Nanotechnology Perceptions
ISSN 1660-6795
www.nano-ntp.com

Machine Learning in Cloud Environments: Leveraging SQL and Python for Big Data Analytics

Bharath Muddarla[1], Praneeth Reddy Vatti[2]

[1]Senior TIBCO Engineer at Whiz IT Solutions
[2]Staff Software Engineer | System Intelligence and Machine Learning, Apple

The integration of SQL and Python in machine learning workflows within cloud environments offers a robust solution to the challenges of big data analytics. This study explores the synergistic capabilities of SQL for efficient data extraction and Python for flexible data transformation, model development, and visualization. Utilizing cloud platforms such as AWS and Google Cloud, the research established scalable pipelines for processing large datasets, training machine learning models, and deploying them in real-time. Statistical analyses validated the reliability and accuracy of models, with Neural Networks achieving the highest performance metrics. Resource utilization tests highlighted the critical role of GPUs in accelerating computations, while scalability tests demonstrated the adaptability of the pipeline to varying data loads. The results underscore the cost-effectiveness and efficiency of integrating SQL and Python in cloud-based machine learning, offering valuable insights for businesses and researchers aiming to optimize big data workflows. This study provides a practical foundation for leveraging cloud technologies to enhance analytics capabilities and addresses future directions for tool integration and emerging technologies.
Keywords: Big Data Analytics, Machine Learning, SQL, Python, Cloud Environments, Scalability, Data Processing, Resource Utilization.

1. Introduction

The Evolution of Cloud Computing and Big Data

The rapid expansion of digital infrastructure has ushered in an era where data is both an asset and a challenge for organizations across industries (Ionescu & Radu, 2024). Cloud computing has emerged as the cornerstone for managing, processing, and storing vast amounts of data. Paired with big data analytics, it enables organizations to gain actionable insights, streamline operations, and enhance decision-making processes (Sharma, 2016). With the advent of

advanced machine learning (ML) techniques, cloud environments are no longer mere repositories of data but have transformed into intelligent ecosystems capable of predictive and prescriptive analytics (Zhang et al. 2017). The integration of ML with cloud environments allows organizations to harness the full potential of big data, driving innovation and operational efficiency (Chillapalli1 and Murganoor, 2024).

The Role of SQL and Python in Machine Learning

Two programming tools that have become instrumental in leveraging machine learning in cloud environments are SQL and Python. SQL (Structured Query Language), a fundamental tool for managing and querying relational databases, ensures efficient data extraction and preparation—a crucial step for big data analytics (Kabanda, 2023). Python, known for its versatility and vast ecosystem of libraries, complements SQL by providing robust frameworks for developing and deploying machine learning models (Chillapalli, 2022). Together, these tools create a seamless pipeline for data ingestion, transformation, model building, and real-time deployment in cloud environments (Nibareke & Laassiri, 2020).

The compatibility of SQL and Python with cloud platforms such as AWS, Azure, and Google Cloud makes them indispensable for businesses navigating the complexities of big data analytics. Python's libraries, including TensorFlow, Scikit-learn, and Pandas, combined with SQL's data manipulation capabilities, enable organizations to unlock the power of machine learning to optimize business processes, predict trends, and automate decisions (Belcastro et al. 2017).

Big Data Challenges in Cloud-Based Machine Learning

Despite its promise, implementing machine learning in cloud environments comes with challenges (Jindal and Nanda, 2024). The sheer volume, velocity, and variety of data require sophisticated techniques to preprocess and manage datasets before applying ML algorithms. Additionally, organizations face hurdles related to data security, computational costs, and maintaining scalability and performance in a cloud infrastructure (Saeed & Ebrahim, 2024).

SQL and Python address some of these challenges by offering solutions for efficient data handling and processing (Armoogum & Li, 2019). While SQL excels in handling structured datasets, Python's adaptability allows for the integration of unstructured data sources, including logs, images, and text (More and Unnikrishnan, 2024). Together, these tools enable organizations to navigate the challenges posed by big data, ensuring that their ML models remain effective and scalable (Kanchepu, 2024).

Research Focus and Objectives

This research article explores the integration of SQL and Python in machine learning pipelines within cloud environments, with a specific focus on big data analytics. The study investigates how these tools can be leveraged to optimize data preprocessing, improve model accuracy, and facilitate real-time deployment. Furthermore, it delves into the role of cloud-based platforms in enhancing the scalability and performance of machine learning workflows.

By examining use cases and providing practical insights, the research aims to demonstrate the synergy between SQL, Python, and cloud technologies. The study also addresses the challenges and limitations of this integration, offering strategies for overcoming common

obstacles and maximizing the benefits of machine learning in cloud environments.

Importance of the Study

In an era driven by data, organizations that effectively harness machine learning within cloud ecosystems gain a competitive edge. The insights presented in this article will benefit researchers, data scientists, and business leaders seeking to leverage SQL and Python for big data analytics. By bridging the gap between theoretical understanding and practical application, this research contributes to the broader discourse on the role of technology in driving innovation and efficiency.

This introduction sets the stage for a deeper exploration of machine learning in cloud environments, highlighting the critical role of SQL and Python as enablers of big data analytics.

2. Methodology

Machine Learning in Cloud Environments

The methodology for this study revolves around the implementation of machine learning workflows within cloud environments to address challenges in big data analytics. Cloud platforms such as Amazon Web Services (AWS), Microsoft Azure, and Google Cloud were selected for their scalability, performance, and support for advanced machine learning services. These platforms were used to create virtual environments equipped with necessary computational resources, including GPU and TPU instances, for handling large datasets and executing complex machine learning algorithms.

The research focused on establishing a streamlined machine learning pipeline, including data ingestion, preprocessing, model training, validation, and deployment, within the cloud infrastructure. Various datasets representing real-world scenarios, such as sales trends, user behavior, and system logs, were used to test the robustness of the pipeline. Cloud-based tools like AWS SageMaker and Google AI Platform were utilized to evaluate the performance and scalability of the implemented models under varying data loads and computational demands.

Integration of SQL and Python for Big Data Analytics

To manage the volume, velocity, and variety of big data, SQL and Python were integrated to create an efficient data analytics workflow. SQL was employed for data extraction and querying from structured data sources, such as relational databases and cloud-hosted warehouses like Amazon Redshift, Google BigQuery, and Azure SQL Database. Specific SQL operations, including filtering, joining, and aggregating data, were tailored to reduce data redundancy and prepare datasets for machine learning tasks.

Python was incorporated as a complementary tool to handle data transformation, cleaning, and analysis. Libraries such as Pandas and NumPy were used for preprocessing, while Matplotlib and Seaborn were utilized for data visualization. Machine learning libraries, including Scikit-learn, TensorFlow, and PyTorch, were integrated to build, train, and evaluate models. The interoperability between SQL and Python allowed for seamless data movement, with Python scripts executing SQL queries to extract and preprocess data, followed by machine learning

workflows in Python.

The study also explored Python's capability to manage unstructured and semi-structured data, leveraging libraries like PySpark and Dask for distributed data processing. This integration enabled the research to address big data challenges, including handling large volumes of data across multiple formats, ensuring the pipeline's adaptability to real-world scenarios.

Statistical Analysis and Model Evaluation

Statistical analysis played a critical role in validating the performance of the machine learning models developed during the study. Key statistical methods were applied to evaluate data integrity, check for biases, and ensure model reliability. Descriptive statistics were used to summarize data characteristics, while inferential statistics helped identify patterns and relationships between variables.

Model evaluation metrics such as precision, recall, F1-score, and the area under the receiver operating characteristic (ROC) curve were employed to assess model accuracy and robustness. Cross-validation techniques were implemented to prevent overfitting, and hyperparameter tuning was conducted to optimize model performance. Cloud-native tools like AWS CloudWatch and Azure Monitor were used to log and analyze model performance metrics in real-time, ensuring their suitability for deployment in dynamic cloud environments.

Workflow Optimization and Scalability Testing

The final phase of the methodology involved testing the scalability of the machine learning pipeline under different data loads. Simulated datasets of varying sizes and complexities were used to evaluate the performance of SQL and Python workflows in cloud environments. Automated scaling features provided by the cloud platforms were analyzed to measure their impact on latency and cost-efficiency.

This methodology outlines a comprehensive approach to integrating machine learning, SQL, and Python in cloud environments, providing a framework for tackling big data challenges while ensuring model scalability and reliability.

3. Results

Table 1: Data Preprocessing Time Across Datasets with Statistical Analysis

Dataset Type	Data Size (GB)	SQL Query Time (sec)	Python Transformation Time (sec)	Total Preprocessing Time (sec)	Std. Dev. of Preprocessing Time (sec)	Success Rate (%)
Sales Data	50	12	18	30	2.5	98
User Behavior Logs	100	25	35	60	5.4	95
System Logs	200	50	80	130	10.2	93

The data preprocessing efficiency was evaluated by measuring the time required to extract, transform, and prepare datasets for machine learning. Table 1 highlights the preprocessing time across different datasets, showing that SQL performed exceptionally well in structured data extraction, while Python was effective in data transformation tasks. The overall preprocessing time remained consistent, with a standard deviation of 2.5–10.2 seconds, and success rates exceeded 93% for all datasets, indicating high reliability in data preparation

workflows.

Table 2: Model Training Performance with Statistical Analysis

Model	Training Time (sec)	Accuracy (%)	Computational Cost ($/hour)	Accuracy/Cost (%)	Training Time Variance (sec)
Random Forest	120	88.5	0.25	354.0	15
Neural Network	300	92.1	0.50	184.2	22
Gradient Boosting	180	90.3	0.40	225.8	18

The performance of different machine learning models, including Random Forest, Neural Networks, and Gradient Boosting, was assessed based on training time, accuracy, and computational cost. As shown in Table 2, Neural Networks achieved the highest accuracy (92.1%) but incurred higher computational costs, reflecting a trade-off between precision and resource consumption. The accuracy-to-cost ratio further revealed that Random Forest and Gradient Boosting models offered more cost-efficient solutions, particularly for mid-range predictive tasks.

Table 3: Statistical Analysis of Model Performance with Variability Metrics

Model	Precision (%)	Recall (%)	F1-Score (%)	ROC-AUC (%)	Std. Dev. of F1-Score (%)	Confidence Interval (95%)
Random Forest	87.0	86.5	86.7	89.0	1.2	[85.5, 87.9]
Neural Network	92.5	91.8	92.1	94.2	0.8	[91.5, 92.7]
Gradient Boosting	89.1	88.6	88.8	91.5	1.0	[87.8, 89.8]

To validate the effectiveness of the machine learning models, statistical metrics such as precision, recall, F1-score, and ROC-AUC were analyzed. Table 3 presents these results, indicating that the Neural Network consistently outperformed other models across all metrics, with an F1-score of 92.1% and a ROC-AUC of 94.2%. The standard deviation for these metrics was minimal (0.8–1.2%), and the 95% confidence intervals confirmed the robustness and reliability of these results.

Table 4: Scalability Metrics Under Varying Data Loads with Additional Parameters

Data Size (GB)	Latency (ms)	Cost ($/hour)	Throughput (GB/sec)	Cost-Efficiency ($/GB)	Latency Std. Dev. (ms)
50	150	0.20	0.33	0.004	5
100	280	0.35	0.36	0.0035	8
200	450	0.60	0.44	0.003	12

Scalability testing examined the pipeline's efficiency under varying data loads. As illustrated in Table 4, latency and cost increased with data size, but the system maintained a steady throughput and cost-efficiency. For instance, latency rose from 150 ms for 50 GB of data to 450 ms for 200 GB, but throughput improved from 0.33 GB/sec to 0.44 GB/sec, demonstrating effective scaling. The variance in latency remained within acceptable limits (5–12 ms), ensuring consistent performance.

Table 5: Resource Utilization Metrics with Variance Analysis

Resource	Utilization During Training (%)	Utilization During Inference (%)	Peak Utilization (%)	Variance in Utilization (%)
CPU	70	30	85	10
GPU	85	40	95	12
Memory	60	25	80	8

Resource utilization metrics provided insights into the efficiency of cloud infrastructure during model training and inference. Table 5 highlights the utilization of CPU, GPU, and memory, with GPU usage peaking at 95% during training. The variance in utilization was minimal, indicating that resources were efficiently allocated and managed across tasks. These results underscore the importance of leveraging GPUs for computationally intensive machine learning workflows.

Table 6: Comparative Analysis of SQL and Python with Performance Metrics

Task	SQL Execution Time (sec)	Python Execution Time (sec)	Execution Time Variance (sec)	Success Rate (%)	Best Tool
Data Extraction	15	50	2.0	99	SQL
Data Transformation	45	25	3.5	96	Python
Data Visualization	N/A	20	2.2	98	Python

A detailed comparison of SQL and Python for various data-handling tasks is presented in Table 6. SQL excelled in structured data extraction, achieving execution times as low as 15 seconds with a success rate of 99%. Python demonstrated superior flexibility in data transformation and visualization, with execution times of 25–50 seconds and success rates of 96–98%. The variance in execution times for both tools remained low, reflecting their reliability and suitability for big data analytics in cloud environments.

4. Discussion

The results of this study highlight the potential of integrating SQL and Python with machine learning workflows in cloud environments, providing insights into their complementary strengths and the challenges of scaling big data analytics. The discussion focuses on the key findings, their implications, and avenues for future research.

Data Preprocessing and Workflow Efficiency

The study underscores the synergy between SQL and Python in data preprocessing. SQL's strength in handling structured datasets efficiently, as evidenced by its low query execution times (Table 1), positions it as an indispensable tool for initial data extraction. Python, with its robust libraries and flexibility, complements SQL by managing data transformation and visualization tasks effectively. The high success rates (93–99%) and minimal standard deviation in preprocessing times indicate that this integrated approach ensures consistency and reliability, even with varying dataset sizes and complexities (Ed-daoudy & Maalmi, 2019).

The observed efficiency highlights the importance of leveraging SQL and Python together to optimize preprocessing workflows, especially in scenarios involving multi-format datasets. However, future studies could explore the inclusion of automated data cleaning tools and advanced schema-mapping techniques to further enhance preprocessing capabilities (Pop, et al. 2016).

Model Performance and Statistical Robustness

The machine learning models demonstrated strong performance, with Neural Networks achieving the highest accuracy and statistical robustness (Table 3). The minimal variance in F1-scores and confidence intervals confirms the reliability of these models, even under different data distributions (Kadapal and More, 2024). These results emphasize the value of

using cloud-based computational resources for handling complex predictive tasks (Pintye et al. 2021).

The trade-offs between accuracy and computational cost (Table 2) also reveal important considerations for model selection. While Neural Networks offered superior predictive capabilities, their higher resource requirements suggest that models like Random Forest or Gradient Boosting may be better suited for cost-sensitive applications (Assefi et al. 2017). This balance between accuracy and cost efficiency will be a critical factor for organizations adopting machine learning in resource-constrained environments (Weissman & van de Laar, 2020).

Scalability and Resource Utilization

The scalability tests (Table 4) demonstrated the resilience of the pipeline under increasing data loads, with throughput improving as data size increased (Fernández et al. 2014). The steady cost-efficiency metrics and manageable increases in latency highlight the adaptability of cloud-native machine learning workflows to big data challenges (Mittal et al. 2019). The findings also reinforce the importance of automated scaling features provided by cloud platforms, which enable dynamic resource allocation and cost optimization (Demirbaga et al. 2024).

Resource utilization metrics (Table 5) further validate the efficiency of cloud infrastructure, particularly the role of GPUs in accelerating model training. The high GPU utilization (85–95%) suggests that investing in GPU-optimized cloud instances can significantly enhance processing speed, particularly for deep learning tasks (Balakrishnan, 2024). Future work could investigate the integration of alternative accelerators, such as TPUs, to compare performance and cost benefits (Elshawi et al. 2018).

Comparative Strengths of SQL and Python

The comparative analysis (Table 6) illustrates the distinct advantages of SQL and Python in big data analytics. SQL's low execution times and high success rates for structured data tasks make it a reliable tool for data extraction (Pop, 2016). Python's versatility, demonstrated in its superior performance for data transformation and visualization, highlights its critical role in creating end-to-end machine learning workflows (Qolomany et al. 2019).

However, the findings also suggest areas for improvement. For instance, SQL's limitations in handling unstructured data could be mitigated by incorporating hybrid tools like SparkSQL (Ciaburro et al, 2018). Similarly, enhancing Python's native support for distributed data processing could further streamline workflows for extremely large datasets (Fowdur et al. 2018).

Implications and Future Directions

This study emphasizes the importance of selecting the right combination of tools and platforms to address the unique challenges of big data analytics (Kadapal et al. 2024). The results provide a roadmap for organizations looking to leverage cloud-based machine learning pipelines, offering practical insights into balancing performance, scalability, and cost (Olabanji, 2023).

Future research could expand on this framework by exploring advanced integration strategies, such as using machine learning orchestration tools like Apache Airflow or Prefect to automate

pipeline management (Jindal, 2024). Additionally, examining the impact of emerging cloud technologies, such as serverless computing and edge-based analytics, could provide further insights into optimizing big data workflows (Murganoor, 2024).

The integration of SQL and Python with machine learning in cloud environments offers a powerful solution for big data analytics. By combining SQL's efficiency in data handling with Python's flexibility in model development, organizations can build scalable and cost-effective workflows (Jain, 2024). The findings of this study pave the way for further innovations in cloud-based machine learning, ensuring its continued relevance in addressing the challenges of an increasingly data-driven world (Jain, 2023).

5. Conclusion

This study highlights the transformative potential of integrating SQL and Python with machine learning workflows in cloud environments to address the challenges of big data analytics. The findings demonstrate that SQL excels in managing structured data extraction with high efficiency, while Python's versatility and robust libraries enable seamless data transformation, model development, and visualization. By leveraging the strengths of both tools, organizations can build scalable, reliable, and cost-effective machine learning pipelines. The statistical robustness of the evaluated models, coupled with efficient resource utilization and scalable cloud infrastructure, underscores the feasibility of deploying such workflows in real-world applications. However, the study also identifies trade-offs between computational cost and accuracy, emphasizing the need for context-specific model selection. These insights provide a practical framework for organizations aiming to enhance their data analytics capabilities, while also setting the stage for future research to explore advanced orchestration tools, emerging technologies, and hybrid data processing approaches. Ultimately, this integration represents a significant step toward harnessing the full potential of big data in a cloud-enabled era.

References

1. Armoogum, S., & Li, X. (2019). Big data analytics and deep learning in bioinformatics with hadoop. In Deep learning and parallel computing environment for bioengineering systems (pp. 17-36). Academic Press.
2. Assefi, M., Behravesh, E., Liu, G., & Tafti, A. P. (2017, December). Big data machine learning using apache spark MLlib. In 2017 ieee international conference on big data (big data) (pp. 3492-3498). IEEE.
3. Balakrishnan, A. (2024). Data Virtualization Architecture Framework using Multi-Engine Data Platforms for Big Data Analytics and Machine Learning. International Journal of Computer Trends and Technology, 72(2), 82-91.
4. Belcastro, L., Marozzo, F., Talia, D., & Trunfio, P. (2017). Big data analysis on clouds. Handbook of big data technologies, 101-142.
5. Chillapalli, N.T.R. (2022). Software as a Service (SaaS) in E-Commerce: The Impact of Cloud Computing on Business Agility. Sarcouncil Journal of Engineering and Computer Sciences, 1.10: pp 7-18.
6. Chillapalli1, N.T.R and Murganoor, S. (2024). The Future of E-Commerce Integrating Cloud Computing with Advanced Software Systems for Seamless Customer Experience. Library

Progress International, 44(3): 22124-22135
7. Ciaburro, G., Ayyadevara, V. K., & Perrier, A. (2018). Hands-on machine learning on google cloud platform: Implementing smart and efficient analytics using cloud ml engine. Packt Publishing Ltd.
8. Demirbaga, Ü., Aujla, G. S., Jindal, A., & Kalyon, O. (2024). Cloud Computing for Big Data Analytics. In Big Data Analytics: Theory, Techniques, Platforms, and Applications (pp. 43-77). Cham: Springer Nature Switzerland.
9. Ed-daoudy, A., & Maalmi, K. (2019). A new Internet of Things architecture for real-time prediction of various diseases using machine learning on big data environment. Journal of Big Data, 6(1), 104.
10. Elshawi, R., Sakr, S., Talia, D., & Trunfio, P. (2018). Big data systems meet machine learning challenges: towards big data science as a service. Big data research, 14, 1-11.
11. Fernández, A., del Río, S., López, V., Bawakid, A., del Jesus, M. J., Benítez, J. M., & Herrera, F. (2014). Big Data with Cloud Computing: an insight on the computing environment, MapReduce, and programming frameworks. Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery, 4(5), 380-409.
12. Fowdur, T. P., Beeharry, Y., Hurbungs, V., Bassoo, V., & Ramnarain-Seetohul, V. (2018). Big data analytics with machine learning tools. Internet of things and big data analytics toward next-generation intelligence, 49-97.
13. Ionescu, S. A., & Radu, A. O. (2024, September). Assessment and Integration of Relational Databases, Big Data, and Cloud Computing in Financial Institutions: Performance Comparison. In 2024 International Conference on INnovations in Intelligent SysTems and Applications (INISTA) (pp. 1-7). IEEE.
14. Jain, S. (2024). Integrating Privacy by Design Enhancing Cyber Security Practices in Software Development. Sarcouncil Journal of Multidisciplinary, 4.11 (2024): pp 1-11
15. Jain, S. 2023). Privacy Vulnerabilities in Modern Software Development Cyber Security Solutions and Best Practices. Sarcouncil Journal of Engineering and Computer Sciences, 2.12 (: pp 1-9.
16. Jindal, G and Nanda, A. (2024): AI and Data Science in Financial Markets Predictive Modeling for Stock Price Forecasting. Library Progress International, 44(3), 22145-22152.
17. Jindal, G. (2024). The Impact of Financial Technology on Banking Efficiency A Machine Learning Perspective. Sarcouncil Journal of Entrepreneurship and Business Management, 3.11: pp 12-20
18. Kabanda, G. (2023). Performance of Machine Learning and Big Data Analytics Paradigms in Cyber Security. In AI, Machine Learning and Deep Learning (pp. 191-241). CRC Press.
19. Kadapal, R. and More, A. (2024). Data-Driven Product Management Harnessing AI and Analytics to Enhance Business Agility. Sarcouncil Journal of Public Administration and Management, 3.6: pp 1-10.
20. Kadapal, R., More, A. and Unnikrishnan, R. (2024): Leveraging AI-Driven Analytics in Product Management for Enhanced Business Decision-Making. Library Progress International, 44(3): 22136-22144
21. Kanchepu, N. (2024). Unleashing the Power of Cloud Computing for Data Science. In Practical Applications of Data Processing, Algorithms, and Modeling (pp. 222-233). IGI Global.
22. Mittal, M., Balas, V. E., Goyal, L. M., & Kumar, R. (Eds.). (2019). Big data processing using spark in cloud (p. 2019). Berlin, Germany: Springer.
23. More, A. and Unnikrishnan, R. (2024). AI-Powered Analytics in Product Marketing Optimizing Customer Experience and Market Segmentation. Sarcouncil Journal of Multidisciplinary, 4.11: pp 12-19
24. Murganoor, S. (2024) Cloud-Based Software Solutions for E-Commerce Improving Security and Performance in Online Retail. Sarcouncil Journal of Applied Sciences, 4.11 (2024): pp 1-9

25. Nibareke, T., & Laassiri, J. (2020). Using Big Data-machine learning models for diabetes prediction and flight delays analytics. Journal of Big Data, 7(1), 78.
26. Olabanji, S. O. (2023). Advancing cloud technology security: Leveraging high-level coding languages like Python and SQL for strengthening security systems and automating top control processes. Journal of Scientific Research and Reports, 29(9), 42-54.
27. Pintye, I., Kail, E., Kacsuk, P., & Lovas, R. (2021). Big data and machine learning framework for clouds and its usage for text classification. Concurrency and Computation: Practice and Experience, 33(19), e6164.
28. Pop, D. (2016). Machine learning and cloud computing: Survey of distributed and saas solutions. arXiv preprint arXiv:1603.08767.
29. Pop, D., Iuhasz, G., & Petcu, D. (2016). Distributed platforms and cloud services: Enabling machine learning for big data. Data Science and Big Data Computing: Frameworks and Methodologies, 139-159.
30. Qolomany, B., Al-Fuqaha, A., Gupta, A., Benhaddou, D., Alwajidi, S., Qadir, J., & Fong, A. C. (2019). Leveraging machine learning and big data for smart buildings: A comprehensive survey. IEEE access, 7, 90316-90356.
31. Saeed, A., & Ebrahim, H. (2024). The Intersection of Machine Learning, Artificial Intelligence, and Big Data. In Big Data Computing (pp. 111-131). CRC Press.
32. Sharma, S. (2016). Expanded cloud plumes hiding Big Data ecosystem. Future Generation Computer Systems, 59, 63-92.
33. Weissman, B., & van de Laar, E. (2020). SQL Server Big Data Clusters. Apress.
34. Zhang, Y., Ordonez, C., & Johnsson, L. (2017, August). A cloud system for machine learning exploiting a parallel array DBMS. In 2017 28th International Workshop on Database and Expert Systems Applications (DEXA) (pp. 22-26). IEEE.