International Journal of Engineering & Extended Technologies Research (IJEETR)

|ISSN: 2322-0163| www.ijeetr.com | A Bimonthly, Peer Reviewed, Scholarly Indexed Journal |

| Volume 4, Issue 6, November-December 2022 |

DOI:10.15662/IJEETR.2022.0406013

Distributed Data Engineering Pipelines for Real-Time Insurance Claim Processing

Shashikala Valiki

Independent Researcher, India

shashikala.valiki.researcher@gmail.com

ABSTRACT: In today's data-driven world, organizations seek to rapidly and accurately convert multi-source data into proactive intelligence to drive smart business decisions. Allowing real-time or near-real-time ingestion, evaluation, and scoring of operational data can improve the speed and efficiency of existing business processes, enabling organizations to uncover risks and opportunities faster than traditional batch-oriented approaches. This paper discusses the ground-up architectural design considerations required to allow real-time and end-to-end processing of multi-source claim data at an insurance firm.

The key architecture design principles focus on ingesting and pretreating operational data, stream processing alternatives with integrated data quality and compliance capabilities, and defining and operationalizing real-time scoring and evaluation pipelines with observability and monitoring capabilities. Such a design is instrumental in developing distributed data engineering pipelines that efficiently link cloud-based microservices with inevitable data lakes and warehouses. While the work focuses specifically on claim processing, the principles are applicable to other industry verticals where event-driven implementations of core business processes are required.

Keywords: Real-Time Data Processing, Data Engineering Pipelines, Insurance Claims Analytics, Event-Driven Architecture, Stream Processing, Multi-Source Data Integration, Real-Time Scoring, Cloud Microservices, Data Lakes and Warehouses, Operational Intelligence, Data Quality and Compliance, Observability and Monitoring.

I. INTRODUCTION

The insurance claim process is typically perceived as time-consuming and bureaucratic, often taking weeks or months for resolution. Such delays can lead to customer dissatisfaction and dismay, particularly in contexts such as loss of property or accidents. To better express empathy for customers in such situations, insurance players are under pressure to reduce the claim resolution time. In contrast to a traditional approach that aggregates claims on a batch basis, an architecture following the principles of data engineering allows for the assessment of every single claim almost immediately (within seconds), without requiring human intervention. With a limited set of rules and stream processing, claims can be authorized heuristically. For others cases, a machine learning model can provide a score representing the propensity for fraud, risk category, or other relevant aspects. Claims with low enough risk can be directly authorized, while those above a threshold can undergo more careful assessment.

Such architecture operates a distributed data engineering pipeline in the back end, and its main elements can be clustered into three major groups. Their deployment in an industrialized and automated approach is the final piece of remaining considerations on effective architecture for real-time processing of insurance claims. The focus is on deployment strategies suitable in an enterprise context, considering aspects such as containerization, observability, monitoring, and alerting.

1.1. Overview of the Document Structure

The discussion proceeds in the following parts. First, a number of background considerations fundamental to the design of a pipeline support¬ing real-time processing of insurance claim data are introduced. Next, a number of corresponding architectural principles are established, providing a guiding framework for an engineering pipeline that supports ingestion, pretreatment, and downstream analytics.

International Journal of Engineering & Extended Technologies Research (IJEETR)

|ISSN: 2322-0163| www.ijeetr.com | A Bimonthly, Peer Reviewed, Scholarly Indexed Journal |

| Volume 4, Issue 6, November-December 2022 |

DOI:10.15662/IJEETR.2022.0406013

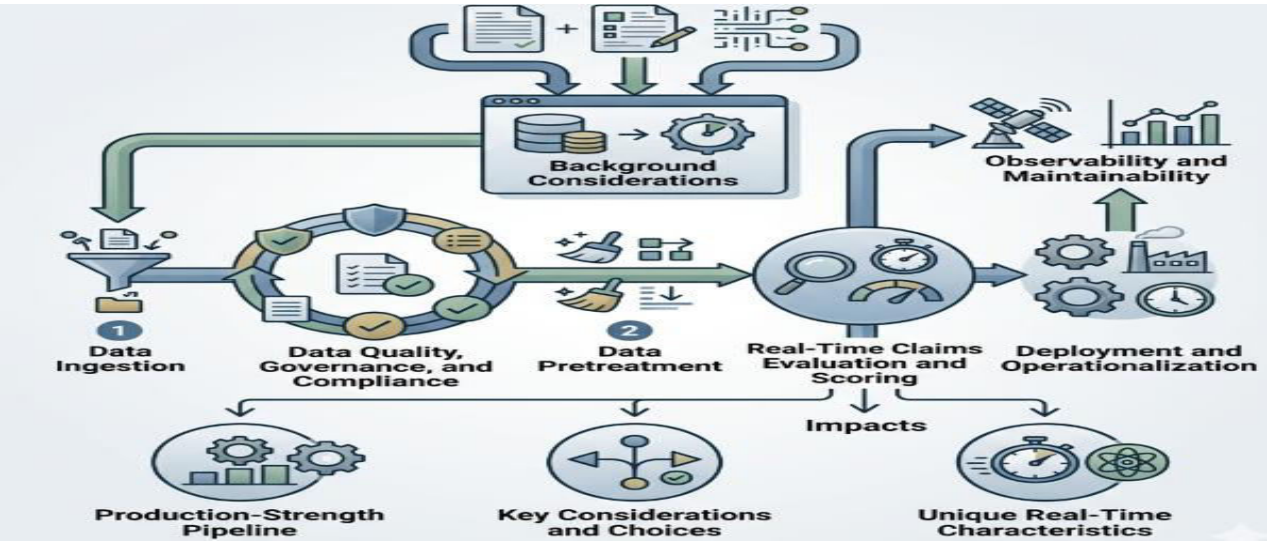


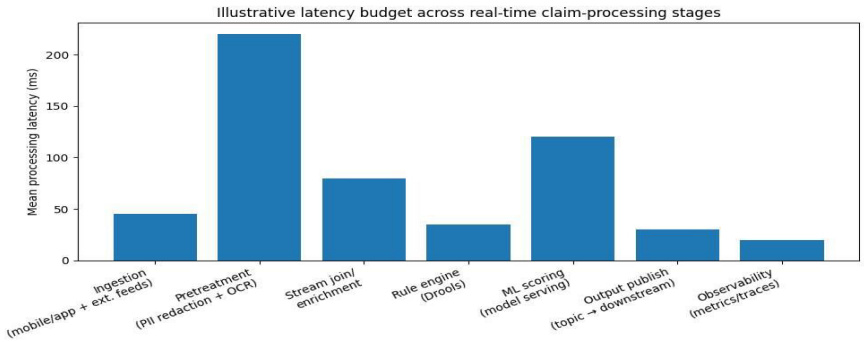
Fig 1: Architectural Principles for Real-Time Insurance Claim Analytics: A Framework for High-Integrity Ingestion, Governance, and Operationalization

Specific attention is devoted to issues surrounding data quality, governance, and compliance, before focusing on the real-time evaluation and scoring of received claims. The conversation then turns to the deployment and operationalization of a distributed pipeline, including strategies for maintainability and observability. The aim is not to propose a prescriptive reference architecture, but rather to highlight key considerations and choices that underpin a production-strength data pipeline. While many of these concerns are found in any type of data processing, they take on unique characteristics when processing insurance claim data in a real-time context. In this situation, the functional requirements for a processing pipeline intersect profoundly with issues of data quality, governance, compliance, and observability. In subsequent sections of this document, a number of these considerations are examined in depth.

II. BACKGROUND AND MOTIVATION

The gradual move from batch processing towards near real-time architectures supports business agility in insurance claim processing. Faster claim evaluations lead to earlier payments, higher customer satisfaction, and lower costs. A dedicated process maintains supervisors' and claims handlers' focus on paying genuine claims, deterring fraudulent claims, and improving overall customer experience.

Optimizing real-time claim evaluation latency leads to a sub-second design that, together with sufficient throughput, justifies a blended rule-based and AI/ML scoring approach. Utilizing the automated and real-time capabilities of microservices for rule and score maintenance, and embedding them in a general-purpose event processing and streaming framework, operationalizes the processing of incoming and outgoing changes to the scoring data context. Continuous proofing, continuous data validation, and pipeline observability guard against erroneous rule content or execution.



International Journal of Engineering & Extended Technologies Research (IJEETR)

|ISSN: 2322-0163| www.ijeetr.com | A Bimonthly, Peer Reviewed, Scholarly Indexed Journal |

| Volume 4, Issue 6, November-December 2022 |

DOI:10.15662/IJEETR.2022.0406013

Equation A. End-to-end latency of a distributed claim pipeline

Pipeline as sequential stages (ingestion → pretreatment → enrichment → scoring → publish → monitoring):

1. Let the pipeline have stages $i = 1..n$
2. Each stage has:
   - processing time t_i (compute time)
   - waiting/queueing time q_i (backlog due to load)
3. End-to-end latency:

$$L_{\text{e2e}} = \sum_{i=1}^{n} (t_i + q_i)$$

4. If you ignore queueing (best case / lightly loaded), the "latency budget" is:

$$L_{\text{proc}} = \sum_{i=1}^{n} t_i$$

2.1. Key Considerations for Effective Claim Processing Architecture

Data ingestion and pretreatment must support all the other architectural building blocks throughout their life cycles and provide clean data for the downstream components. Stream processing frameworks are necessary for the timely and scalable governance of the large volumes of information arriving at the rate of every few seconds to many per minute. Data-driven processing engines require a high degree of automation and transparent deployment in preconfigured cloud environments. Minimizing latency while observing throughput and determinism is critical for building and operationalizing machine learning-enabled pipelines. Automated and predictive observability and monitoring are essential for managing and proactively mitigating processing failures. Event-based end-to-end systems for real-time claim processing are inherently observable at all levels for root cause identification and analysis. The success or failure of a deployed data pipeline reflects itself in how quickly and effectively it is recognized and acted upon.

III. ARCHITECTURAL PRINCIPLES FOR REAL-TIME CLAIM PROCESSING

The architectural considerations outlined above can inform a customized solution for a real-time claims processing use case that manages external data ingestion and permanent storage alongside distributed pipelines. Claims and supporting documents such as photographs, recorded calls, and torn documents are tagged by a quality control team before being made accessible to the processing service. After that, a pretreatment pipeline removes personally identifiable information from the documents for privacy reasons and obtains structured data from the pictures. These pretreatment steps create multiple topics for the grouped messages, which are functionally map-reduced into a single topic, assuring that the latter is always available for the scoring pipeline.

Building on the earlier considerations regarding latency, throughput, and determinism, a stream processing framework is employed for both data-in and data-out of the scoring service. The processing is completed in fit-for-purpose microservices, and new deployments are seamlessly orchestrated to support external ingestion and processing overheads during load spikes. Once the claim is evaluated and labeled as fraud or legit, the output message is delivered on a different topic, where it is absorbed by a follow-up pipeline executing investigations for identified fraud cases. Two further pipelines supervising the quality of the classification remain to be defined, focusing on the alignment of the estimated_probability with that of previously checked claims and on the identification of unclear cases by using an ensemble strategy.

3.1. Data Ingestion and Pretreatment

Modern software architectures and services directed toward real-time insurance claim evaluation and scoring, focus on the processing of claim execution results, typically obtained from a mobile application using claim submission and other document management capabilities. The quality of such application-deployed documents is crucial for the acceptance of claims. In this context, the overall performance of claim processing can be maximized with a data engineering pipeline that efficiently preprocesses all data injected into operational streams.

International Journal of Engineering & Extended Technologies Research (IJEETR)

|ISSN: 2322-0163| www.ijeetr.com | A Bimonthly, Peer Reviewed, Scholarly Indexed Journal |

| Volume 4, Issue 6, November-December 2022 |

DOI:10.15662/IJEETR.2022.0406013

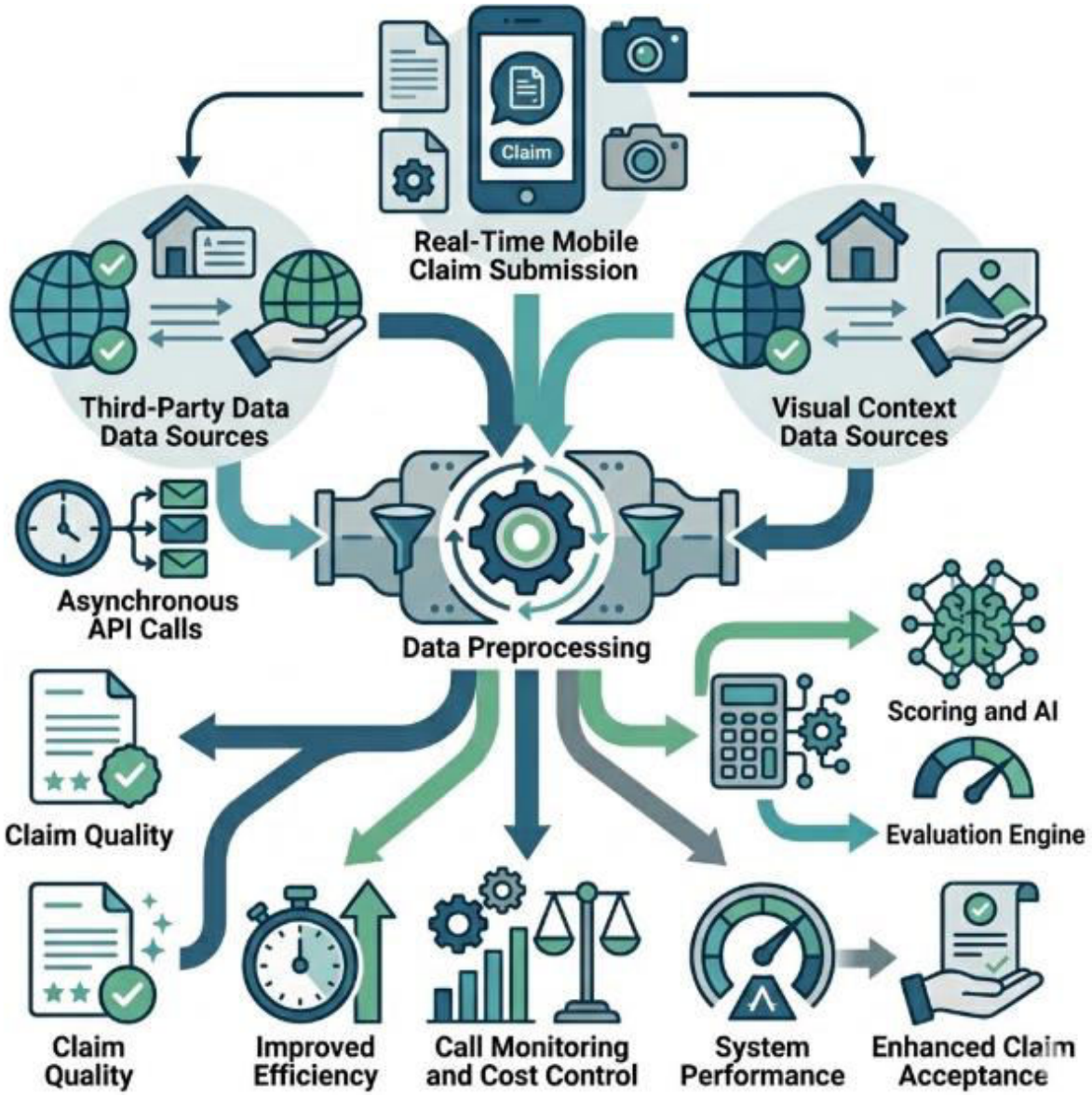


Fig 2: Optimizing Real-Time Mobile Insurance Claim Evaluation through a Preprocessing Data Engineering Pipeline: A Study on Maximizing Processing Performance

Data Ingestion and Pretreatment Processing engines, including all types of stream processing engines—including Spark Streaming, Apache Flink, and Akka Streams—support integration with incoming data sources, whether organized in file systems or delivering data in real-time events. Claims must always be accompanied by data from third-party providers (e.g., external address validation, clime history, visual sources related to an accidnet) and such data elements are usually obtained in an asynchronous pattern. Any of these real-time calls can be costly, requiring monitoring and batching, or possibly even and service abstraction to care for any backup ahead of claim submission. Such data injections into the operation processing flow are typically non-delaying and prefilling features directed toward improving claim quality, nevertheless they deserve a special treatment module.

3.2. Stream Processing Frameworks

Stream processing frameworks support superset and closed-world scenario use cases; they provide constant throughput, low-latency pipelines capable of scaling-to-zero when no events are present. Reliable, stateless (or same-state), fast, and lightweight processing engines, and low-latency implementation require careful consideration of programming patterns. Location and source of the event(s) results and their importance relative to evaluation-time constraints effect data movement—timeliness has precedence over cost—but special treatment is needed for speeding up rule

International Journal of Engineering & Extended Technologies Research (IJEETR)

|ISSN: 2322-0163| www.ijeetr.com | A Bimonthly, Peer Reviewed, Scholarly Indexed Journal |

| Volume 4, Issue 6, November-December 2022 |

DOI:10.15662/IJEETR.2022.0406013

evaluations. Batch processing is an option in superset cases, while complete-world or large-sample-size queries are amenable to less responsive but resource-expensive solutions.

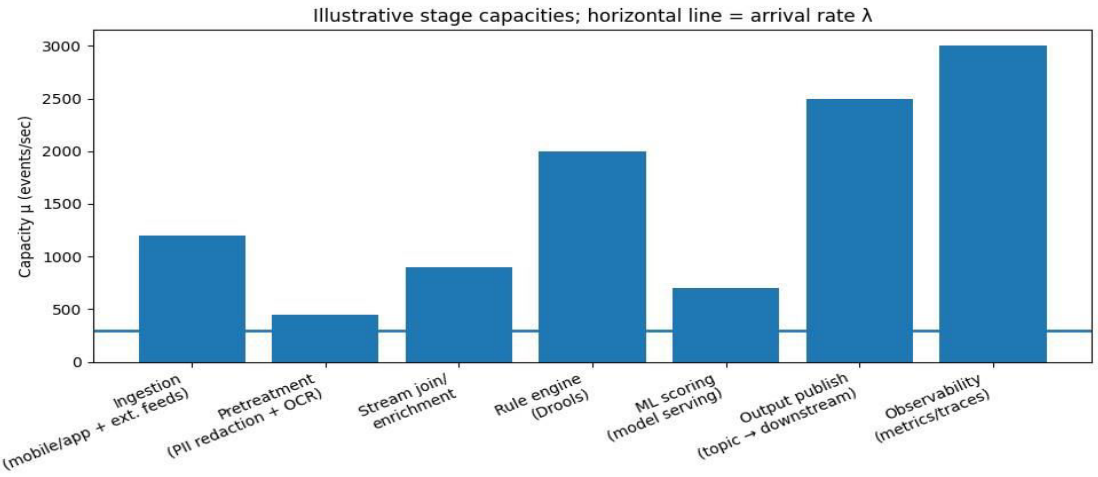
Data collection occurs at such scale and frequency that moving all events to a central location for evaluation is wasteful. An evaluation rule can be executed by application servers located closest to the events being scored, rather than by central resources. Natively, engine selection is driven by speed requirements, event volume (canizares2021polytopic), and engine availability. Where events must be sent back to the source or passing through nodes for additional side effects, throughput is paramount. The architecture allows all defined engines to cohabit.

IV. DATA QUALITY, GOVERNANCE, AND COMPLIANCE

Data quality, governance, and compliance underpin the accuracy and consistency of incoming and outgoing streams of data, as well as the correctness and inferences made during processing and evaluation. Effective claim processing architecture must ensure that data used during ingestion, pretreatment, and stream-processing phases meet expectations for quality, integrity, and security, and is accompanied by the necessary documentation to satisfy regulatory requirement. Labor-intensive data-labeling processes that are common in AI development must be combined with proper data governance practices in order to raise the quality and provenance of training data within reasonable cost.

Data Provenance and Lineage

AI/ML training data can arise from external sources, originate in the organization, or be produced as a by-product of other stream-processing components. Regardless of the data source, organizations must be able to produce metadata detailing the lineage of the data under governance, keeping track of how, when, and under what circumstances the data is produced. The scalable and cost-effective generation of training data remains a challenge, since non-expert users will not generate and curate high-quality nuisance data in sufficient quantity without incentive. Incentivizing other departments within a large organization to make a nuisance-data contribution is often difficult, bordering on impossible. As an alternative, the curriculum-learning approach leverages transparency in the input and output of stream-processing components to automatically generate undesired negative samples in combination with the desired positive samples whenever these are not sufficiently abundant or representative. Knowledge about the nature of the data can help prepare it for model training, but AI still requires a considerable amount of cataloguing, labeling and tagging by humans.



Equation B. Throughput and the bottleneck rule

4. Let stage i process at capacity μ_i events/sec
5. In a series pipeline, steady-state throughput is limited by the slowest stage:

$$T_{\max} = \min_i(\mu_i)$$

3. If arrival rate is λ, stability requires:

$$\lambda < \mu_i \quad \forall i$$

International Journal of Engineering & Extended Technologies Research (IJEETR)

|ISSN: 2322-0163| www.ijeetr.com | A Bimonthly, Peer Reviewed, Scholarly Indexed Journal |

| Volume 4, Issue 6, November-December 2022 |

DOI:10.15662/IJEETR.2022.0406013

4.1. Data Provenance and Lineage

Establishing data provenance and lineage builds trust in results generated through data pipelines and determines whether data is safe and reliable for specific use cases. Data provenance provides information about the origin of data elements that can be used for many useful purposes, such as assessing data freshness. Data lineage provides a complete and faithful view of the data lifecycle, tracking the data from point of origin to point of consumption while maintaining key details at each stage in the process. Maintenance of provenance and lineage information along the lifecycle of the data—covering extract, transform, and load (ETL) operations and all activities performed downstream on the data—is important for detecting malicious or unintended modification of the data, which can affect AI-generated decisions. Financial institutions are required to do this due diligence under the guidance of organizations such as the Basel Committee on Banking Supervision (BCBS). Other industries, such as healthcare, are starting to impose similar requirements.

Data quality check post ETL play a vital part in proving correct provenance and lineage, and many data quality frameworks have been developed. Companies adopting data mesh are also building data quality as product and deriving key data quality metrics from it, which can act as intermediate provenance or lineage information. These inform simpler and narrow ETL as multiple quality issues from disparate data sources are resolved before they are enriched or fused together. Tau-Client engineered a fine-grained automatic data provenance with low overheads. Built on cash register logs, it tracks data and control dependencies in queries to trace cash flow sources. Additionally, causal relation analysis provides a method to derive lineage rules and detect similarity.

4.2. Privacy, Security, and Regulatory Considerations

Privacy, security, and regulatory concerns must be carefully evaluated when designing the architecture of a real-time insurance claims processing pipeline. Data used for training AI/ML models, especially when dealing with PII or sensitive information, must be protected appropriately. Data security policies, access control mechanisms, and data anonymization techniques should be applied when necessary. Model training-related data may be modeled separately and incorporated into the real-time architecture only when required. The aforementioned considerations can also lead to a separation of the scoring model when deep learning is employed for scoring, considerably increasing performance. A reliable approach must be taken to ensure that the scoring pipeline processes requests in a timely manner. It must also meet the non-functional requirements for both scale and latency. As insurance claim scoring usually requires only a small amount of data, the risk of multiple models and different model versions executing in parallel is also small.

Insurance companies are required to comply with regulations that define the financial, technical, and operational standards to be applied in information processing. An example of such regulation is the requirement to follow the DNB guidelines defining the minimum requirements for the management of IT risk that apply in part III of the document. This defines important aspects for management, governance, and organizational design to control IT risk. In addition, the processing of data containing PII or information classified as sensitive must adhere to GDPR and the regulations imposed by other law enforcers. The potential impact of these regulations on insurance claim processing architecture varies depending on the organization, its profile, and its model.

The processing of insurance claims represents an operational activity in insurance organizations, and the construction of the related architecture must focus on its automatic application.

Equation C. Utilization, queueing delay, and "sub-second" design risk

If a stage is too "hot", queueing blows up and determinism suffers.

6. Define utilization of stage i:

$$\rho_i = \frac{\lambda}{\mu_i}$$

2. As $\rho_i \to 1$, waiting grows rapidly.
3. A common first-order approximation (M/M/1 queue) for expected **total** time at a stage:

$$E[W_i] = \frac{1}{\mu_i - \lambda}$$

and expected waiting-only time:

$$E[W_{q,i}] = \frac{\lambda}{\mu_i(\mu_i - \lambda)}$$

5. Then:

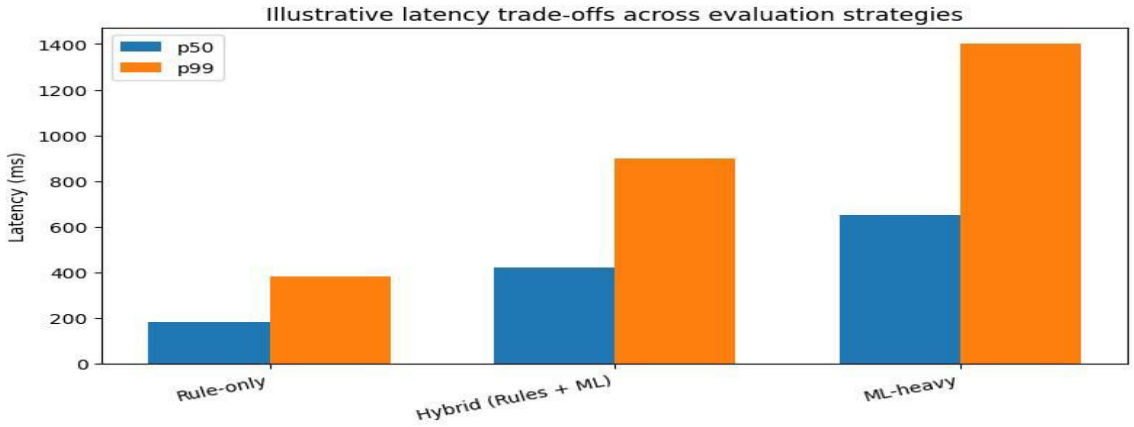
$$E[L_{\text{e2e}}] = \sum_i E[W_i]$$

International Journal of Engineering & Extended Technologies Research (IJEETR)

|ISSN: 2322-0163| www.ijeetr.com | A Bimonthly, Peer Reviewed, Scholarly Indexed Journal |

| Volume 4, Issue 6, November-December 2022 |

DOI:10.15662/IJEETR.2022.0406013



V. REAL-TIME CLAIM EVALUATION AND SCORING

Given the inherent characteristics of the above processing pattern, it becomes crucial to make the scoring phase of the claims pipeline as light and deterministic as possible. Therefore, concentrated efforts should be geared towards processing flow aspects such as:

• Building AI/ML models for scoring and tagging of claims that ensures minimum lag for low resource latency and minimal impact on Allocation/Genetic Knowledge Based Systems.

• Strategic integration of the AI/ML pipeline within the Claim Processing Pipeline - inline with the claim processing flow to minimize strain on Claim Processing Engineers (CPE) and ensure real-time latency thresholds are met.

• A clear understanding of the unique characteristics of a real-time business streams. While the processing latency window is significantly lower for Real-time streams when compared to ETL streams, the volume of the real-time streams could be significantly larger in most cases and hence it is essential to always consider relative latency numbers.

• Throughput patterns for Real-time processing streams also differ from ETL streams. The volume of incoming records dictates the resource requirements of the processing in order to meet established latency thresholds. Hence the throughput of the "following" systems that consume messages from these processing systems is also a key consideration.

• Balance between reducing latency & improving predictability. The focus on AI models and techniques should be in not only reducing latency but also ensuring that workloads are balanced across assigned resources.

• Ensure end-to-end predictability and determinism across Score and Stage processing flows. One of key factor of Risk Engines is predictability. Risk Engines have been designed to ensure that all scores across distinct claims arrive on a same time frame, thereby ensuring that (Allocation/Genetic KB Systems) are able to leverage these scores in identifying risk mitigation opportunities with minimal effort.

Stage	Mean processing latency (ms)	Capacity μ (events/sec)	Utilization ρ=λ/μ
Ingestion (mobile/app + ext. feeds)	45.0	1200.0	0.25
Pretreatment (PII redaction + OCR)	220.0	450.0	0.6666666666666666
Stream join/ enrichment	80.0	900.0	0.3333333333333333
Rule engine (Drools)	35.0	2000.0	0.15

5.1. Rule-Based and AI/ML Integration

In many insurance companies, constraints on the real-time evaluation of claims come from the scores assigned to each claim, based on criteria such as fraud risk, appropriateness of coverage, etc. These are usually arrived at through a combination of experience, statistical analyses of historical claims, and regulatory considerations, and codified in the organization's Claim Evaluation Guide. The analysis determines whether standard rules are sufficient for evaluation or if AI/ML techniques need to be deployed. Even though AI/ML techniques offer the promise of improved accuracy, there are numerous complications in their deployment. If standard rules are judged sufficient for evaluation, this can be accomplished with minimal latency, freeing the processing capacity to handle additional claims or cases of greater severity.

Until recently, rule-based evaluation followed a traditional if-then style employed by enterprise rule engines. More recently, the Drools engine has been deployed for this purpose, taking query-like input and providing ruleset execution

International Journal of Engineering & Extended Technologies Research (IJEETR)

|ISSN: 2322-0163| www.ijeetr.com | A Bimonthly, Peer Reviewed, Scholarly Indexed Journal |

| Volume 4, Issue 6, November-December 2022 |

DOI:10.15662/IJEETR.2022.0406013

as a service, thus improving maintainability, simplifying modeling and decreasing cost. There is considerable literature on the suitability of AI/ML techniques for this problem domain and on the many confounding factors in deploying AI/ML models for this purpose. For fraud detection models in particular, the typical high class imbalance requires techniques such as clustering together cases to form balanced mini-datasets, specific resampling approaches, or the direct use of synthetic datasets like SMOTE, ADASYN, etc. Many insurance companies develop their own AI/ML models for fraud detection or join forces in consortia to leverage shared experience.

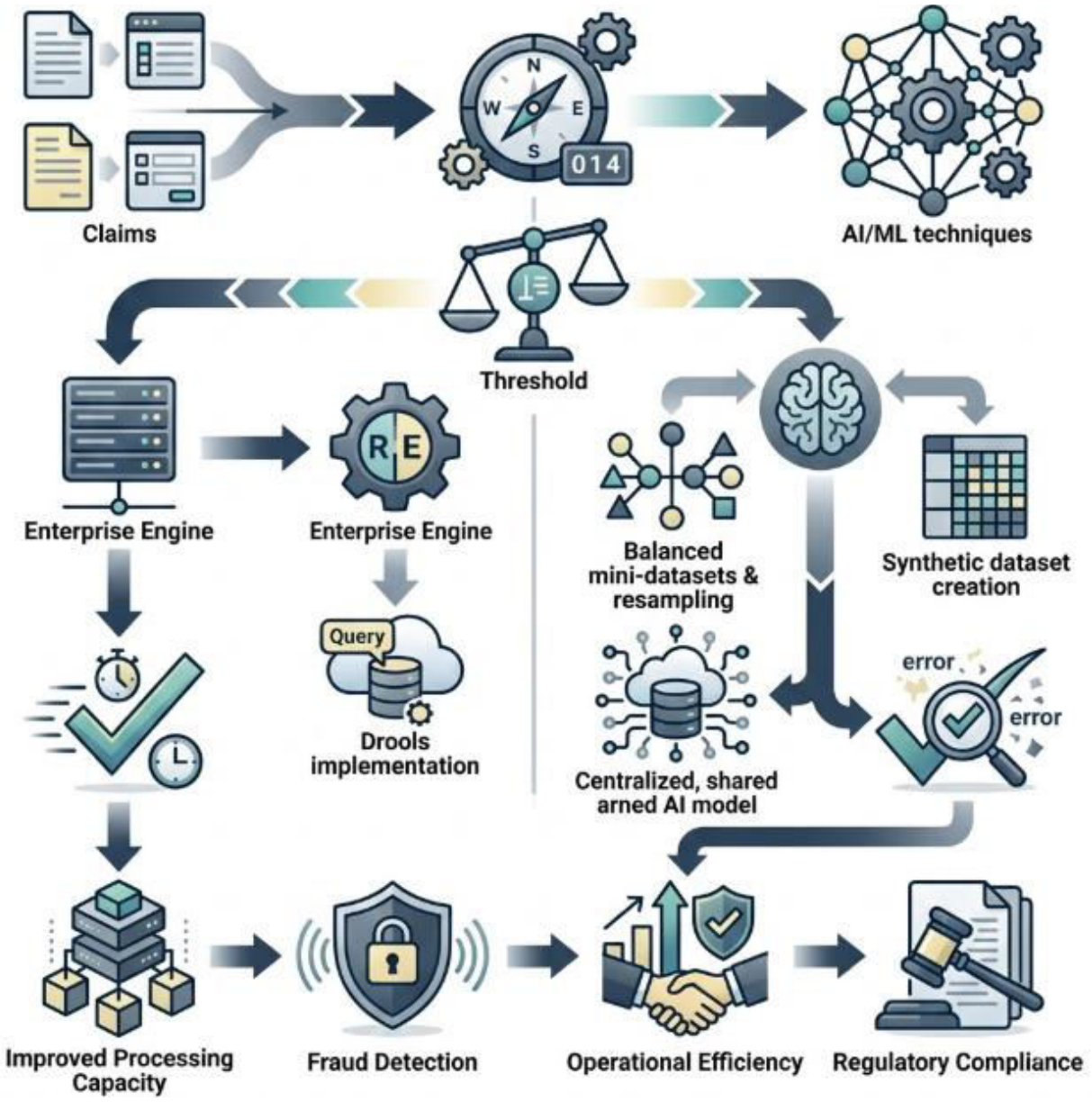


Fig 3: Hybrid Intelligence in Insurance Claims Evaluation: Assessing the Impact of Modern Rule Engines and AI Deployment on Operational Latency

5.2. Latency, Throughput, and Determinism in Scoring

In addition to rule-based evaluation, distributed data pipelines can also support the activation of AI/ML-based models, retraining of models, and detection of anomalies to report suspicious claims. A major challenge in building the pipelines is achieving low latency, high throughput, and deterministic behavior. In many scenarios, grouping claims by region, or another grouping logic, will allow for increased throughput while not affecting throughput. Once grouped, the activation and scoring of AI/ML models is a key challenge. Due to the requirement of activating and scoring any

International Journal of Engineering & Extended Technologies Research (IJEETR)

|ISSN: 2322-0163| www.ijeetr.com | A Bimonthly, Peer Reviewed, Scholarly Indexed Journal |

| Volume 4, Issue 6, November-December 2022 |

DOI:10.15662/IJEETR.2022.0406013

model with low latency, low-latency, high-throughput, non-deterministic systems may not be viable; model serving solutions supporting pre-loading of all necessary models are a consideration. With the support of an appropriate serving solution, the triggering of model retraining and anomaly detection may also be addressed. The creation of a re-training and anomaly-detection solution, combined with support for a unified fraud detection scoring system, will provide an even higher degree of confidence of tackling the problem.

Equation D. Determinism (predictability) as a measurable objective

A practical way to *quantify* determinism is with tail-latency and jitter:

1. Let L be the latency random variable.
2. Tail latency (e.g., p99):

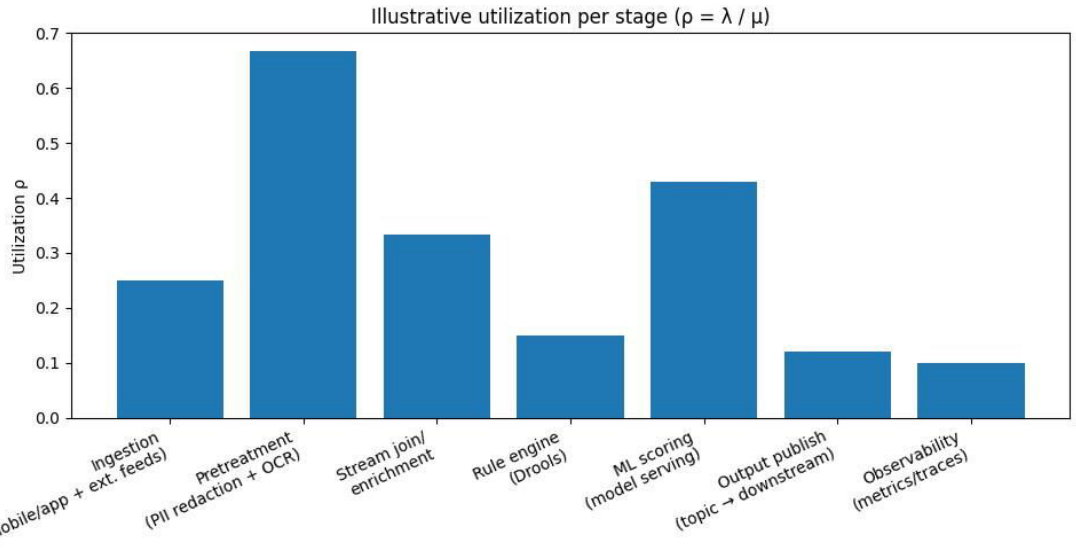
$$L_{p99} = \text{Quantile}_{0.99}(L)$$

3. Jitter (variability) often measured by standard deviation:

$$\sigma_L = \sqrt{E[L^2] - (E[L])^2}$$

5. A "determinism target" can be written as:

$$L_{p99} \leq \text{SLA}_{p99} \quad \text{and} \quad \sigma_L \leq \sigma_{\max}$$



VI. OPERATIONALIZING DISTRIBUTED PIPELINES

The entire approach described thus far leads to the construction of distributed data pipelines designed to enable real-time insurance claim evaluation. Distributing the claim-processing architecture over multiple pipelines brings several advantages. The different data sources supporting claim processing can be covered with independent ingestion pipelines, each handling the volume and variety best suited to their data source. Likewise, the generation of claim-supporting context data from other operational data (e.g., a weather pipeline) can also be served by independent context-pipeline. Such an architecture is logically aligned with a microservice architecture. Claim evaluation also lends itself to being distributed over multiple stages and/or microservices. Each claim can be processed independently. Rule-based scoring is typically not latency-critical, and scoring could also be performed on a best-effort basis. These properties extend to AI/ML scoring, especially when integrated in a cascade. Periodically updating fetching of model metadata from blob storage also lowers overhead. All of these allow the use of lighter-weight, developer-friendly open-source stream processing frameworks (like Faust) for processing at scale

While these distribution elements allow the pipelines to be scaled appropriately for volume and latency, they also add operational overhead. Deployment of several pipelines (ingestion, context generation, evaluation, etc.) adds deployment complexity, and these need to be developer-tested one-to-one for deployment success. Containerization simplifies deployment by packaging all dependencies. Typical industrial setups use observability and monitoring to proactively catch and fix issues before they affect the running pipeline. Observability focuses on service definition: what the service does, how it is used, what data it generates/consumes, and its use pre-condition. Monitoring automatically

International Journal of Engineering & Extended Technologies Research (IJEETR)

|ISSN: 2322-0163| www.ijeetr.com | A Bimonthly, Peer Reviewed, Scholarly Indexed Journal |

| Volume 4, Issue 6, November-December 2022 |

DOI:10.15662/IJEETR.2022.0406013

checks if a deployed service is behaving as expected. Monitoring signals deviations from expected behavior, raising alerts that trigger debugging and/or remedial actions.

Approach	p50 latency (ms)	p99 latency (ms)	Fraud recall (TPR)
Rule-only	180.0	380.0	0.55
Hybrid (Rules + ML)	420.0	900.0	0.78
ML-heavy	650.0	1400.0	0.82

6.1. Deployment Strategies and Containerization

A review of cloud-native architectural blueprints presented in Section 4 of Part 3 shows that the control and data availability characteristics of the real-time requirement allow for a dynamically scalable hybrid, in which the specific pipelines may execute processing and integration algorithms on a parallel cluster and material stations, such as the database and artificial intelligence (AI)/machine learning (ML) serving engine, may be collected on dedicated scaling groups. The use clusters should preferably be deployed in the data and control cloud availability zone, while material cloud resources should be reserved for configuration and testing management.

Microservices approaches are being extended to data engineering processes. With the ability to define any part of a process as a microservice in general and deploy and interconnect them dynamically in particular, emerging and trailing parts of a data flow dynamically scale according to the specific cloud operation and data availability zone blueprint pattern, improving hardware resource utilization and economizing related costs. The observation that a data engineering pipeline is often composed of several sequential and parallel processes, by nature often acting in the same DC-ASP, lends itself to the development of a microservices orchestration process that defines and produces the complete orchestration specification. Kubernetes and its ecosystems leverage container-based virtualization to enable the definition of any process and data transformation, configuration, and processing microservices as containers. Moreover, containers have become the key technology for implementing and enabling dynamic service provision in cloud environments.

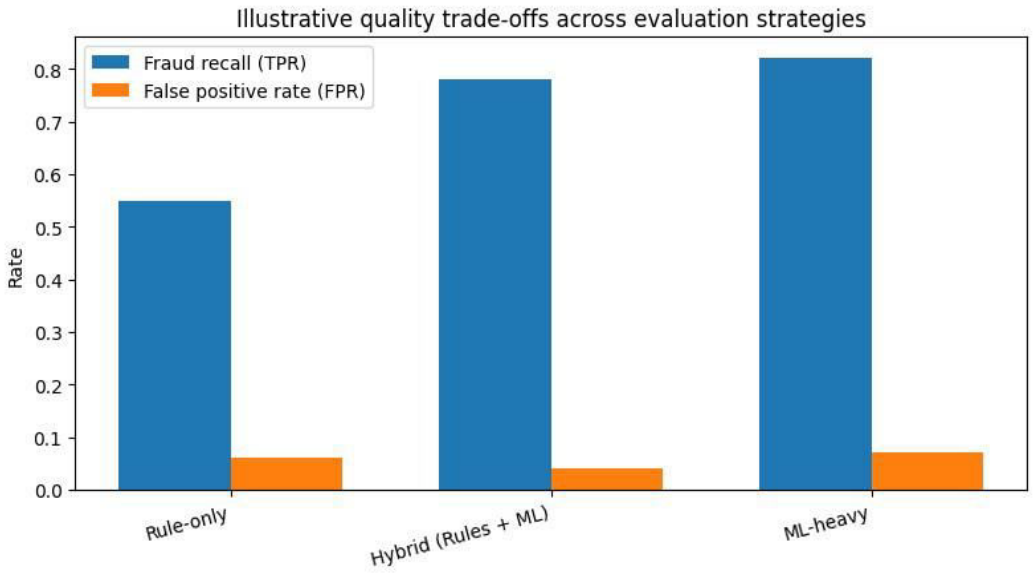
Equation E. Rule + ML hybrid scoring and threshold decisioning

1. Let ML output a risk score $s \in [0,1]$ interpreted as fraud propensity.
2. Decision threshold τ:

$$\text{Decision}(s) = \begin{cases} \text{Auto-approve} & s < \tau \\ \text{Manual review / investigate} & s \geq \tau \end{cases}$$

3. If you have a cost of false negative C_{FN} (missed fraud) and false positive C_{FP} (unnecessary investigation), you choose τ to minimize expected cost:

$$\min_{\tau} C_{FN} \cdot FN(\tau) + C_{FP} \cdot FP(\tau)$$



International Journal of Engineering & Extended Technologies Research (IJEETR)

|ISSN: 2322-0163| www.ijeetr.com | A Bimonthly, Peer Reviewed, Scholarly Indexed Journal |

| Volume 4, Issue 6, November-December 2022 |

DOI:10.15662/IJEETR.2022.0406013

6.2. Observability, Monitoring, and alerting

The implementation of distributed streaming systems introduces a new focus on observability and monitoring. Systems often become visible only when operational issues arise, which raises the difficulty to manage complex distributed architectures. Using a toolset capable of collecting large amounts of telemetry data for distributed streaming systems is essential to understand which latency and throughput level are acceptable and to find the trade-off between these two objectives. Such a toolset should be able to inform the right instant to scale the system in or out. Abnormal patterns on metrics collected at individual components or aggregated through the system (e.g., the 99th percentile of the latency for a rule-based pipeline) can indicate anomalies in the processing and entail automatic alerts, supported by human expertise or machine learning.

A development workflow that allows coding and testing of the business logic in knowledge files outside the production environment is recommended in detriment of a development and test pipeline that introduces a slower deployment cycle for the business logic. A "knowledge file" is an artifact that may be created, modified, and tested by domain specialists without requiring knowledge of the overall architecture or underlying coding/programming skills. The creation and update of knowledge files can be automatically deployed without interfering with the specialized knowledge of the domain expert.

VII. CONCLUSION

During the investigation prompted by the need for a business unit to obtain a claim evaluation from a customer using a streaming front-end (e.g., chat), it became clear that the creation of a distributed data engineering pipeline would enable automated notifications and a dashboard that visualize the current status of claims, identifying those that do not comply with the conditions.

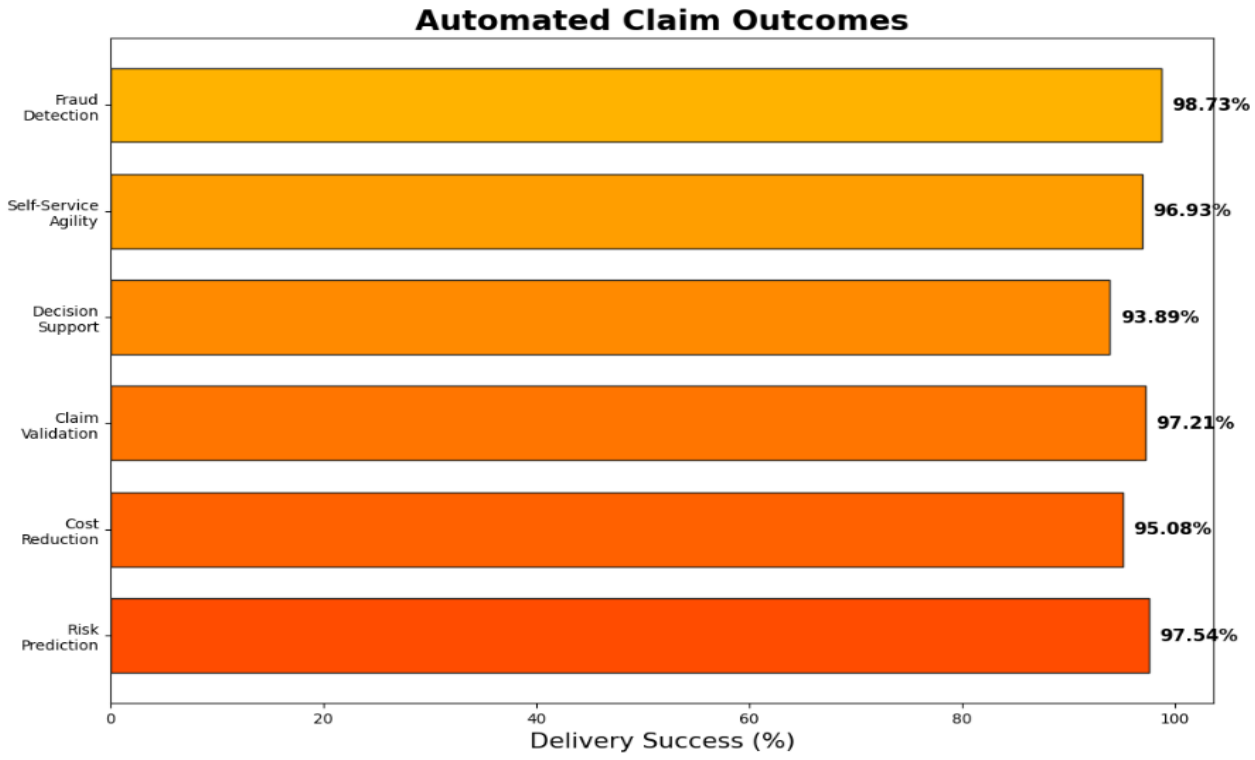


Fig 4: Automated Claim Outcomes

The distributed pipeline also allows for the real-time scoring of claims using traditional rules and AI/ML, thereby offering risk and cost prediction and proposing rough reservations to trigger a human evaluation. The combination of distributed data engineering, data quality, governance, compliance, and operationalization open the way for self-service solutions that make the necessary information available for decision-making.

|ISSN: 2322-0163| www.ijeetr.com | A Bimonthly, Peer Reviewed, Scholarly Indexed Journal |

| Volume 4, Issue 6, November-December 2022 |

DOI:10.15662/IJEETR.2022.0406013

Distributed data engineering pipelines operating on business event streams are relevant for modern services that require the creation of real-time dashboards, intelligent notifications, and the scoring of processes or entities (e.g., customers, suppliers, claims, contracts) embedded in services to predict risk or to reduce costs associated with manual evaluation or processing. Data quality and governance are essential ingredients to guarantee that the information is trusted and secure, while observability, monitoring, and alerting ensure that the system operates as expected.

7.1. Final Thoughts and Future Directions

The combination of different pipelines will allow to support a wide range of additional use cases, such as demand forecasting for insurance products, a new claim damage detection model which will rank and score claims, and a telemetry pipeline taking dispatch logs from a telematics company, converting them into a uniform time-series format, allowing for the dispatch behaviour to be combined with the insurance-fraud and other scoring outputs.

As parts of these pipelines are deployed into production, it is crucial not only to continuously monitor those already in operation, but also to create a systematic approach for start-up of newly implemented services. Containerization of all pipeline components has proven to be an efficient way to deliver new code across different environments (e.g. from development/test to QA and production) without producing configuration incosistencies across them.

Using container orchestration platforms like Kubernetes allows to define horizontal deployment strategies, reducing maintenance efforts, allowing for more effective use of public cloud resources, and improving overall robustness of the system. Adding Helm charts to the deployment framework allows for development and testing of additional components in a defined way before moving them into production. Those deployments allow to future-proof the real-time pipelines with production-quality observability, monitoring and alerting facilities, and – when testing the possible integration of complex machine learning or AI solutions – for real-time decisioning of simpler, rule-based solutions in order to significantly reduce average claim evaluation times whilst keeping detection performance similar to evaluations performed by human teams.

REFERENCES

[1] Davuluri, P. N. Event-Driven Compliance Systems: Modernizing Financial Crime Detection Without Machine Intelligence.

[2] Goutham Kumar Sheelam, "Semiconductor Innovation for Edge AI: Enabling Ultra-Low Latency in Next-Gen Wireless Networks," International Journal of Advanced Research in Computer and Communication Engineering (IJARCCE), DOI: 10.17148/IJARCCE.2022.111258

[3] Andry, J. F., Hartono, H., & Jo, J. Analysis and prediction of supermarket sales with data mining using RapidMiner. AIP Conference Proceedings, 2693(1). https://doi.org/10.1063/5.0118725.

[4] Davuluri, P. N. (2020). Improving Data Quality and Lineage in Regulated Financial Data Platforms. Finance and Economics, 1(1), 1-14.

[5] Li, H., Wei, H., Zhao, W., & Zheng, X. Research on geographic information data circulation supports the construction of digital China. The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences, XLVIII-1/W2-, 97–104.

[6] Annapareddy, V. N., Preethish Nandan, B., Kommaragiri, V. B., Gadi, A. L., & Kalisetty, S. (2022). Emerging Technologies in Smart Computing, Sustainable Energy, and Next-Generation Mobility: Enhancing Digital Infrastructure, Secure Networks, and Intelligent Manufacturing.

[7] Armbrust, M., Das, T., Davidson, A., Ghodsi, A., Or, A., Rosen, J., Stoica, I., Wendell, P., Xin, R., & Zaharia, M. (2021). Delta Lake: High-performance ACID table storage over cloud object stores. Proceedings of the VLDB Endowment, 13(12), 3411–3424.

[8] Avinash Reddy Segireddy. (2022). Terraform and Ansible in Building Resilient Cloud-Native Payment Architectures. International Journal of Intelligent Systems and Applications in Engineering, 10(3s), 444–455. Retrieved from https://www.ijisae.org/index.php/IJISAE/article/view/7905

[9] Kotlinski, M., & Calkowska, J. K. (2022). U-space and UTM deployment as an opportunity for more complex UAV operations including UAV medical transport. Journal of Intelligent & Robotic Systems, 106, 12. https://doi.org/10.1007/s10846-022-01681-6

[10] Chava, K., Chakilam, C., & Recharla, M. (2021). Machine Learning Models for Early Disease Detection: A Big Data Approach to Personalized Healthcare. International Journal of Engineering and Computer Science, 10(12), 25709–25730. https://doi.org/10.18535/ijecs.v10i12.4678

|ISSN: 2322-0163| www.ijeetr.com | A Bimonthly, Peer Reviewed, Scholarly Indexed Journal |

| Volume 4, Issue 6, November-December 2022 |

DOI:10.15662/IJEETR.2022.0406013

[11] Kalisetty, S., Vankayalapati, R. K., Reddy, L., Sondinti, K., & Valiki, S. (2022). AI-Native Cloud Platforms: Redefining Scalability and Flexibility in Artificial Intelligence Workflows. Linguistic and Philosophical Investigations, 21(1), 1-15.
[12] Sriram, H. K. (2022). Advancements in Credit Score Analytics using Deep Learning and Predictive Modeling Techniques. Available at SSRN 5255128.
[13] Bifet, A., & Gavaldà, R. (2007). Learning from time-changing data with adaptive windowing. Proceedings of the 2007 SIAM International Conference on Data Mining, 443–448.
[14] Muthusamy, S., Kannan, S., Lee, M., Sanjairaj, V., Lu, W. F., Fuh, J. Y., ... & Cao, T. (2021). Cover Image, Volume 118, Number 8, August 2021. Biotechnology and Bioengineering, 118(8), i-i.
[15] Gondhi, P. K. FinTech cloud-based data lakes: Performance, governance, and scalability. Journal of Computer Science and Technology Studies, 7(2), 1–12.
[16] Vadisetty, R., Polamarasetti, A., Guntupalli, R., Raghunath, V., Jyothi, V. K., & Kudithipudi, K. (2021). Privacy-Preserving Gen AI in Multi-Tenant Cloud Environments. Sateesh kumar and Raghunath, Vedaprada and Jyothi, Vinaya Kumar and Kudithipudi, Karthik, Privacy-Preserving Gen AI in Multi-Tenant Cloud Environments (January 20, 2021).
[17] Chen, M., Mao, S., & Liu, Y. (2014). Big data: A survey. Mobile Networks and Applications, 19(2), 171–209.
[18] Dwaraka Nath Kummari. (2022). Fiscal Policy Simulation Using AI And Big Data: Improving Government Financial Planning. Kurdish Studies, 10(2), 934–945. https://doi.org/10.53555/ks.v10i2.3855
[19] Chen, T., & Guestrin, C. (2016). XGBoost: A scalable tree boosting system. Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 785–794.
[20] Gadi, A. L. The Role of Digital Twins in Automotive R&D for Rapid Prototyping and System Integration.
[21] Das, T., Zhu, A., Li, S., Narayanamurthy, S., & Bhat, P. (2013). Distributed and fault-tolerant streaming computation in Spark. Proceedings of the ACM Symposium on Cloud Computing, 1–12.
[22] Siva Hemanth Kolla. (2022). Knowledge Retrieval Systems for Enterprise Service Environments. International Journal of Intelligent Systems and Applications in Engineering, 10(3s), 495–506. Retrieved from https://ijisae.org/index.php/IJISAE/article/view/8037
[23] Dean, J., & Ghemawat, S. (2008). MapReduce: Simplified data processing on large clusters. Communications of the ACM, 51(1), 107–113.
[24] Paleti, S. (2022). Financial Innovation through AI and Data Engineering: Rethinking Risk and Compliance in the Banking Industry. Available at SSRN 5250726.
[25] DeCandia, G., Hastorun, D., Jampani, M., Kakulapati, G., Lakshman, A., Pilchin, A., Sivasubramanian, S., Vosshall, P., & Vogels, W. (2007). Dynamo: Amazon's highly available key-value store. Proceedings of the 21st ACM Symposium on Operating Systems Principles, 205–220.
[26] Sriram, H. K., ADUSUPALLI, B., & Malempati, M. (2021). Revolutionizing Risk Assessment and Financial Ecosystems with Smart Automation, Secure Digital Solutions, and Advanced Analytical Frameworks.
[27] Dwork, C. (2008). Differential privacy: A survey of results. Proceedings of the 5th International Conference on Theory and Applications of Models of Computation, 1–19.
[28] Paleti, S., Singireddy, J., Dodda, A., Burugulla, J. K. R., & Challa, K. (2021). Innovative financial technologies: Strengthening compliance, secure transactions, and intelligent advisory systems through ai-driven automation and scalable data architectures.
[29] Elmagarmid, A. K., Ipeirotis, P. G., & Verykios, V. S. (2007). Duplicate record detection: A survey. IEEE Transactions on Knowledge and Data Engineering, 19(1), 1–16.
[30] Dwaraka Nath Kummari,. (2022). Machine Learning Approaches to Real-Time Quality Control in Automotive Assembly Lines. Mathematical Statistician and Engineering Applications, 71(4), 16801–16820. Retrieved from https://philstat.org/index.php/MSEA/article/view/2972
[31] Fader, P. S., Hardie, B. G. S., & Lee, K. L. (2005). "Counting your customers" the easy way: An alternative to the Pareto/NBD model. Marketing Science, 24(2), 275–284.
[32] Inala, R. (2022). Engineering Data Products for Investment Analytics: The Role of Product Master Data and Scalable Big Data Solutions. International Journal of Scientific Research and Modern Technology, 155-171.
[33] Davuluri, P. N. (2020). Improving Data Quality and Lineage in Regulated Financial Data Platforms. Finance and Economics, 1(1), 1-14.
[34] Kalisetty, S., & Ganti, V. K. A. T. (2019). Transforming the Retail Landscape: Srinivas's Vision for Integrating Advanced Technologies in Supply Chain Efficiency and Customer Experience. Online Journal of Materials Science, 1, 1254.
[35] Ghemawat, S., Gobioff, H., & Leung, S. T. (2003). The Google file system. Proceedings of the 19th ACM Symposium on Operating Systems Principles, 29–43.

|ISSN: 2322-0163| www.ijeetr.com | A Bimonthly, Peer Reviewed, Scholarly Indexed Journal |

| Volume 4, Issue 6, November-December 2022 |

DOI:10.15662/IJEETR.2022.0406013

[36] Singireddy, J. (2022). Leveraging Artificial Intelligence and Machine Learning for Enhancing Automated Financial Advisory Systems: A Study on AIDriven Personalized Financial Planning and Credit Monitoring. Mathematical Statistician and Engineering Applications, 71 (4), 16711–16728.
[37] Yandamuri, U. S. (2021). A Comparative Study of Traditional Reporting Systems versus Real-Time Analytics Dashboards in Enterprise Operations. Universal Journal of Business and Management, 1(1), 1–13. Retrieved from https://www.scipublications.com/journal/index.php/ujbm/article/view/1357
[38] Kolla, S. K. (2021). Architectural Frameworks for Large-Scale Electronic Health Record Data Platforms. Current Research in Public Health, 1(1), 1–19. Retrieved from https://www.scipublications.com/journal/index.php/crph/article/view/1372.
[39] Hastie, T., Tibshirani, R., & Friedman, J. (2009). The elements of statistical learning: Data mining, inference, and prediction (2nd ed.). Springer.
[40] Vadisetty, R., Polamarasetti, A., Guntupalli, R., Raghunath, V., Jyothi, V. K., & Kudithipudi, K. (2022). AI-Driven Cybersecurity: Enhancing Cloud Security with Machine Learning and AI Agents. Sateesh kumar and Raghunath, Vedaprada and Jyothi, Vinaya Kumar and Kudithipudi, Karthik, AI-Driven Cybersecurity: Enhancing Cloud Security with Machine Learning and AI Agents (February 07, 2022).
[41] Hellerstein, J. M., Haas, P. J., & Wang, H. J. (1997). Online aggregation. Proceedings of the 1997 ACM SIGMOD International Conference on Management of Data, 171–182.
[42] Garapati, R. S. (2022). Web-Centric Cloud Framework for Real-Time Monitoring and Risk Prediction in Clinical Trials Using Machine Learning. Current Research in Public Health, 2, 1346.
[43] Hu, Y., Koren, Y., & Volinsky, C. (2008). Collaborative filtering for implicit feedback datasets. Proceedings of the 2008 IEEE International Conference on Data Mining, 263–272.
[44] Amistapuram, K. (2022). Fraud Detection and Risk Modeling in Insurance: Early Adoption of Machine Learning in Claims Processing. Available at SSRN 5741982.
[45] Davuluri, P. S. L. N. (2021). Event-Driven Compliance Systems: Modernizing Financial Crime Detection Without Machine Intelligence. Journal of International Crisis and Risk Communication Research , 339–354. https://doi.org/10.63278/jicrcr.vi.3636
[46] Meda, R. (2022). Integrating Edge AI in Smart Factories: A Case Study from the Paint Manufacturing Industry. International Journal of Science and Research (IJSR), 1473-1489.
[47] Jagadish, H. V., Gehrke, J., Labrinidis, A., Papakonstantinou, Y., Patel, J. M., Ramakrishnan, R., & Shahabi, C. (2014). Big data and its technical challenges. Communications of the ACM, 57(7), 86–94.
[48] Segireddy, A. R. (2020). Cloud Migration Strategies for High-Volume Financial Messaging Systems.
[49] Meda, R. Enabling Sustainable Manufacturing Through AI-Optimized Supply Chains.
[50] Amistapuram, K. (2021). Digital Transformation in Insurance: Migrating Enterprise Policy Systems to .NET Core. Universal Journal of Computer Sciences and Communications, 1(1), 1–17.
[51] Kleppmann, M. (2017). Designing data-intensive applications. O'Reilly Media.
[52] Nagabhyru, K. C. (2022). Bridging Traditional ETL Pipelines with AI Enhanced Data Workflows: Foundations of Intelligent Automation in Data Engineering. Available at SSRN 5505199.
[53] Lahiri, M., & Venkatasubramanian, S. (2013). Robust record linkage. Proceedings of the 2013 ACM SIGMOD International Conference on Management of Data, 101–112.
[54] Aitha, A. R. (2022). Cloud Native ETL Pipelines for Real Time Claims Processing in Large Scale Insurers. Available at SSRN 5532601.
[55] Leskovec, J., Rajaraman, A., & Ullman, J. D. (2014). Mining of massive datasets (2nd ed.). Cambridge University Press.
[56] Adusupalli, B., Pandiri, L., & Singireddy, S. (2019). DevOps Enablement in Legacy Insurance Infrastructure for Agile Policy and Claims Deployment. risk, 7(12).
[57] Linden, G., Smith, B., & York, J. (2003). Amazon.com recommendations: Item-to-item collaborative filtering. IEEE Internet Computing, 7(1), 76–80.
[58] Choudhary, V., Kartik, & Bala, N. Cloud-based data lake. International Conference on Artificial Intelligence and Quantum Computation-Based Sensor Application (ICAIQSA), 1–5.
[59] Lin, J., Kolcz, A., & Szymanski, B. K. (2012). Large-scale machine learning at Twitter. Proceedings of the 2012 ACM SIGMOD International Conference on Management of Data, 793–804.
[60] Nandan, B. P. (2022). AI-Powered Fault Detection In Semiconductor Fabrication: A Data-Centric Perspective.
[61] Manyika, J., Chui, M., Brown, B., Bughin, J., Dobbs, R., Roxburgh, C., & Byers, A. H. (2011). Big data: The next frontier for innovation, competition, and productivity. McKinsey Global Institute.
[62] Vadisetty, R., Polamarasetti, A., Guntupalli, R., Rongali, S. K., Raghunath, V., Jyothi, V. K., & Kudithipudi, K. (2021). Legal and Ethical Considerations for Hosting GenAI on the Cloud. International Journal of AI, BigData, Computational and Management Studies, 2(2), 28-34.

|ISSN: 2322-0163| www.ijeetr.com | A Bimonthly, Peer Reviewed, Scholarly Indexed Journal |

| Volume 4, Issue 6, November-December 2022 |

DOI:10.15662/IJEETR.2022.0406013

[63] Mikolov, T., Chen, K., Corrado, G., & Dean, J. (2013). Efficient estimation of word representations in vector space. Proceedings of the International Conference on Learning Representations, 1–12.
[64] Adusupalli, B., Singireddy, S., Sriram, H. K., Kaulwar, P. K., & Malempati, M. (2021). Revolutionizing Risk Assessment and Financial Ecosystems with Smart Automation, Secure Digital Solutions, and Advanced Analytical Frameworks. Universal Journal of Finance and Economics, 1(1), 101-122.
[65] Montoya, D. Y., Neto, A. M., & da Silva, A. S. (2016). A survey of entity resolution in big data. Journal of Big Data, 3(1), 1–22.
[66] Aitha, A. R. (2021). Optimizing Data Warehousing for Large Scale Policy Management Using Advanced ETL Frameworks.
[67] Zaharia, M., Chowdhury, M., Franklin, M. J., Shenker, S., & Stoica, I. (2010). Spark: Cluster computing with working sets. Proceedings of the 2nd USENIX Conference on Hot Topics in Cloud Computing, 1–7.
[68] Challa, K. (2021). Cloud Native Architecture for Scalable Fintech Applications with Real Time Payments. International Journal Of Engineering And Computer Science, 10(12).
[69] Meda, R. (2021). Digital Infrastructure for Predictive Inventory Management in Retail Using Machine Learning. International Journal of Advanced Research in Computer and Communication Engineering (IJARCCE), DOI, 10.
[70] Segireddy, A. R. (2021). Containerization and Microservices in Payment Systems: A Study of Kubernetes and Docker in Financial Applications. Universal Journal of Business and Management, 1(1), 1–17.
[71] Zhai, C., & Massung, S. (2016). Text data management and analysis: A practical introduction to information retrieval and text mining. ACM & Morgan Claypool.
[72] Davuluri, P. N. (2020). Event-Driven Architectures for Real-Time Regulatory Monitoring in Global Banking.
[73] Bojanowski, P., Grave, E., Joulin, A., & Mikolov, T. (2017). Enriching word vectors with subword information. Transactions of the Association for Computational Linguistics, 5, 135–146.
[74] Keerthi Amistapuram , "Energy-Efficient System Design for High-Volume Insurance Applications in Cloud-Native Environments," International Journal of Innovative Research in Electrical, Electronics, Instrumentation and Control Engineering (IJIREEICE), DOI 10.17148/IJIREEICE.2020.81209
[75] Kannan, S. (2021). Advanced Computational Technologies in Vehicle Production, Digital Connectivity, and Sustainable Transportation: Innovations in Intelligent Systems, Eco-Friendly Manufacturing, and Financial Optimization. Universal Journal of Finance and Economics.
[76] Kothapalli Sondinti, L. R., & Syed, S. (2022). The Impact of Instant Credit Card Issuance and Personalized Financial Solutions on Enhancing Customer Experience in the Digital Banking Era. Universal Journal of Finance and Economics, 1(1), 1223. Retrieved from https://www.scipublications.com/journal/index.php/ujfe/article/view/1223
[77] Gottimukkala, V. R. R. (2021). Digital Signal Processing Challenges in Financial Messaging Systems: Case Studies in High-Volume SWIFT Flows.
[78] Bhasin, H., & Bhatia, P. (2020). Clickstream data mining for web analytics and customer behavior modeling: A review. ACM Computing Surveys, 53(6), 1–34.
[79] Kolla, S. H. (2021). Rule-Based Automation for IT Service Management Workflows. Online Journal of Engineering Sciences, 1(1), 1–14. Retrieved from https://www.scipublications.com/journal/index.php/ojes/article/view/1360
[80] Gottimukkala, V. R. R. (2022). Licensing Innovation in the Financial Messaging Ecosystem: Business Models and Global Compliance Impact. International Journal of Scientific Research and Modern Technology, 1(12), 177-186
[81] Abedjan, Z., Golab, L., & Naumann, F. (2016). Profiling relational data: A survey. The VLDB Journal, 24(4), 557–581.
[82] Yandamuri, U. S. (2022). Big Data Pipelines for Cross-Domain Decision Support: A Cloud-Centric Approach. International Journal of Scientific Research and Modern Technology, 1(12), 227–237. https://doi.org/10.38124/ijsrmt.v1i12.1111
[83] Dwaraka Nath Kummari. (2022). AI-Driven Audit Frameworks For Enhancing Compliance In Modern Manufacturing Systems. Migration Letters, 19(S8), 2150–2177. Retrieved from https://migrationletters.com/index.php/ml/article/view/11912
[84] Goutham Kumar Sheelam, "Semiconductor Innovation for Edge AI: Enabling Ultra-Low Latency in Next-Gen Wireless Networks," International Journal of Advanced Research in Computer and Communication Engineering (IJARCCE), DOI: 10.17148/IJARCCE.2022.111258.
[85] Baesens, B., Van Vlasselaer, V., & Verbeke, W. (2021). Fraud analytics using descriptive, predictive, and social network techniques: A guide to data science for fraud detection (2nd ed.). Wiley.
[86] Challa, K. (2022). The Future of Cashless Economies Through Big Data Analytics in Payment Systems. International Journal of Scientific Research and Modern Technology, 60-70.

|ISSN: 2322-0163| www.ijeetr.com | A Bimonthly, Peer Reviewed, Scholarly Indexed Journal |

| Volume 4, Issue 6, November-December 2022 |

DOI:10.15662/IJEETR.2022.0406013

[87] Buccella, A., Cechich, A., Saurin, F., Montenegro, A., Rodríguez, A., & Muñoz, A. A context-based perspective on frost analysis in reuse-oriented big data-system developments. Information, 15(11), 661. https://doi.org/10.3390/info15110661

[88] Garapati, R. S. (2022). AI-Augmented Virtual Health Assistant: A Web-Based Solution for Personalized Medication Management and Patient Engagement. Available at SSRN 5639650.

[89] Gottimukkala, V. R. R. (2020). Energy-Efficient Design Patterns for Large-Scale Banking Applications Deployed on AWS Cloud. power, 9(12).

[90] Rosário, A. T., & Raimundo, R. Internet of Things and Distributed Computing Systems in Business Models. Future Internet, 16(10), 384. https://doi.org/10.3390/fi16100384.

[91] Almeida, A., Brás, S., Sargento, S., & Pinto, F. C. Time series big data: a survey on data stream frameworks, analysis and algorithms. Journal of Big Data, 10(1). https://doi.org/10.1186/s40537-023-00760-1.

[92] Juwita, J., Safii, M., & Damanik, B. E. (2022). Naïve Bayes algorithm for predicting sales at the Pematang Siantar VJCakes store. JOMLAI: Journal of Machine Learning and Artificial Intelligence, 1(4). https://doi.org/10.55123/jomlai.v1i4.1674

[93] Uday Surendra Yandamuri. (2022). Cloud-Based Data Integration Architectures for Scalable Enterprise Analytics. International Journal of Intelligent Systems and Applications in Engineering, 10(3s), 472–483. Retrieved from https://ijisae.org/index.php/IJISAE/article/view/8005

[94] Inala, R. Advancing Group Insurance Solutions Through Ai-Enhanced Technology Architectures And Big Data Insights.

[95] Maguluri, K. K., Pandugula, C., Kalisetty, S., & Mallesham, G. (2022). Advancing Pain Medicine with AI and Neural Networks: Predictive Analytics and Personalized Treatment Plans for Chronic and Acute Pain Managements. Journal of Artificial Intelligence and Big Data, 2(1), 112-126.

[96] Otto, B. (2011). A Morphology of the Organisation of Data Governance. European Journal of Information Systems, 20(4), 429–442.

[97] Avinash Reddy Aitha. (2022). Deep Neural Networks for Property Risk Prediction Leveraging Aerial and Satellite Imaging. International Journal of Communication Networks and Information Security (IJCNIS), 14(3), 1308–1318. Retrieved from https://www.ijcnis.org/index.php/ijcnis/article/view/8609

[98] Armbrust, M., Fox, A., Griffith, R., Joseph, A. D., Katz, R., Konwinski, A., Lee, G., Patterson, D., Rabkin, A., & Stoica, I. (2010). A View of Cloud Computing. Communications of the ACM, 53(4), 50–58.

[99] Goutham Kumar Sheelam. (2022). Reconfigurable Semiconductor Architectures For AI-Enhanced Wireless Communication Networks. Kurdish Studies, 10(2), 1027–1040. https://doi.org/10.53555/ks.v10i2.3867

[100] Khatri, V., & Brown, C. V. (2010). Designing Data Governance. Communications of the ACM, 53(1), 148–152.

[101] Ramesh Inala. (2022). Cross-Domain MDM Integration Using AI-Driven Data Governance: A Case Study In Financial Technology Architecture. Migration Letters, 19(2), 280–304. Retrieved from https://migrationletters.com/index.php/ml/article/view/11982

[102] Grolinger, K., Higashino, W. A., Tiwari, A., & Capretz, M. A. M. (2013). Data Management in Cloud Environments: NoSQL and NewSQL Data Stores.

[103] Sheelam, G. K. Power-Efficient Semiconductors for AI at the Edge: Enabling Scalable Intelligence in Wireless Systems. International Journal of Innovative Research in Electrical, Elec-tronics, Instrumentation and Control Engineering (IJIREEICE), DOI, 10.

[104] Alharthi, A., Krotov, V., & Bowman, M. (2017). Addressing Barriers to Big Data. Business Horizons, 60(3), 285–292.

[105] Dodda, A., Lakkarasu, P., Singireddy, J., Challa, K., & Pamisetty, V. (2022). Optimizing Digital Finance and Regulatory Systems Through Intelligent Automation. Secure Data Architectures, and Advanced Analytical Technologies.

[106] Abraham, R., Schneider, J., & vom Brocke, J. (2019). Data Governance: A Conceptual Framework, Structured Review, and Research Agenda.

[107] Pandiri, L., Singireddy, S., & Adusupalli, B. (2020). Digital Transformation of Underwriting Processes through Automation and Data Integration. Global Research Development (GRD) ISSN, 2455-5703.

[108] Ladley, J. (2012). Data Governance: How to Design, Deploy, and Sustain an Effective Data Governance Program. Morgan Kaufmann.

[109] Sheelam, G. K., & Nandan, B. P. (2022). Integrating AI And Data Engineering For Intelligent Semiconductor Chip Design And Optimization. Migration Letters, 19, 2178-2207.

[110] Tallon, P. P. (2013). Corporate Governance of Big Data: Perspectives on Value, Risk, and Cost. Computer, 46(6), 32–38.