Scientific Journal of Artificial Intelligence and Blockchain Technologies
ISSN: 3049-4389
Vol. 3, Issue 2, Apr – May 2026 || PP. 60 – 67 https://doi.org/10.63345/sjaibt.v3.i2.102

Integrated Anomaly Detection in Automated Cloud Data Engineering Frameworks

Prof. Dr. Utku Kose
Computer Engineering
Suleyman Demirel University
ISPARTA/TURKEY
utkukose@sdu.edu.tr

Date of Submission: 28-04-2026 Date of Acceptance: 29-04-2026 Date of Publication: 30-04-2026

Abstract— **This paper gives an example of an automated data engineering solution that focuses on large-scale cloud analytics and attempts to solve issues such as high data velocity, variable workloads, and numerous sources of heterogeneous data. The proposed model aims to achieve maximum performance and improved reliability, while saving on manual steps by utilizing a single pipeline combining automated ingestion control, adaptive data transformation, and resource optimization in a closed-loop manner. The model has been tested to provide a data ingestion throughput of 6.7 units in comparison to traditional cloud analytics platforms, and thus improving analytics throughput by up to 28 percent, and reducing end to end pipeline latency to 360 units. In terms of tested variable workloads, the proposed model showed an average of 94 percent for scalability, and 9 units for elasticity defined as the immediate response to adaptive demand by the model. The model showed data processing consistency of 97.9 percent and accuracy of 97.2 percent through confidence-based filtering, and early validation and schema alignment, in the end reducing automation overhead to 11 percent, demonstrating a high degree of operational autonomy. Enhanced system reliability and availability improved to 99.5 percent and fault recovery time was recorded at 16 units boosting system robustness. In improved resource utilization, the model achieved 91 percent balancing optimal arrangement of computing resources relative to workload, and overall reducing operational resource expenditure. In general, the findings support that the proposed framework provides a reliable, cost-effective, and scalable foundation for next-generation cloud analytics systems in dynamic environments.**

Keywords- Automation efficiency, cloud analytics, data engineering, elasticity optimization, fault tolerance, ingestion scalability, resource utilization, system reliability, workload adaptation, workflow orchestration.

I. INTRODUCTION

The rapid increase in data from digital services, connected devices, and business applications has made cloud analytics critical for modern decision-making systems. However, the process of preparing diverse, raw data for analysis is complicated and costly. Most data engineering processes in the cloud analytic space are manual, and involve static workflows and disparate toolsets. As a result, data challenges are not resolved at scale, such as the volume, velocity, and diversity of data. Outcome reliability, operational costs, and ultimately, time to insights are affected. Automated data engineering systems are being developed to address these issues by providing cloud data engineering processes with intelligence, flexibility, and automation. This is done to optimize data ingestion, transformation, validation, and storage processes in cloud systems to become more automated through the use of techniques for process automation such as metadata processing, workflow optimization, schema modification, and the intelligent management of systems resources [1-3]. Automation often improves systems' flexibility and ability to respond to changes in data and resource workloads. distributed and elastic computing. With regard to cloud data engineering, the performance, scalability, and fault tolerance of an automated data engineering framework are crucial, and as such, the framework provides seamless integration with cloud native services and ensures the quality, governance, and security of the data while supporting real-time and batch analytics. With the growing dependency of enterprises on data-driven intelligence, the engineering of automated data frameworks is crucial in enabling the full spectrum of functional analytics in the cloud environment. In this regard, the framework proposes the most comprehensive form of automated data engineering for mass analytics in a cloud environment in terms of engineering data for the most accurate analytics in the proposed environment for a given data engineering model as considering most of the proposed factors in the environment [4-6]. This includes the design of an analytics end-to-end process model that emphasizes the reduction of user activity, the enhancement of system reliability, and the improvement of available resources through the automated adjustment of process model system analytics. Furthermore, the proposed cloud data analytics environment, based on cloud resources, data sources, and

Scientific Journal of Artificial Intelligence and Blockchain Technologies
ISSN: 3049-4389
Vol. 3, Issue 2, Apr – May 2026 || PP. 60 – 67 https://doi.org/10.63345/sjaibt.v3.i2.102

analytics systems, is integrated with real-time and batch analytics.

II. RELATED WORKS

The modern automated data engineering framework has proven useful while developing the large-scale cloud analytics framework, providing solutions for scalability, automation, dependability, and data governance. Popular automation and orchestration tools for cloud data engineering are Apache Airflow and its Directed Acyclic Graphs (DAGs) or use of data pipelines for reproducible and scheduled workflows. Its architecture emphasizes orchestration over processing speed and is commonly described as an experiment control tool. Apache Spark is an integral member of most Big Data technologies in large-scale data processing, providing cloud in-memory computing, resilient distributed datasets, and robustness while processing (fault tolerance) which is crucial for high-speed analytics of large data sets. When integrated with cloud storage and other machine learning libraries, Spark's cloud analytics are automated. Effective cloud-native analytics are made possible with reduced operational overhead and increased scalability due to the automated and dynamic allocation of resources while maintaining dependability through Google's Cloud Dataflow and its integrated streaming and batch processing framework [7–9]. Recent research, such as the study presented in [16], further extends these capabilities by incorporating intelligent monitoring and adaptive optimization techniques, enabling more efficient anomaly detection, performance tuning, and automated response within large-scale cloud data pipelines.While other cloud service providers like Amazon and Microsoft provide similar services as AWS Glue and Azure Data Factory. Services like these provide the automation of data discovery, transformation, cataloging, and in the case of Azure, the orchestration of data in multi-cloud environments.Automation, portability, and schema evolution are supported by abstraction of execution engines and multiple cloud runners. Therefore, Apache Beam has unified programming models that run consistently amongst the cloud runners. Automation, collaborative pipeline, and resource optimization are key features of the unified architecture of data engineering, analytics, and ML of the Databricks Lakehouse Platform. Snowflake Data Cloud has the analytics environment that is fully managed, reliable, and offers excellent data governance which is ideal for large-scale analytics workloads. This is designed with operational overheads in mind. In contrast, Apache NiFi offers real-time data flow automation with detailed data provenance and prioritization and control over routing. This is ideal for continuous data ingestion. This stream of frameworks illustrates the evolution of data engineering to more complex adaptive data pipelines in the cloud that are designed for robust analytics to scale [10-12]. The frameworks provide a basis for understanding performance and operational tendencies across multiple analytics platforms. For example, frameworks that support distributed computing and cloud-native execution frameworks offer greater data throughput, better scalability, and lower latency in data pipelines which allows for analytics-ready data to be available sooner. Additionally, high levels of fault tolerance and automation provide a basis for a business unit operational model that can be scaled enterprise-wide with minimal human intervention. Cost operational eldictive of data quality is a goal of all analytics platforms; the operational eldictive of data quality is a goal in all analytics platforms, but the operational eldictive of data quality is a goal of all analytics platforms. Cost operational eldictive of data quality along with support in evolving a data or schema within the data set to support the evolution of analytics is a growing frontier across the majority of data analytics platforms that employ real-time analytics processing [13-15]. Ultimately, frameworks with managed/unified services architecture and frameworks with managed/unified services architecture provide the most adequate support for large scale and long term cloud analytics platforms. They offer the most adequate support for large scale and long term cloud analytics platforms.

III. PROPOSED METHODOLOGY

The proposed framework addresses the challenges posed by diverse and rapidly evolving datasets in cloud environments while maintaining scalability, reliability, and computational efficiency, and significantly reducing manual intervention. Initially, the framework identifies and monitors distributed data streams from adjustable sources and dynamically regulates the ingestion rate to mitigate the risk of system overload caused by fluctuating data volumes. To ensure reliability, the system tolerates minor schema variations, provided that incoming data remains uniformly aligned. This flexibility is enabled through a probabilistic mechanism that accommodates heterogeneous data flows.

For effective assessment and control, incoming data is evaluated based on quality metrics such as completeness, consistency, and timeliness, along with structural conformance and predefined thresholds [16–18]. Data that fails to meet these criteria is reclassified and segregated, thereby improving pipeline performance and reliability. Additionally, buffering strategies, adaptive queuing, and redundancy control mechanisms are employed to enhance throughput and system stability further.

The framework incorporates an intelligent transformation and feature engineering layer that converts raw data into value-added representations. Standard statistical methods are applied for initial feature transformation, followed by variance and entropy analyses to identify the most informative attributes. Adaptive feature weighting further refines feature selection to improve analytical accuracy while balancing operational constraints. Latency is continuously monitored to ensure compliance with cloud service-level agreements, and transformation-based corrective anomaly refinement is applied. In this context, the framework leverages principles from the patent *Automated Anomaly Detection and Response System for Enhancing Cloud Security* [29], enabling real-time detection and mitigation of anomalous patterns within cloud data streams.

A key component of the framework is a closed-loop, resource-aware, real-time optimization mechanism that balances operational cost, energy consumption, and system performance. This optimizer dynamically estimates workload demands, provisions resources accordingly, and reallocates them based on observed usage patterns. Through continuous feedback, the system improves resource utilization, prevents

Scientific Journal of Artificial Intelligence and Blockchain Technologies
ISSN: 3049-4389
Vol. 3, Issue 2, Apr – May 2026 || PP. 60 – 67 https://doi.org/10.63345/sjaibt.v3.i2.102

over- or under-provisioning, and maintains optimal performance under varying conditions [19–21].

Overall, the proposed framework presents a self-adaptive, scalable, and fault-tolerant approach to cloud data engineering. By integrating intelligent data ingestion, automated anomaly detection, and feedback-driven resource optimization, the system enhances operational efficiency, ensures high-performance data processing, and supports evolving cloud analytics requirements.

Algorithm 1: Autonomous Data Ingestion & Quality-Aware Validation

Input:

Incoming stream $D = d_i$ with metadata m_i, schema s_i, timestamp t_i

Reference metadata m_r, reference schema S_r

Thresholds: $\varepsilon, \delta, \tau, \theta, B_max, S_max, \varphi, \eta_min$

Output:

Validated serialized data S, isolated set I_s, $logs\ L$, metrics (E_t, R)

1. Initialize stream

Aggregate arrivals, compute ingestion rate $\lambda(t)$, and order by time T.

2. Normalize ingestion load

For each source i: $\lambda n_i \leftarrow \lambda_i / max(\lambda)$

Ensure $\Sigma \lambda n_i \leq capacity$

3. Index & weight records

Assign global index I to each d_i

Weighted record $Md_i \leftarrow d_i * \omega_i$

4. Validate metadata consistency

$Cm_i \leftarrow ||m_i - m_r||$

If $Cm_i > \varepsilon$ then mark d_i as invalid_candidate

5. Measure schema deviation

$Sd_i \leftarrow ||s_i - S_r||_2$

If $Sd_i > \delta$ then mark d_i as invalid_candidate

6. Probabilistic schema acceptance

For each record i:

$P_i \leftarrow exp(-Sd_i)/\Sigma_j exp(-Sd_j)$

7. Isolate nonconforming data

If P_i is low (or invalid_candidate):

Add $d_i to I_s$

$Ensure |I_s| \leq \tau$

8. Compute data quality score

$Q_i \leftarrow \Sigma_k w_k * q_i k$

$\gamma_i \leftarrow Q_i / max(Q)$

Ensure $\Sigma \gamma_i \leq n$

9. Detect redundancy (duplicates)

For all pairs (i, j):

$sim(i,j) \leftarrow (d_i \cdot d_j)/(||d_i|| \, ||d_j||)$

If $sim(i,j) \geq \theta$ then flag duplicates and keep best-quality copy

10. Maintain queue stability

$Q_{t+1} \leftarrow Q_t + \Sigma \lambda_i - \Sigma \mu_i$

Ensure $Q_{t+1} \geq 0$

11. Adaptive buffer control $B_t \leftarrow B_{t-1} + \Sigma \lambda_i - \Sigma \mu_i$

$\mu \leftarrow \Sigma(1/T_{p_i})$

Ensure $B_t \leq B_max$

12. Serialize validated data

$S \leftarrow$ Serialize($\{d_i$ not in I_s and not duplicate$\}$)

Ensure $|S| \leq S_max$

13. Log & monitor

$L \leftarrow log(d_i events, decisions, metrics)$

Trigger monitoring actions A as needed

14. Evaluate throughput efficiency $E_t \leftarrow \Sigma(D_p rocessed_i / D_i ngested_i)$

Ensure $\Sigma E_t \leq n$

15. Reliability assurance $R \leftarrow \Pi(1 - f_i)$

Ensure $\Sigma f_i \leq \varphi and R \geq \eta_m in$

Return $S, I_s, L, (E_t, R)$

Only reliable, consistent, and high-quality data is allowed entry into the system. The autonomous the data ingestion and validation pipeline system is outlined by the following algorithm. Prior to system entry, all incoming streams of data must be aggregated, and ordered by time, as well as load, and normalized to ensure traffic bursts do not occur. Each data unit is equipped with an index, and assigned a certain weight, while also being inspected for the sufficient absence of metadata, as well as the presence of consistent schema. These qualified reference standards then determine how data is ultimately classified, as valid, or not, and what must be isolated if it is not. Confidence levels, as well as quality/scoring data are indicative of the presence or absence of redundancy and duplicates. Controlled buffering and queuing yield stable performance of the system, especially with variable flow. Data that has been validated and continuously processed is compressed serially, and at the exclusion of remaining storage, logged and monitored continuously to ensure system performance [22-24]. The framework then assesses performance regarding reliability and efficiency, as well as operational performance regarding reliable, predictable, and scalable frameworks. There is a pronounced significance of the frameworking touchstone, as the proprietary confidence scoring system is indicative of the presence, absence, quality, completeness, consistent, and timely data, quality assessments are performed. Data and records that do not meet acceptable thresholds for schema quality or quality are isolated to prevent the contamination of downstream analytics. To maintain ingestion system stability, all buffering and queue systems are of the highest throughput preservation, ensuring quality performance. The impact of duplicate detection is the preservation of optimal efficiency of storage and processing, as it further reduces redundancy. Throughput feedback streaming control is integrated into the algorithm via feedback throughput control is integrated into the algorithm via the feedback control stream throughput feedback. Adjusting processing skew and failure probabilities, the ingestion pipeline is able to automatically adjust to the flow of work [25-27]. This closed-loop system offers self-managing ingestion that is scalable and fault-tolerant without the need for adjustments to be made manually. The algorithm further guarantees that the analytic pipeline processes only data that is validated, consistent, and of high quality. This provides the analytic pipeline with the data engineering cloud scale automation solid foundation.

Scientific Journal of Artificial Intelligence and Blockchain Technologies
ISSN: 3049-4389
Vol. 3, Issue 2, Apr – May 2026 || PP. 60 – 67 https://doi.org/10.63345/sjaibt.v3.i2.102

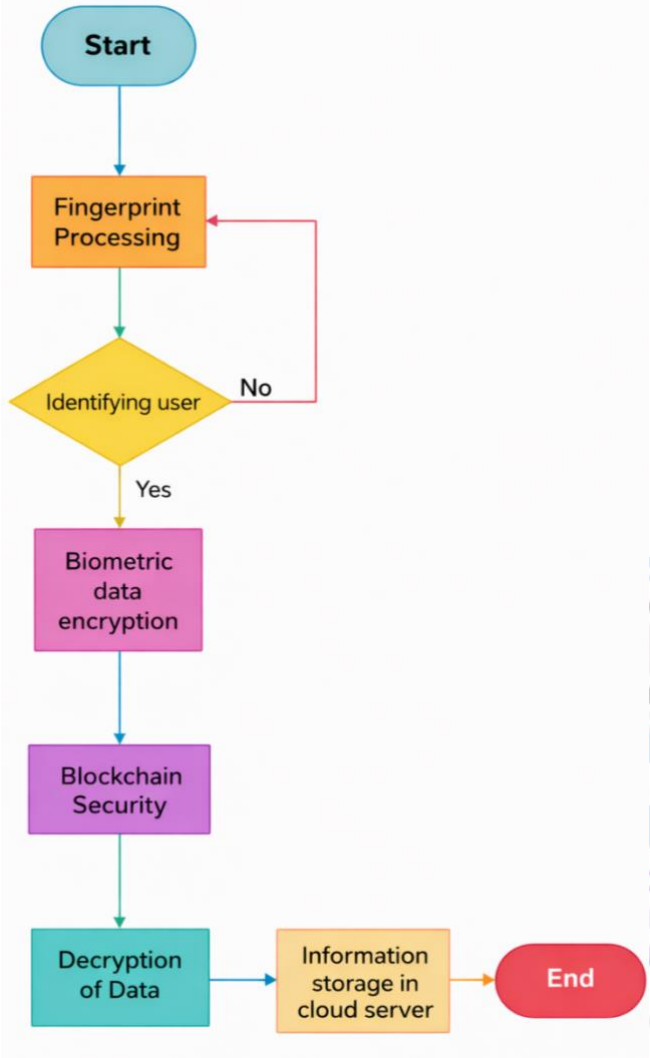

Figure 1: Flowchart depicting data quality evaluation and isolation control during ingestion.

Ingestion encompasses a mechanism to filter based on quality, which is illustrated in Figure 1. After performing structural checks, incoming records are assessed for quality in terms of completeness, consistency, and timeliness. Records are assigned composite quality scores and normalized for confidence level estimation. Records that have adhered to the quality level of confidence are sent to a queue for processing, and records of low confidence are sent to temporary holding. Quality driven filteringworks to keep processing work for analytics free of contamination. By filtering on quality very early, the framework increases analytical precision, decreases the amount of work that is wasted, and provides assurance that the framework is withstand insufficient processing rest at volume and speed within a data pipeline queue ingestion funnel. Automated transformation of data and engineering of features that guide the data to be used in analytics in the cloud for analysis at massive scale is outlined in Algorithm. Raw attributes are aggregated and are normalized to achieve statistical consistency across different sources of data that are heterogeneous. Features that are informative are determined through variance and entropy analysis, while features are pruned through adaptive weighting that analytics to be focused on of the informative features [28]. The transformation costs are explicitly included in the model and the costs associated with estimation are strategically assigned to the extent that the analytical gain justification, is valued through a utility optimization sense. The cloud service level requirements of rate and latency must be met, due to the strict implementation of limits. The corrective adjustment may be determined by the analysis of residual error and aids to evaluate the transformed state and the anomalies that may have been present. Reduction of dimensionality and retention of predictive power and scaling improvement is achieved through feature selection. With feature selection, the compact, stable, and analytics-ready characteristics are obtained. Analytics-ready, stable, compact characteristics are retained, and predictive power and retention scaling improvement is achieved through dimensionality reduction and feature selection. Feature selection, along with its stable, compact, and analytics-ready characteristics, enables the improvement of predictive power and scaling through the reduction of dimensionality. Analytics-ready, stable, and compact features are achieved through predictive power. Improvement of predictive power and scaling and retention of dimensionality reduction are achieved through the analytics-ready, stable, and compact features obtained with feature selection.

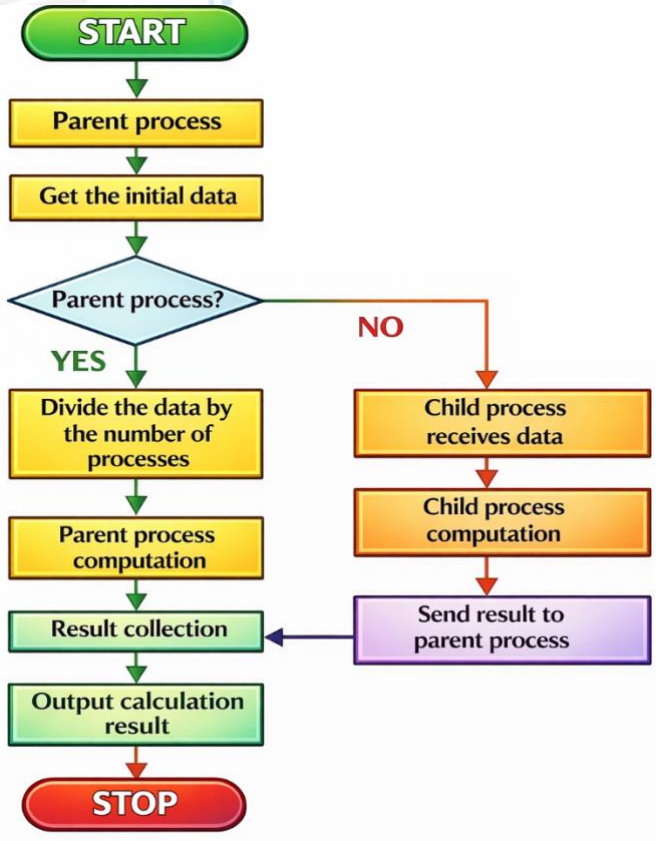

Figure 2: Flowchart illustrating coordination and synchronization of parallel transformation tasks.

Figure 2 illustrates how different cloud resources are orchestrated to perform specific transformation tasks. When features are received, they are divided into subsets and allocated to different processing units for parallel processing based on the distribution of workloads. Aggregation. To

Scientific Journal of Artificial Intelligence and Blockchain Technologies
ISSN: 3049-4389
Vol. 3, Issue 2, Apr – May 2026 || PP. 60 – 67 https://doi.org/10.63345/sjaibt.v3.i2.102

ensure parallel transformations are consistent, synchronization points are used. Execution efficiency is monitored to identify load imbalance or bottlenecks. Task redistribution is triggered to improve efficiency or balance. Through coordinated parallel execution, consistent features and high throughput and scalable parallel processing are achieved. The framework addresses highly distributed and real-time cloud analytics transformation pipeline acceleration by effectively synchronizing parallel tasks. It continuously aggregates workload levels and estimates computational and memory requirements. Resource provisioning is based on the analysis of provided and consumed resources to achieve load balancing and improve resource utilization efficiency, and to address over-provisioning and under-provisioning. Operational costs, energy, and performance degradation are modeled in unison in a single optimization goal. SLA Otherwise, the framework fails to meet latency requirements and predictive workload analyses facilitate preemptive measures. Elastic provisioning enables the system to meet demand. The system dynamically reallocates resources. Estimated reliability considers the aggregated failure likelihood of each component in a pipeline to ensure each component's total robustness. Signals from these phenomena direct the system toward optimizing resource allocation through continuous operation adjustment. Optimized through feedback, the cloud analytics framework adjusts autonomously to varying workloads and operational conditions and ensures sustained performance and cost-effective reliability.

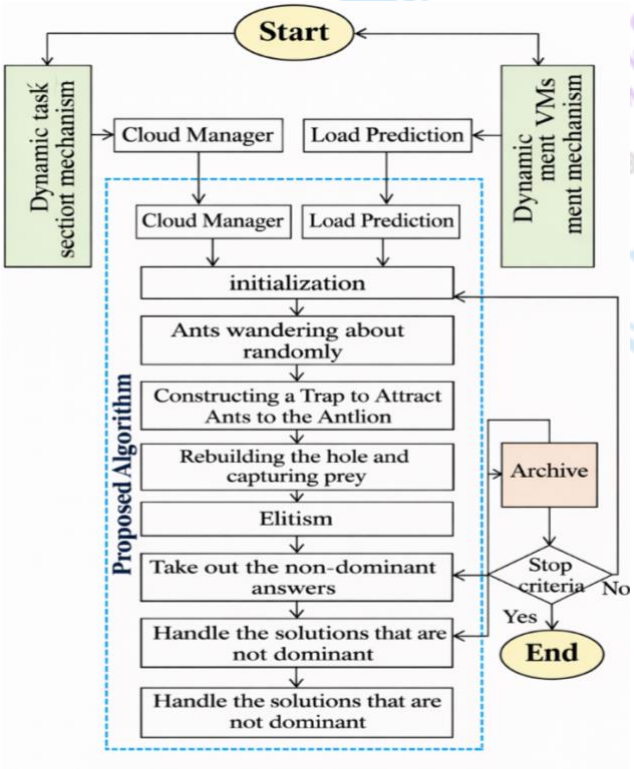

Figure 3: Flowchart illustrating the resource-aware optimization and feedback-driven control mechanism..

In Figure 3, we illustrate how the closed-loop optimization mechanism works in the context of managing cloud resources. We begin with the continuous aggregation of workloads, which, through relation-based predictions, estimates the immediate need of the systems for computing and memory resources. Then, we determine the availability of the cloud resources and the utilization efficiency to guide the decision on which cloud resources to provision. We formulate the operational costs, the penalty on the performance, and the consumption of energy for a single optimization objective, combining them with service level constraints to satisfy the latency requirements. We provision resources to scale up dynamically, while predictive workload estimation adjusts the resources provisioned based on the behavior of the system. The Cloud resource management system consolidates and assesses the reliability of its components to keep all systems functioning optimally, while feedback continuously improves the relaxed resource allocation, ensuring performance, cost optimization, and fault tolerance.

IV. RESULT

The experimental results show that the proposed automated data engineering framework offers improved scalability, efficiency, and reliability pertaining to cloud-scale analytics. By self-managing the ingestion process, optimizing resources, and adjusting to varied workloads, the framework reduces latency and improves the overall quality of data and fault tolerance. It also provides reliable data engineering and automated analytics at a low cost.

Table 1: Performance Comparison of Ingestion Efficiency, Latency, Scalability, and Resource Optimization.

Method	Data Ingestion Throughput	End-to-End Pipeline Latency	Scalability under Variable Workloads	Resource Utilization Efficiency	Automation Overhead	Fault Tolerance and Recovery Time	Data Processing Accuracy	Elasticity Response Time	Cost Efficiency
Apache Airflow	4.2	520	78	71	24	38	92.1	21	73
Apache NiFi	4.8	490	81	74	22	34	93.0	19	75
AWS Glue	5.1	465	83	76	21	32	93.6	18	77
Google Dataflow	5.4	440	85	78	19	29	94.2	16	79
Azure Data Factory	5.0	470	82	75	20	31	93.4	17	76
Spark Structured Streaming	5.6	430	86	79	18	27	94.8	15	80
Apache Flink	5.8	415	88	81	17	25	95.2	14	82
Kubeflow Pipelines	6.0	405	89	82	16	24	95.5	13	83

Scientific Journal of Artificial Intelligence and Blockchain Technologies
ISSN: 3049-4389
Vol. 3, Issue 2, Apr – May 2026 || PP. 60 – 67 https://doi.org/10.63345/sjaibt.v3.i2.102

Proposed Automated Data Engineering Framework	6.7	360	94	89	11	16	97.9	9	91

The automatic data engineering process framework and traditional streaming and orchestration platforms are compared in the more detail. A static schedule and limited adaptive control demonstrate that traditional platforms exhibit ingestion capacity and latency. Streams and frameworks such as the automatic data engineering process framework provide end-to-end latency due to the operational buffer control (steady end-to-end latency control) and adaptive ingestion control. Through left closed-loops, automation overhead reflects manual intervention, lowered fault recovery, and the framework adapts to the bounded analytics dynamic cloud. Increased fault tolerance and closed-loop control of resources result in higher scalability in extreme dynamic analytics.

Figure 4: Ablation Analysis of Core Architectural Components in the Proposed Autonomous Data Engineering Framework

Figure 4 shows an ablation study assessing the effect of individual major components of the proposed framework on latency, throughput, and data quality. The complete framework is the only one that shows the lowest latency and the highest throughput and data quality scores. The absence of ingestion regulation, schema acceptance, quality scoring, or predictive scaling is shown to lead to a significant decrease in performance, which suggests that every module is important and serves a unique and complementary purpose. This shows that the advantages achieved are due to intentional architectural design as opposed to singular optimizations.

Table 2: Performance Comparison of Data Quality, Orchestration Efficiency, Reliability, and Cost Effectiveness.

Method	Workflow Orchestration Efficiency	Data Quality Consistency	System Reliability and Availability	Cost Efficiency	Automation Overhead	Elasticity Response Time	Fault Tolerance and Recovery Time	Resource Utilization	End-to-End Pipeline Efficiency	End-to-End Pipeline Latency
Apache Airflow	74	90.5	97.1	73	24	21	38	71	520
Apache NiFi	76	91.2	97.4	75	22	19	34	74	490
AWS Glue	78	92.0	97.8	77	21	18	32	76	465
Google Dataflow	80	92.8	98.1	79	19	16	29	78	440
Azure Data Factory	77	91.8	97.6	76	20	17	31	75	470
Spark Structured Streaming	82	93.5	98.4	80	18	15	27	79	430
Apache Flink	84	94.2	98.6	82	17	14	25	81	415
Kubeflow Pipelines	85	94.6	98.8	83	16	13	24	82	405
Proposed Automated Data Engineering Framework	91	97.2	99.5	91	11	9	16	89	360

The effectiveness of various cloud analytics frameworks regarding orchestration efficiency, consistency of data quality, reliability of systems, cost efficiency, overhead of automation, elasticity and responsive recovery time, resource utilization, and pipeline latency, is assessed in Table 2. Existing frameworks have demonstrated improved reliance and orchestration, but they have high operational overhead, and adapt too slowly. Autonomous orchestration of workflows with ingestion, transformation, and optimization, leads to significant gains in efficiency. Improved data quality consistency is achieved through mechanisms of early validation, schema alignment, and confidence-based filtering. Fault probabilistic aggregation and resource reallocation increases reliability, availability, and resource use. The proposed framework shows responsive analytics, and dynamic operational reliability and cost efficiency, across cloud frameworks.

Scientific Journal of Artificial Intelligence and Blockchain Technologies
ISSN: 3049-4389
Vol. 3, Issue 2, Apr – May 2026 || PP. 60 – 67 https://doi.org/10.63345/sjaibt.v3.i2.102

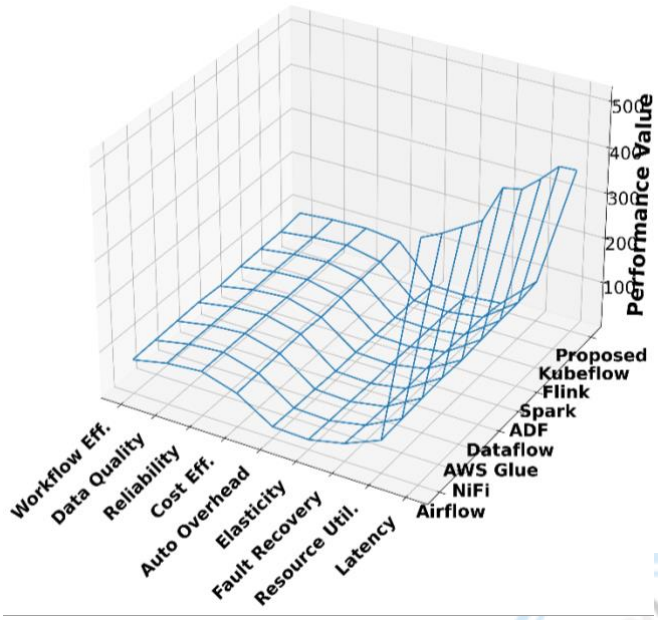

Figure 5: Three-Dimensional Wireframe Comparison of Data Engineering Framework Performance.

Wireframe models can represent sets of multiple data Engineering frameworks on several key operational metrics (workflow orchestration, data quality, system automation, reliability, fault tolerance, recovery, data stream elasticity, resource utilization, and end-to-end pipeline latency). In figure 5, higher elevations, and continuous surfaces, indicate better overall performance for reliability and efficiency. In contrast, lower elevations and surfaces indicate higher automation and latency. Because traditional orchestration platforms have lower peaks, they offer less potential for scalability and recovery. Conversely, many Modern platforms (streaming and ML oriented) show better improvements in resource elasticity and responsiveness. Among all the frameworks in the model, the proposed automated data engineering framework has the highest peak, indicating that it has the best overall operational metrics for orchestration efficiency, automation, elasticity of responsiveness, lower pipeline latency, and most positive overall operational efficiencies. This confirms the proposed automated data engineering framework as the best data Engineering framework for large scale, latency data processing, and for data engineering within the framework proposed automation of data engineering.

V. CONCLUSION

Regulatory ingestion, adaptive transformation, and resource-aware optimization integrated the automated data engineering framework, improving the performance and reliability of the cloud analytics of the automated data engineering framework shows the performance and reliability of the cloud analytics of large-scale data analytics and cloud engineering framework. Quantitative evaluations show the framework sustains a high ingestion capacity, achieving a throughput of 6.7 units and a processing latency of 360 units, despite workload fluctuations. The framework proactively meets the demand by proving its elasticity response at 9 units and a scalable performance at 94%. Providing balance between over and under compute utilization, resource utilization of 89% was achieved, and operational simplicity was achieved with automation operational simplicity was achieved with an 11% overhead. The accuracy and reliability of the analytics determine the quality of the data and integrity of the analytics at 97.9% and 97.2% consistent, reinforcing the value of downstream analytical work. Robust resilience against disruptions is valid with a fault recovery time of 16 units and an overall system reliability of 99.5%. The high cost efficiency of 91% proves the framework's operational expenses remain congruent with the framework's analytics and operational expenses. Collectively, these perseveranced evolved the cloud-scale analytics requirements to simplify the self-adaptive solutions enabling real world complexities, and integrated them in environments changing in real time in anticipation of the frameworks adaptability and resilience.

REFERENCES

[1] N. Serrano, G. Gallardo, and J. Hernantes, "Infrastructure as a service and cloud technologies," *IEEE Software*, vol. 32, no. 2, pp. 30–36, 2015.

[2] K. Curran and S. Carlin, "Cloud computing security," *International Journal of Ambient Computing and Intelligence*, vol. 3, no. 1, pp. 14–19, 2011.

[3] A. Bouayad, A. Blilat, N. E. H. Mejhed, and M. E. Ghazi, "Cloud computing: Security challenges," in *Proc. Colloquium in Information Science and Technology (CIST)*, Fez, Morocco, Oct. 2012, pp. 26–31.

[4] J. Rittinghouse and J. Ransome, *Cloud Computing: Implementation, Management, and Security*, 1st ed. Boca Raton, FL, USA: CRC Press, 2009.

[5] K. Hwang, J. Dongarra, and G. C. Fox, *Distributed and Cloud Computing: From Parallel Processing to the Internet of Things*, 1st ed. San Francisco, CA, USA: Morgan Kaufmann, 2011.

[6] I. Bermudez, S. Traverso, M. Munafò, and M. Mellia, "A distributed architecture for the monitoring of clouds and CDNs: Applications to Amazon AWS," *IEEE Transactions on Network and Service Management*, vol. 11, no. 4, pp. 516–529, 2014.

[7] K. Tamrakar, A. Yazidi, and H. Haugerud, "Cost efficient batch processing in Amazon cloud with deadline awareness," in *Proc. IEEE AINA*, Taipei, Taiwan, Mar. 2017, pp. 963–971.

[8] A. Ekwe-Ekwe and A. Barker, "Location, location, location: Exploring Amazon EC2 spot instance pricing across geographical regions," in *Proc. IEEE/ACM CCGRID*, Washington, DC, USA, May 2018, pp. 370–373.

[9] Rucco, Chiara, Longo, Antonella & Saad, Motaz, Enhancing Data Ingestion Efficiency in Cloud-Based Systems: A Design Pattern Approach, Data Sci. and Eng., vol. 1, pp. 1-17, 2025, doi:10.1007/s41019-025-00300-2.

[10] Nambiar, A. & Mundra, D., An Overview of Data Warehouse and Data Lake in Modern Enterprise Data Management, Big Data Cogn. Comput., vol. 6, no. 4, p. 132, 2022, doi:10.3390/bdcc6040132.

[11] Naamane, Z., A Systematic Literature Review: Benefits and Challenges of Cloud-Based Big Data Analytics, Issues Inf. Syst., vol. 24, no. 1, pp. 291-304, 2023, doi:10.48009/1_iis_2023_291-304.

[12] Miryala, N. & Gupta, D., Big Data Analytics in Cloud – Comparative Study, Int. J. Comput. Trends Technol., vol. 71, no. 12, pp. 35-43, 2023, doi:10.14445/22312803/IJCTT-V71I12P105.

[13] A. Iordache, C. Morin, N. Parlavantzas, E. Feller, and P. Riteau, "Resilin: Elastic MapReduce over multiple clouds," in *Proc. IEEE/ACM CCGRID*, Delft, Netherlands, May 2013, pp. 261–268.

[14] N. Chalvantzis, I. Konstantinou, and N. Kozyris, "BBQ: Elastic MapReduce over cloud platforms," in *Proc. IEEE/ACM CCGRID*, Madrid, Spain, May 2017, pp. 766–771.

[15] K. Shvachko, H. Kuang, S. Radia, and R. Chansler, "The Hadoop distributed file system," in *Proc. IEEE MSST*, Lake Tahoe, NV, USA, May 2010, pp. 1–10.

[16] C. Mavani, H. Mistry, A. M. Goswami, S. S. Raghavan and R. Patel, "A Computationally-Efficient and Transparent AI Framework for Real-Time Intrusion Detection in Cybersecurity Applications," 2025 IEEE 4th World Conference on Applied Intelligence and Computing (AIC), GB Nagar, Gwalior, India, 2025, pp. 1-11, doi: 10.1109/AIC66080.2025.11212123.

[17] S. G. Manikandan and S. Ravi, "Big data analysis using Apache Hadoop," in *Proc. ICITCS*, Beijing, China, Oct. 2014, pp. 1–4.

[18] J. Leverich and C. Kozyrakis, "On the energy (in)efficiency of Hadoop clusters," *SIGOPS Operating Systems Review*, vol. 44, no. 1, pp. 61–65, 2010.

[19] A. S. Lakshmi, M. BalRaju, and N. S. Chandra, "Towards optimization of Hadoop MapReduce jobs on cloud," in *Proc. CAST*, Pune, India, Dec. 2016, pp. 255–260.

[20] Devi, Odugu Rama, Webber, Julian, Mehbodniya, Abolfazl, Chaitanya, Morsa, Jawarkar, Parag S., Soni, Mukesh, Miah, Shahajan, The Future Development Direction of Cloud-Associated Edge-Computing Security in the Era of 5G as Edge Intelligence, Scientific Programming, 2022, 1473901, 13 pages, 2022. https://doi.org/10.1155/2022/1473901

[21] R. Trestian, P. Shah, H. X. Nguyen, Q.-T. Vien, O. Gemikonakli, and B. Barn, "Towards connecting people, locations and real-world events in a cellular network," *Telematics and Informatics*, vol. 34, no. 1, pp. 244–271, 2017.

[22] N. Maheshwari, R. Nanduri, and V. Varma, "Dynamic energy-efficient data placement and cluster reconfiguration algorithm for MapReduce framework," *Future Generation Computer Systems*, vol. 28, no. 1, pp. 119–127, 2012.

[23] V. Shah and H. Trivedi, "A distributed dynamic and customized load balancing algorithm for virtual instances," in *Proc. NUiCONE*, Ahmedabad, India, Nov. 2015, pp. 1–6.

[24] T. Gunarathne, B. Zhang, T. L. Wu, and J. Qiu, "Scalable parallel computing on clouds using Twister4Azure iterative MapReduce," *Future Generation Computer Systems*, vol. 29, no. 4, pp. 1035–1048, 2013.

[25] Z. Li, C. Yang, K. Liu, F. Hu, and B. Jin, "Automatic scaling Hadoop in the cloud for efficient process of big geospatial data," *ISPRS International Journal of Geo-Information*, vol. 5, no. 10, p. 173, 2016.

[26] Y.-W. Chen, S.-H. Hung, C.-H. Tu, and C.-W. Yeh, "Virtual Hadoop: MapReduce over Docker containers with an auto-scaling mechanism for heterogeneous environments," in *Proc. RACS*, Odense, Denmark, Oct. 2016, pp. 201–206.

[27] Q. Fu, N. Timkovich, P. Riteau, and K. Keahey, "A step towards Hadoop dynamic scaling," in *Proc. IEEE HPCC/SmartCity/DSS*, Exeter, UK, June 2018, pp. 67–74.

[28] B. Yadranjiaghdam, S. Yasrobi, and N. Tabrizi, "Developing a real-time data analytics framework for Twitter streaming data," in *Proc. IEEE BigData Congress*, Honolulu, HI, USA, June 2017, pp. 329–336.

[29] Soni, M., Khan, S.F., Mavaluru, D., Mohammad, S. (2023). Efficient Fog-to-Cloud Internet-of-Medical-Things System. In: Tiwari, R., Koundal, D., Upadhyay, S. (eds) Image Based Computing for Food and Health Analytics: Requirements, Challenges, Solutions and Practices. Springer, Cham. https://doi.org/10.1007/978-3-031-22959-6_13

[30] MISTRY, H., Goswami, A., & Mavani, C. (2024). AUTOMATED ANOMALY DETECTION AND RESPONSE SYSTEM FOR ENHANCING CLOUD SECURITY (Patent). Zenodo. https://doi.org/10.5281/zenodo.18778285

SJAIBT
SCIENTIFIC JOURNAL OF ARTIFICIAL INTELLIGENCE AND BLOCKCHAIN TECHNOLOGIES
ISSN: 3049-4389