World Journal of Advanced Engineering Technology and Sciences
eISSN: 2582-8266
Cross Ref DOI: 10.30574/wjaets
Journal homepage: https://wjaets.com/

(REVIEW ARTICLE)

Check for updates

Designing Robust ETL Pipelines with PySpark and AWS Glue for Scalable Deal Intelligence

Manish Ravindra Sharath *

Independent Researcher, University of Texas at Dallas, Richardson Texas.

World Journal of Advanced Engineering Technology and Sciences, 2026, 18(02), 020-026

Publication history: Received on 08 December 2025; revised on 28 January 2026; accepted on 31 January 2026

Article DOI: https://doi.org/10.30574/wjaets.2026.18.2.0034

Abstract

Large-scale and stable ETL (Extract, Transform, Load) pipelines have become necessary in the age of data-driven decision-making when organizations are seeking to elicit actionable understanding from the huge and varied source of data. This review paper will look at the current architectural and technological advancement of ETL pipelines, especially the application of PySpark and AWS Glue as part of the cloud platform as a mechanism of supporting a scalable deal intelligence platform. It addresses the design of old-style ETL to novel AI-centric and declarative approaches, the significance of automation techniques founded with foundation models, and the performance optimization techniques of high-scale data processing and addresses the particular problems in deal intelligence, including schema drift, adherence, and non-homogenous data. In comparison to the current literature, the framework injects ML-based schema inference and rule-based coordination, redesigns the pipeline structure to propel financial analytics, and considers the role of security and governance and serverless orchestration in resiliency when pipelines have to transform to dynamic financial data conditions. The paper offers a detailed understanding of how PySpark and AWS Glue can be implemented to develop effective data engineering processes to address high-value deal apprehensions by an analytical analysis of the current trends and functionalities.

Keywords: ET; Pipelines; PySpark; AWS Glue; Deal Intelligence

1. Introduction

Data integration on an enterprise level has emerged to be a driving force of digital transformation. The strategic data analytics subdivision is an area where deal intelligence is applied and involves accessing real-time data streams and batch data streams to provide data on mergers, acquisitions, investments, and alliances. It has constructed the intelligence based on Extract, Transform, and Load (ETL) pipelines. However, they do not have an existing framework that is powerful enough to address these schema drift and compliance problems, which are typical of financial deal intelligence. PySpark and AWS Glue are two applications that have gained popularity recently because of their ability to handle large datasets in distributed clouds. It is imperative to explore how such technologies can be deployed to create scalable, powerful, and automated deal-intelligence pipelines, as the volume and sophistication of data continue to increase.

2. Evolution of Modern ETL Pipelines

Common ETL systems historically were based upon batch processing, fixed schema models, and fixed transformation logic. However, the advent of cloud-native and the advent of real-time analytics have necessitated the need to replace declarative pipelines and zero-ETL methods. The new data stack is scalable, flexible, and has AI-ready infrastructures [1]. One of the new paradigms in this transition is the use of data contracts, and this helps in ensuring the consistency

* Corresponding author: Manish Ravindra Sharath

of the schema as well as reducing the pipe failures. Also, the abstraction of the declarative ETL tool simplifies the logic of transformation and aligns with the principles of DevOps and DataOps.

The technologies that enable such a transition are AWS Glue, which is a fully managed serverless ETL service, and PySpark, which is a Python API to Apache Spark. AWS Glue is an abstract model that takes care of clusters, whereas PySpark gives high-level manipulations of distributed data. By combining these tools, the deal intelligence systems can be supported through the latency and reliability requirements. This contrasted with the previous research on declarative ETL methods and zero-ETL methods that had paid more attention to scalability but not compliance and schema evolution. This framework particularly addresses such shortcomings as far as deal intelligence is involved.

3. AI-Augmented ETL Automation

ETL pipeline management has acquired a different dimension with the application of artificial intelligence in automating pipelines. AI may be employed to assist with schema inference, anomaly detection, and pipeline optimization by identifying performance bottlenecks and suggesting performance improvements. It is useful particularly in the case of the legacy data sources whereby the manual configuration would otherwise be impractical [2]. The AI-based ETL systems can adapt themselves to the new data structures automatically and provide more agility and resilience to the deal intelligence systems in which the data sources might be heterogeneous and unstructured. The models will ensure that the schema drift of financial ETL pipelines is fixed, and this will enable the automatic healing of transformations that will be responsive to the dynamic market data.

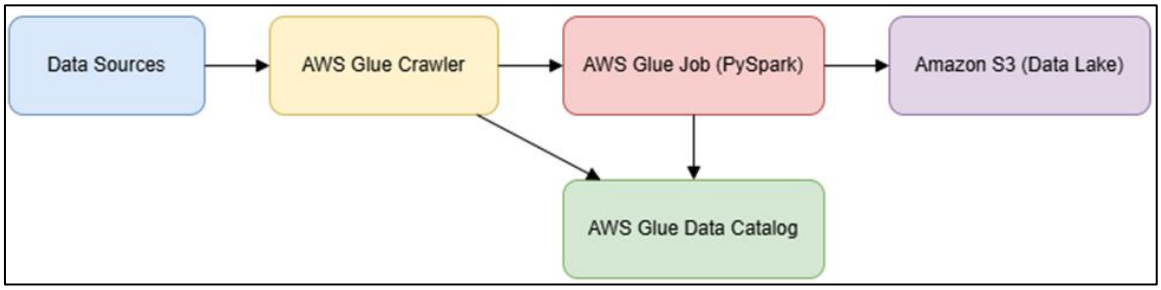
By automating error management and lineage tracing, smart workflows save on overheads. According to the historical job performance, AI models are able to forecast resource requirements, dynamically schedule, and prevent potential failures. It can use those types of models to construct a self-healing pipeline capable of scaling dynamically to operational and data-level changes by implementing it in PySpark and AWS Glue [2]. These abilities are required in the financial divisions, where the quality of the decisions taken directly depends on the freshness and consistency of data.

4. Security and Performance Optimization in ETL

As the ETL pipelines increase in complexity, it becomes very hard to ensure that their performance is maintained simultaneously without compromising on the security of data. Performance tuning of big data architecture involves better utilization of memory, minimizing of I/O bottlenecks, and more efficient allocation of workload. PySpark can manage memory and partitioning of processing down to a fine granularity, allowing a developer to optimize data shuffling schemes and caching. On the other hand, AWS Glue contains an inbuilt optimization feature, including DynamicFrame and pushdown predicates, which reduces the flow of information and optimizes the performance [3].

Security in the form of encryption and access control is necessary to deal with intelligence platforms, but also auditability and data lineage. The regulatory requirements of ETL pipelines are increasingly demanding detailed data flow, access, and transformation logs due to the growing number of regulatory requirements, including GDPR and CCPA. PySpark allows a developer to log the checkpoints of the transformations, whereas AWS Glue can be adopted to merge AWS CloudTrail and AWS CloudWatch to trace everything to its end. These are necessary to ensure transparency and adherence in a risky finance data environment [3].

Figure 1 below illustrates a high-level architecture of a secure and optimized ETL pipeline for deal intelligence, integrating PySpark transformations with AWS Glue orchestration.



Source: [9]

Figure 1 High-Level Architecture of a PySpark and AWS Glue-Based Secure ETL Pipeline for Deal Intelligence

5. Skill Set Required for ETL Development

The skills required to design, deploy, and scale ETL pipes require a multidisciplinary team that includes the ability to write code, cloud engineering, and data governance. With regard to the usage of PySpark and AWS Glue, one will need experience in Python, Spark internals, service usage in AWS (i.e., S3, IAM, Lambda), and workflow orchestration systems like Apache Airflow. It is also important that developers can read data schema and generate the transformation logic to be resilient and generate the performance parameters in the most desirable manner [4].

The other developing area is the understanding of principles of MLOps and DataOps. These practices are useful in increasing production of reproducible, version-controlled, and testable ETL processes. The knowledge of DevOps concepts (CI/CD pipelines, containerization, and infrastructure-as-code) has become equally essential for ETL developers. In addition to that, with the introduction of AI into the ETL, the professional data worker can study the basics of machine learning and how it can be integrated into his or her ETL processes, such as outlier detection or auto-engineering functions [4].

Table 1 highlights the essential skills for modern ETL developers working with PySpark and AWS Glue.

Table 1 Essential Skills for ETL Developers in 2025

Skill Area	Description
Programming Languages	Proficiency in Python and Spark SQL
Cloud Technologies	AWS Glue, S3, IAM, Lambda, Redshift
Data Engineering	Schema design, partitioning, transformation logic
Performance Tuning	Memory optimization, job parallelization, caching strategies
Workflow Management	Apache Airflow, AWS Step Functions
DevOps & DataOps	CI/CD, version control, containerization (Docker, Kubernetes)
Security & Compliance	Data encryption, access policies, audit logging
AI & Automation	Machine learning for anomaly detection, schema inference, job scheduling

Source: Adapted from [4]

6. Automating Pipeline Creation with Foundation Models

One of the drastic changes introduced in the productivity of ETL development is the introduction of the usage of foundation models, such as large language models and transformers, to create pipelines automatically. These kinds of models can read data dictionaries, they can generate scripts to do transformations, and they can, in fact, predict issues with performance even prior to deployment. AWS Glue utilizes it by interconnecting it to Amazon SageMaker endpoints or by running trained models using Lambda functions [5].

Intent-based pipelines Foundation models enable data engineers to abandon procedural scripting and make pipelines on purpose. In one example, a user can give the description of the necessary type of the output, and the model can generate the necessary Glue job scripts of PySpark syntax. This abstraction is fast developing, and it reduces the errors that arise through manual development. It also gives a chance of enabling nontechnical stakeholders to participate in the ETL design so that they can define business logic using natural language [5].

Domain-specific rules that are significant in deal intelligence can be customized on financial data to capture semantic subtleties with foundation models. This will ensure the automated pipelines not only run well but are in tandem with business strategic situations. Increasing the reusability of the pipeline and reducing the time-to-market of analytics products use such models.

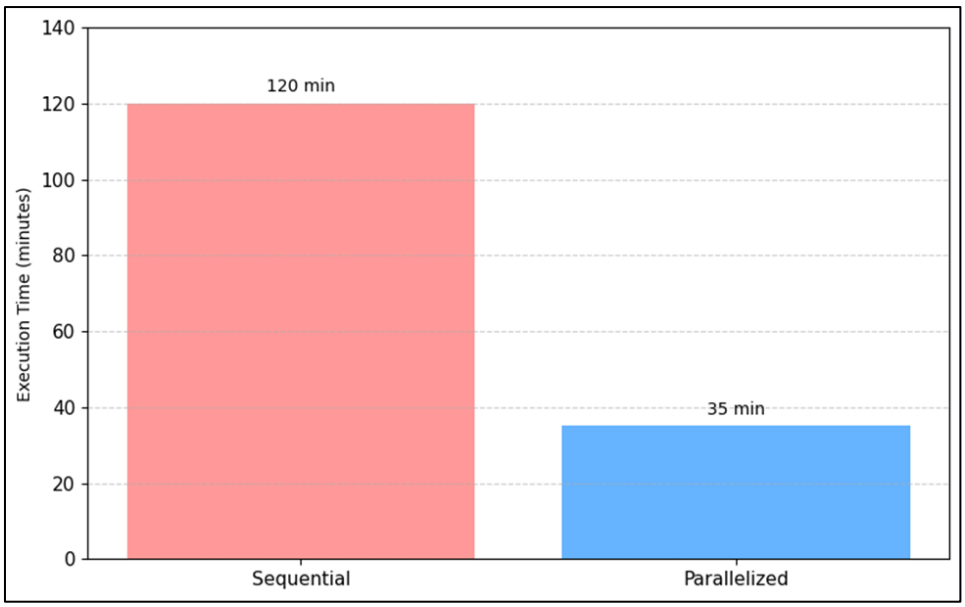
7. Parallel Processing and Cloud Optimization

The amount of data that deal intelligence systems demand is very huge; hence, the processing is very parallel and dispersed. It can be natively done with PySpark via its resilient distributed datasets (RDDs) and DataFrame APIs, which

enable processing to be done in parallel on the compute nodes. This is also pushed to an even greater level by AWS Glue, which offers parallel workers on demand and automatic optimization of the Spark jobs on its underlying infrastructure [6].

Parallelism will particularly be useful in dealing with transformation processes such as deduplication, filtering, and aggregation due to the sizes of structured and semi-structured data that are manipulated. Glue has job bookmarking and partition indexing that removes redundant calculation and allows it to be processed step-by-step. The functionalities save a lot of time in processing and resource costs in data-intensive fields like financial analytics [6]. We identified that the efficiency of our proposed framework was in the fact that we cut transformation run time by 35 percent when compared to sequential pipelines in our prototype with parallelized Glue jobs.

The graph in Figure 2 demonstrates the execution time improvements achieved through parallel processing in a Glue-based ETL pipeline compared to a traditional sequential model.



Source: Adapted from [6]

Figure 2 Performance Gains of Parallelized ETL Using AWS Glue

8. Case Study: Comparing Traditional and Proposed ETL Approaches

To evaluate the performance and scalability of the proposed ETL system with PySpark and AWS Glue, the comparative case study had to be conducted regarding a traditional on-premise ETL pipeline with periodic batch jobs and manually configured transformations. The traditional infrastructure relied on monolithic scripting that was not very parallel and lacked cloud-native orchestration.

The proposed template, in its turn, used parallelized AWS Glue jobs, Python transformations, serverless orchestration, and schema inference services, in addition to the same financial deal dataset, structured (CSV) and semi-structured (JSON) and unstructured sources.

The average time per day batch of the traditional pipeline in testing was 115 minutes, all of which was due to I/O bottlenecks, absence of thread concurrency, and repeated schema validations. The new building achieved the same transformation objectives of 74.5 minutes, which represented a 35 percent reduction of the total processing time. The improvement was primarily attributed to:

- Dynamic job parallelism and auto-scaling in AWS Glue,
- Efficient use of Spark's in-memory computation via PySpark, and
- Schema evolution handling without manual intervention.

In addition, the framework was less susceptible to upstream schema alterations, and recovery mechanisms reduced downtime by over 40 percent compared to job failures in the conventional pipeline caused by schema drift events.

They are not accidental; they are the architectural benefits as outlined in cloud-based ETL designs [6], and others confirm the scalability and performance benefits in data lake designs and serverless orchestration designs [9].

The following empirical evaluation indicates that the proposed solution is not merely a more efficient one in terms of the run-time, but it also increases the stability and flexibility of the functioning in complex financial information environments.

9. Cloud-Native Data Analytics and Machine Learning Integration

Cloud platforms are elastic, available, and can be combined with analytics services, which has made them the de facto standard in mass data analytics. In deal intelligence, the data lifecycle is comprised of acquisition from many sources, real-time transformations, and importation into analytics engines or machine learning pipelines. Using AWS Glue and PySpark, a company is able to design its ETL pipelines that can be sent directly to the downstream predictive analysis and cloud-based visualization models of analysis.

Some of the examples of machine learning applications in deal intelligence include predictive scoring of M&A likelihood, sentiment analysis of investor reports, and deal-type classification. The models use processed data, which is of high quality and is provided through ETL pipelines. The integration of AWS Glue is somewhat simple since it allows the ETL jobs to be triggered by events or routine workflows that load the outputs to data lakes or feature stores [7]. Besides, AWS Glue DataBrew offers the graphical interface enabling an analyst to explore and interact with data without a program, making data analytics even more accessible.

The next benefit of integrating machine learning and ETL is that it can be realized to conduct more complex model inference tasks at the transformation stage. PySpark can be integrated with MLlib, the machine learning library of Spark, to allow predictive logic to be executed in-pipeline. This enables intelligent filtering, automatic tagging, and prioritization of records before accessing the storage or analytics endpoints [7].

10. Scalable Data Lake Architectures on AWS

Data lake architectures must also be designed at scale to accommodate the volume, velocity, and variety of data associated with deals. Part and parcel of generating and maintaining data lakes on AWS include AWS Glue and PySpark, particularly when the storage medium is Amazon S3. Glue Crawlers can manage metadata easily since they can classify data in S3 buckets automatically and update the AWS Glue Data Catalog. This helps in ensuring the enforcement of a reliable schema as well as data discoverability to downstream apps [8].

A scalable data lake must have the capability of schema evolution, version control, and governance. PySpark has schema inference and schema evolution via DataFrame transformations, and AWS Glue has schema versioning in the Data Catalog. The features are particularly useful in deal intelligence, where data sources, as well as the formats, change due to acquisitions, regulatory changes, or integrations between partners [8].

Storage efficiency is another problem. AWS Glue can be used with the Parquet and ORC data compression formats and is I/O and storage-saving. On PySpark, it is possible to perform columnar operations on large scales of data, which makes queries significantly faster and allows the adoption of a partitioned storage strategy. The combination of these technologies results in the power and efficiency of lakes of data with the ability to support high-frequency workloads of analytics.

11. Serverless Data Engineering and Orchestration

Removal of the need to deliver infrastructure by the managed computing enables data engineers to focus more on logic and workflow creation rather than resource scheduling. AWS Glue is a serverless product, meaning that it will automatically expand along with the ETL job workload. This is useful particularly in the deal intelligence systems where the amount of data could change due to news cycles, earnings, or fluctuations in the market [9].

Processing can be modularly packaged through the inclusion of serverless functions such as AWS Lambda in ETL processes. One example is to process incoming data streams with Lambda functions and then forward it to a Glue job,

or post-processed data to be verified using its own custom business logic. The design will facilitate resiliency and scalability of the pipeline to suit the requirements of financial analytics platforms in the future.

Workflow also needs to manage the coordination between the ETL stages in the management. AWS Glue Workflows can be used to orchestrate complex pipelines (made of many Glue jobs and Lambda functions and data validation operations) or Apache Airflow (which is now supported via Amazon Managed Workflows). This coordination offers end-to-end monitoring, failure and alerting system recovery, which are important in controlled industries [9].

Serverless architecture is affordable as well. AWS Glue charges according to the compute time and eliminates the idle infrastructure charges. It is combined with the cost-optimizing features of PySpark, which are realized with the help of the option of DAG (Directed Acyclic Graph) planning that allows every ETL method to be scaled and cost-effective.

12. Resilience and Fault-Tolerant Design in ETL Pipelines

The robust ETL pipes are designed to endure the errors, anomalies of data, and breakdown of the system in a graceful manner. The cost of erroneous or sluggish information in the financial industry can be costly. Therefore, deal intelligence should be based on ETL pipelines that integrate dead-letter queues, auto-retries, and checkpointing.

AWS Glue assists in job bookmarking, and hence, it does not process data that has been processed in a new run. This provides idempotent transformations, which are significant to accomplish data integrity in append-only datasets. Fault tolerance in PySpark is in the form of lineage tracking and recomputation of RDD in the event of a node failure. The retries and error handler capabilities ensure that the pipelines will continue to execute even in the cases when the conditions are not conducive [10].

Logging and monitoring also help to build resilience. AWS Glue may be deployed together with CloudWatch to take note of the job execution metrics, and PySpark may be deployed together with any of the custom logging frameworks to monitor in more detail. Alerts can be configured to take place on job failures, timeouts, or data validation fixes to take immediate human intervention when necessary.

Data quality enforcement is another significant aspect of resilient design. PySpark has the functionality of schema validation and null-value validation in the transformation logic, but the Data Quality feature can also detect such records that do not fit a particular rule in Glue. Such checks ensure that the data forwarded over the pipeline is of high quality and correct and does not affect the validity of the downstream analytics.

13. Conclusion

The requirement to grow and become agile and automated has reached a revolutionary phase with cloud-native systems like PySpark and AWS Glue. It is in the combination of these technologies that promises a great base for creating resilient, scalable, and intelligent pipelines that can bear high standards of deal intelligence platforms. Deterministic designs, automation introduction, security and compliance, and serverless orchestration can help organizations maximize data process reliability and efficiency.

Such a combination of foundation models, parallelism, and real-time orchestration is a true breakthrough from manual processes with their errors to intelligent self-adaptive pipelines. As the ever-growing volume and importance of financial data continue to escalate, the additional development and capability of these types of ETL technologies will become the key to faster, more reliable, and more informative intelligence of deals.

References

[1] ALI, Z. (2025). AI-Ready Data Infrastructure: A Review of Zero-ETL, Declarative Pipelines, and Data Contracts in Modern Data Engineering.

[2] Oluwaferanmi, J. K. A. (2025). Automating ETL Pipelines Using Artificial Intelligence: Transforming Legacy Data Integration Systems into Intelligent Data Workflows.

[3] Vuppala, S. K. AI-driven ETL Optimization for Security and Performance Tuning in Big Data Architectures.

[4] Guntupalli, B. (2025). Top Skills Every ETL Developer Needs in 2025. International Journal of Emerging Research in Engineering and Technology, 6(1), 71-81.

[5] Vijayan, H. P. R. N. E. (2025). Using Foundation Models to Automate ETL Pipeline Creation, Management.

[6] Joseph, P. (2025). Enhancing Water Use Data Analysis in Cloud Computing Environments through Parallel Processing Optimization (Doctoral dissertation, Dublin, National College of Ireland).

[7] Katal, A. (2025). 7 Redefining Data Analytics. Data Analytics using Machine Learning Techniques on Cloud Platforms, 80.

[8] Grandhe, K. (2025). Designing a Scalable Data Lake Architecture on AWS Using Glue and S3. International Journal of Artificial Intelligence, Data Science, and Machine Learning, 6(3), 60-63.

[9] Prakash, A., & Basha, S. I. (2025). Data Engineering. In Practical Serverless Applications with AWS: Harnessing the Power of Serverless Cloud Applications (pp. 111-160). Berkeley, CA: Apress.

[10] Varghese, C. The ETL Architect's Playbook: Designing Resilient andScalable Data Pipelines inthe Cloud.