0.9832 on the training data and maintaining strong generalization with an R^2 of 0.8200 on the test data. These results confirm the model's ability to accurately predict workload performance index values and identify potential bottlenecks before they impact production systems. The relatively low error metrics (RMSE = 4.64 for training, 11.47 for testing) confirm the model's robustness for practical use in monitoring and optimizing data vault operations. These findings highlight several important implications for practitioners implementing data vault architectures. First, organizations should prioritize improving transformation logic and complexity management over concerns about data volume scalability, as transformation complexity has a greater impact on performance outcomes. Second, the nonlinear relationship between load time and performance indicates that incremental improvements in data ingestion and stabilization processes can yield disproportionate benefits to overall system performance. Third, the successful application of machine learning techniques opens avenues for predictive performance monitoring and proactive resource allocation in data warehouse environments. Future research should expand this analysis to incorporate additional factors such as network latency, parallel processing configurations, hardware specifications, and real-time data integration scenarios. Investigation of deep learning approaches and ensemble methods beyond random forests could further improve predictive accuracy. Additionally, longitudinal studies examining performance trends over time will provide valuable insights into the long-term sustainability and scalability of data vault implementations as organizational data needs continue to evolve.

REFERENCES

[1]. Krneta, Dragoljub, Vladan Jovanović, and Zoran Marjanović. "A direct approach to physical Data Vault design." Computer Science and Information Systems 11, no. 2 (2014): 569-599.
[2]. Naamane, Zaineb, and Vladan Jovanovic. "Effectiveness of data vault compared to dimensional data marts on overall performance of a data warehouse system." International Journal of Computer Science Issues (IJCSI) 13, no. 4 (2016): 16.
[3]. Vines, Andreea, and Laura Tanasescu. "Data Vault Modeling: Insights from Industry Interviews." In Proceedings of the International Conference on Business Excellence, vol. 18, no. 1, pp. 3597-3605. Sciendo, 2024.
[4]. Ivanova, Milena, Yağiz Kargin, Martin Kersten, Stefan Manegold, Ying Zhang, Mihai Datcu, and Daniela Espinoza Molina. "Data vaults: a database welcome to scientific file repositories." In Proceedings of the 25th International Conference on Scientific and Statistical Database Management, pp. 1-4. 2013.
[5]. Yessad, Lamia, and Aissa Labiod. "Comparative study of data warehouses modeling approaches: Inmon, Kimball and Data Vault." In 2016 International Conference on System Reliability and Science (ICSRS), pp. 95-99. IEEE, 2016.
[6]. Ivanova, Milena, Martin Kersten, Stefan Manegold, and Yagiz Kargin. "Data vaults: Database technology for scientific file repositories." Computing in Science & Engineering 15, no. 03 (2013): 32-42.

Copyright@ REST Publisher

[7]. Sarwar, Muhammad Imran, Muhammad Waseem Iqbal, Tahir Alyas, Abdallah Namoun, Ahmed Alrehaili, Ali Tufail, and Nadia Tabassum. "Data vaults for blockchain-empowered accounting information systems." Ieee Access 9 (2021): 117306-117324.
[8]. Triaji, Bagas, Aloysius Agus Subagyo, and Muhammad Arif Rifai. "Development of a Higher Education Data Warehouse Using the Data Vault 2.0 Method." Sinkron: jurnal dan penelitian teknik informatika 8, no. 4 (2024): 2591-2602.
[9]. Gribova, Svetlana, and F. H. Wedel. "Literature review on data vaults—what is the state of the art of literature on data vaults." (2022).
[10]. Gallego, Daniel. "The Role of Data Vault 2.0 in Supporting Real-Time Analytics and Big Data Integration."
[11]. Subotic, Danijela, Vladan Jovanovic, and Patrizia Poscic. "Data Warehouse and Master Data Management Evolution-A Meta-Data-Vault Approach." Issues in information systems 15, no. 2 (2014).
[12]. Vines, Andreea, Ana-Ramona Bologa, and Andreea-Izabela Bostan. "AI-Powered Data Vault 2.0 Modeling for Business Intelligence and Automation." (2025).
[13]. Ließ, Mareike, Bruno Glaser, and Bernd Huwe. "Uncertainty in the spatial prediction of soil texture: comparison of regression tree and Random Forest models." Geoderma 170 (2012): 70-79.
[14]. Zhang, Wengang, Chongzhi Wu, Yongqin Li, Lin Wang, and P. Samui. "Assessment of pile drivability using random forest regression and multivariate adaptive regression splines." Georisk: Assessment and Management of Risk for Engineered Systems and Geohazards 15, no. 1 (2021): 27-40.
[15]. Jog, Amod, Aaron Carass, Snehashis Roy, Dzung L. Pham, and Jerry L. Prince. "Random forest regression for magnetic resonance image synthesis." Medical image analysis 35 (2017): 475-488.
[16]. Singh, Balraj, Parveen Sihag, and Karan Singh. "Modelling of impact of water quality on infiltration rate of soil by random forest regression." Modeling Earth Systems and Environment 3 (2017): 999-1004.
[17]. Cootes, Tim F., Mircea C. Ionita, Claudia Lindner, and Patrick Sauer. "Robust and accurate shape model fitting using random forest regression voting." In Computer Vision–ECCV 2012: 12th European Conference on Computer Vision, Florence, Italy, October 7-13, 2012, Proceedings, Part VII 12, pp. 278-291. Springer Berlin Heidelberg, 2012.

Copyright@ REST Publisher