Handling Inconsistent Government Data: From Acquisition to Entity Name Matching and Address Standardization

Davyson S. Ribeiro[1,5][a], Paulo V. A. Fabrício[1,5][b], Rafael R. Pereira[1,5][c], Tales P. Nogueira[2,5][d], Pedro A. M. Oliveira[3,5][e], Victória T. Oliveira[1,5][f], Ismayle S. Santos[4,5][g] and Rossana M. C. Andrade[1,5][h]

[1] *Federal University of Ceará, Fortaleza, Brazil*

[2] *University of the International Integration of the Afro-Brazilian Lusophony, Redenção, Brazil*

[3] *Federal Institute of Education, Science, and Technology of Maranhão, Pedreiras, Brazil*

[4] *State University of Ceará, Fortaleza, Brazil*

[5] *Group of Computer Networks, Software Engineering and Systems, Fortaleza, Brazil*

Keywords: Data Science, Big Data, Business Intelligence, Data Analysis, Decision Support Systems, Decision-Making.

Abstract: The integration of Data Science and Big Data is essential for managing large-scale data, but challenges such as heterogeneity, inconsistency, and data enrichment complicate this process. This paper presents a flexible architecture designed to support municipal decision-making by integrating data from multiple sources. To address inconsistencies, an entity matching algorithm was implemented, along with an address standardization library, optimizing data processing without compromising quality. The study also evaluates data acquisition methods (APIs, Web Crawlers, HTTPS requests), highlighting their trade-offs. Finally, we demonstrate the system's practical impact through a case study on health data monitoring, showcasing its role in enhancing data-driven governance.

1 INTRODUCTION

The ethical use of data, information, and knowledge derived from digital traces left by computational devices has been a growing topic in Information Science (Rautenberg and do Carmo, 2019). The intelligent use of data enables the development of innovative applications and services, such as data integration platforms that provide visualization and predictive analysis tools for decision-making and strategic planning, contributing to urban development and public policy improvements. However, handling large volumes of data presents challenges that traditional database systems and batch processing struggle to address due to performance and scalability limitations. In this context, cloud computing offers elasticity and scalability, making it a key solution for managing and analyzing massive datasets (Sandhu, 2021).

Significant investments from the private sector have driven the development of technologies capable of handling large-scale data. Platforms like Amazon Web Services (AWS) provide specialized tools for Data Science, including Amazon S3 for storage, AWS Glue for serverless data integration, and AWS Athena for interactive analysis (Vines and Tanasescu, 2023).

This study reports the experiences gained during the development of a data-driven decision-making platform designed to support municipal government managers in Brazil. The project extensively leveraged cloud computing and Data Science techniques to address key challenges, such as data heterogeneity and inconsistency. The datasets were independently produced by different municipal departments, including Education, Health, and Urban Planning, along with federally managed centralized databases.

[a] https://orcid.org/0000-0002-7375-0684
[b] https://orcid.org/0009-0000-8326-7197
[c] https://orcid.org/0009-0000-4838-6617
[d] https://orcid.org/0000-0002-3266-4632
[e] https://orcid.org/0000-0002-3067-3076
[f] https://orcid.org/0000-0002-1400-522X
[g] https://orcid.org/0000-0001-5580-643X
[h] https://orcid.org/0000-0002-0186-2994

Ribeiro, D. S., Fabrício, P. V. A., Pereira, R. R., Nogueira, T. P., Oliveira, P. A. M., Oliveira, V. T., Santos, I. S. and Andrade, R. M. C.
Handling Inconsistent Government Data: From Acquisition to Entity Name Matching and Address Standardization.
DOI: 10.5220/0013294200003929

In *Proceedings of the 27th International Conference on Enterprise Information Systems (ICEIS 2025) - Volume 1*, pages 286-293
ISBN: 978-989-758-749-8; ISSN: 2184-4992

Despite the potential of Big Data in government decision-making, several challenges hinder its practical implementation. The heterogeneity and inconsistency of data from diverse sources, lack of standardization in collection methods, and the need for scalable, automated data pipelines increase operational costs and compromise analysis quality. This work proposes a robust framework integrating automation, preprocessing, validation techniques, and standardization practices to improve the quality and usability of both textual and geographic data.

The remainder of this paper is structured as follows: Section 2 reviews related works, discussing methodologies relevant to data processing challenges. Section 3 presents the foundational concepts of Data Science and the key stages of the project's implementation. Section 4 details the cloud computing architecture, emphasizing scalability, security, and operational advantages. Section 5 describes the functional and non-functional requirements necessary for efficient Big Data management in government systems. Section 6 explores the main challenges encountered and the solutions implemented. Section 7 evaluates the system's applicability, discusses technical and operational constraints, and presents case studies demonstrating its impact. Finally, Section 8 concludes with a summary of findings and future research directions.

2 RELATED WORK

This section presents works that incorporate various Data Science and Big Data methods, focusing on their application to decision-making and the development of new applications. Sarker (2021) discusses the relevance of advanced data analysis methods across different sectors, emphasizing their impact on decision-making, operational optimization, and trend forecasting. While it highlights the importance of customized applications in healthcare, smart cities, and cybersecurity, it does not specifically address government data standardization and harmonization.

Freitas et al. (2023) propose a data warehousing environment for crime data analysis, supporting public security managers in strategic decision-making. The study identifies key challenges, such as data heterogeneity, lack of standardization, and the need for advanced extraction, transformation, and visualization techniques.

Fugini and Finocchi (2020) focus on documental Big Data processing, introducing an Enterprise Content Management (ECM) system enhanced with machine learning for classification and information extraction. Their work defines quality metrics—Textual Quality Confidence, Classification Confidence, and Extraction Confidence—to assess system accuracy and efficiency. These indicators contribute to data integrity and consistency but do not fully address heterogeneous governmental data integration.

Behringer et al. (2023) present SDRank, a deep learning-based approach for ranking data sources by similarity, optimizing semantic pattern recognition and automated data selection. While this technique improves efficiency and scalability in large-scale data processing, it does not tackle structural inconsistencies in government datasets.

Furtado et al. (2023) investigate digital transformation in smart governance, analyzing how Big Data tools can support policies aimed at vulnerable populations. However, their study does not explore the technical challenges of integrating and standardizing multiple governmental data sources.

These studies provide valuable insights into Big Data applications, yet they lack a detailed architectural perspective on handling heterogeneous and inconsistent government data. This work addresses these gaps by proposing a scalable integration pipeline, combining automation, entity name matching, and address standardization to enhance data quality and usability in public sector applications.

3 DATA SCIENCE, BIG DATA AND PROJECT STAGES

Big Data refers to large, heterogeneous datasets that exceed the processing capabilities of conventional methods due to their dynamic and complex nature. These datasets exhibit characteristics such as volume, velocity, variety, veracity, variability, and value, requiring specialized techniques for their management. In this study, Big Data encompasses diverse governmental datasets, used to build analytical tools that support municipal decision-making. Given these characteristics, a key challenge is ensuring data storage, cataloging, and availability for decision-making processes.

Data Science, an interdisciplinary field integrating statistics, mathematics, and computer science, enables the extraction of valuable insights to support data-driven decisions (Wu et al., 2021). Identifying patterns, trends, and hidden relationships within data is complex but essential for predictions, process optimization, and strategic decision-making (Sarker, 2021). Transforming raw data into actionable knowledge involves several critical stages, as illustrated in Figure 1. The data acquisition stage involves col-

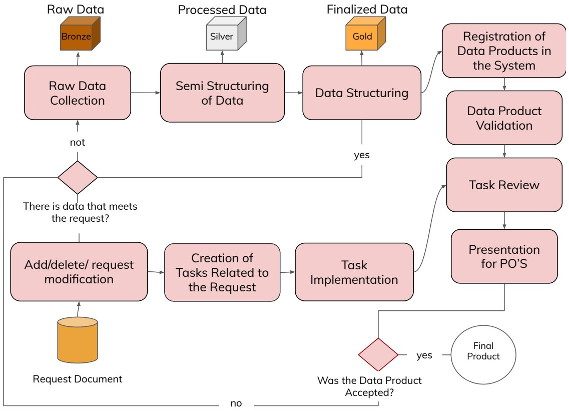

Figure 1: Steps used to build the product.

lecting raw data and metadata from multiple sources based on municipal department requirements. The data ingestion phase transforms and loads this raw data into a centralized repository, structuring it for easy access and analysis. Next, the data exploration stage enables the preliminary study of datasets, rendering them semi-structured to facilitate the definition of workflows and hypotheses. During information extraction, advanced analytical techniques identify relevant patterns and insights, refining the data into actionable knowledge. The information display stage ensures clear communication of insights through detailed reports, interactive dashboards, and visualizations. Finally, in the decision-making phase, analytical models combine with domain expertise to generate strategic recommendations, supported by data visualization and reporting.

4 CLOUD COMPUTING AND PROJECT ARCHITECTURE

Cloud computing has transformed data storage and processing, allowing organizations to reduce costs and enhance scalability by using third-party infrastructure (Silva et al., 2023). The cloud services market is led by AWS, Microsoft Azure, Google Cloud Platform (GCP), and IBM Cloud, each offering solutions across Infrastructure as a Service (IaaS), Platform as a Service (PaaS), and Software as a Service (SaaS) (Gupta et al., 2021). AWS stands out for its wide service range and global presence, while Azure integrates seamlessly with Microsoft tools, GCP focuses on AI and machine learning, and IBM Cloud specializes in hybrid cloud solutions.

This competition fosters constant innovation and price reductions, benefiting users by enhancing service quality, security, and compliance.

4.1 Using the AWS Platform

AWS was chosen due to its scalability, flexibility, security, and cost-effectiveness, providing an intuitive environment for activity monitoring, data structuring, and automated rule enforcement. Key AWS services used include Amazon S3 (storage), AWS Glue (serverless ETL), AWS Athena (querying), and AWS Lambda (automation).

The architecture was designed to ensure compliance with GDPR and LGPD, adhering to strict data protection regulations. To achieve this, principles such as data minimization and storage limitation were implemented, processing only essential information for the shortest necessary period. Anonymization and pseudonymization techniques were applied to safeguard personal data while maintaining analytical accuracy.

Access control is managed through AWS Identity and Access Management (IAM), enforcing granular permissions. Password rotation, two-factor authentication, and continuous monitoring via AWS CloudTrail and Amazon CloudWatch ensure system security. AWS follows a shared responsibility model, managing physical infrastructure security while users configure data protection policies. AWS compliance with ISO/IEC 27001, SOC 1, 2, 3, and PCI DSS facilitates adherence to international regulations.

These security measures ensure the integrity, confidentiality, and reliability of the system, providing a secure environment for data analysis while maintaining ethical and transparent data handling.

4.2 Data Acquisition and Storage

Data acquisition follows three primary methods: APIs, HTTPS requests, and manual ingestion. AWS Lambda functions retrieve data based on provider access methods, storing raw data in AWS S3 buckets.

The system follows the Medallion Architecture, a structured data lake model that organizes data into Bronze, Silver, and Gold layers (Kumar et al., 2023). The Bronze layer stores raw, unprocessed data, maintaining full historical records for auditing. The Silver layer applies data cleaning, filtering, and enrichment, creating a structured dataset. Finally, the Gold layer contains highly refined and aggregated data, ready for business intelligence, machine learning, and analytics.

AWS Glue facilitates data transformation using ProcessingJob (to convert unstructured data into semi-structured data) and BusinessJob (to create structured datasets for BI applications).

4.3 Workflow Automation

Project workflows are automated using AWS Glue components, including Crawlers and Triggers. Crawlers scan and classify data, updating the AWS Glue Data Catalog, which maintains metadata for querying and ETL jobs. Triggers initiate ETL processes based on time-based, event-based, or dependency-based conditions, automating the entire pipeline workflow.

A well-defined automation strategy ensures efficiency in managing heterogeneous data formats and resolving inconsistencies, eliminating the need for manual intervention. Figure 2 illustrates the workflow automation process.

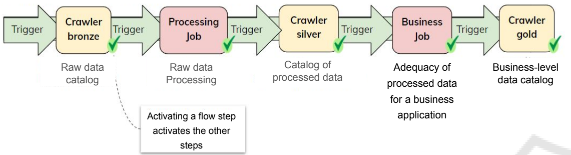
Trigger
Crawler bronze
Trigger
Processing Job
Trigger
Crawler silver
Trigger
Business Job
Trigger
Crawler gold
Raw data catalog
Raw data Processing
Catalog of processed data
Adequacy of processed data for a business application
Business-level data catalog
Activating a flow step activates the other steps

Figure 2: Process of Automating a Workflow.

The automation process consists of five stages: I. Raw Data Verification: A trigger checks for new data in the Bronze bucket and catalogs it using a Bronze Crawler. II. Processing Job Initiation: Upon detecting new Bronze data, a ProcessingJob transforms it into a semi-structured format. III.Silver Data Cataloging: A trigger detects new processed data in the Silver bucket and updates the Silver Crawler. IV. Business Rule Execution: A trigger initiates a BusinessJob that applies domain-specific transformations to prepare data for BI use. V. Gold Data Finalization: A final trigger detects new structured data in the Gold bucket, cataloging it for analysis.

By structuring the workflow this way, the system ensures consistent, reliable, and automated data processing, supporting evidence-based public policies with high-quality, standardized data.

5 REQUIREMENTS FOR EFFICIENT BIG DATA MANAGEMENT IN GOVERNMENT SYSTEMS

The use of Big Data in public administration is fundamental for developing evidence-based policies. This study proposes an integrated and standardized approach to handle large volumes of heterogeneous data, ensuring its interoperability and usability for advanced analysis and decision-making. The system architecture follows a structured workflow covering data acquisition, standardization, entity matching, processing, and visualization. To support its implementation, a set of functional and non-functional requirements was identified, ensuring efficiency, security, and scalability in public sector applications.

5.1 Functional Requirements and Non-Functional Requirements

The system is structured into three key layers: data acquisition and integration, processing and analysis, and access and visualization. The data acquisition and integration layer must collect information from government databases, public APIs, structured/unstructured files, and web scraping. To ensure interoperability, the system must support automatic data conversion and standardization, enabling consistent storage. Additionally, entity matching techniques must be implemented to unify records referring to the same entity, such as institutions, locations, or individuals, reducing duplication and improving data quality.

The processing and analysis layer must handle large-scale data through scalable pipelines, following the Medallion Architecture, which structures data into three layers: Bronze (raw data), Silver (refined data), and Gold (ready-for-analysis data). Machine learning and artificial intelligence models must optimize address standardization, error correction, and predictive analytics, while an automated data quality assessment mechanism should identify and correct inconsistencies before analysis.

The access and visualization layer must present data in an accessible and interactive manner, providing dynamic dashboards and geospatial analysis tools to identify patterns in areas such as urban mobility, public safety, and healthcare. The system interface must be intuitive, responsive, and compliant with digital accessibility guidelines, ensuring broad usability. Access control mechanisms should be enforced to regulate data access, adhering to regulations such as LGPD and GDPR.

For Non-Functional Requirements, the system must guarantee performance, scalability, and security, meeting robust technical requirements. To ensure scalability, the architecture must be distributed and modular, processing large data volumes with low latency and high efficiency, while adapting to increased demand. Optimized processing pipelines should support high-concurrency workloads, ensuring system responsiveness.

Regarding usability and user experience, the interface must facilitate intuitive navigation and provide continuous training and technical support, enabling users to fully utilize analytical tools.

Security and data governance are critical aspects. The system must implement robust encryption to protect sensitive information and multifactor authentication to prevent unauthorized access. Additionally, automated backups must ensure data recovery in case of failures.

For system management and maintenance, institutional responsibilities must be well-defined, ensuring systematic data updates and continuous improvements. Automated monitoring and auditing tools should be implemented to detect anomalies, facilitating proactive adjustments. Comprehensive documentation must be maintained to support future expansions and integrations.

5.2 Strategic Impact of the Architecture

A Big Data architecture for public administration significantly enhances efficiency, transparency, and policy formulation. By optimizing resource allocation, reducing waste, and improving decision-making processes, the system strengthens the government's ability to address complex challenges.

Furthermore, the platform fosters greater transparency, allowing citizens to access and monitor public policies. Aligning the architecture with national and international digital governance frameworks ensures compliance with best practices, driving efficiency and innovation.

The integration of artificial intelligence and open data initiatives positions this model as a cornerstone for modernizing public administration. The clear definition of functional and non-functional requirements, combined with a scalable and robust architecture, ensures that the system can support advanced analytics, efficiently manage data, and drive sustainable and impactful public policies.

6 CHALLENGES AND SOLUTIONS

This section presents the main challenges encountered related to data acquisition methods, as well as the solutions employed to address issues of data heterogeneity and inconsistency.

6.1 Heterogeneity in Formats and Access Methods

Data ingestion is a critical step in data-driven projects, requiring the collection and integration of heterogeneous data sources into a centralized storage system for analysis. In this project, governmental data fragmentation posed challenges due to the lack of standardized acquisition methods across departments, each using distinct operational protocols and management strategies. To address this, three primary data collection methods were employed: APIs, HTTPS requests, and Web Crawlers.

Web Crawlers were used to extract data from public health systems such as SIM (Mortality Information System) and SINASC (Live Birth Information System), both accessible via TABNET[1]. This method ensured real-time data access, independent maintenance, and fewer availability issues compared to APIs. However, it had limitations, including inability to access sensitive microdata, high maintenance costs due to frequent web structure changes, AWS Lambda storage constraints, and potential legal implications related to terms-of-service compliance.

APIs provided structured and reliable data in JSON format, with advantages such as well-documented services, access control via tokens, and secure integration. However, they also presented schema standardization issues, reliance on external providers without guaranteed SLAs, and challenges with large datasets requiring pagination. In cases of extensive data, AWS Glue Jobs were required, increasing operational costs.

HTTPS requests were used for direct data downloads (e.g., Education data in CSV format). This approach was simple and flexible, allowing customized extractions without third-party dependence. However, it required manual schema adjustments, additional processing for unstructured data, and lacked optimization for large datasets, impacting storage costs.

Static datasets, such as neighborhood HDI data, were updated infrequently and manually uploaded to the Bronze storage layer, initiating the data pipeline. The data acquisition process using these three methods is illustrated in Figure 3.

6.2 Inconsistency Between Data from Entities in Different Databases

Data inconsistency is a frequent challenge in multi-source data integration, often arising from spelling variations, inconsistent date formats, and missing values, complicating direct comparisons. To ensure data accuracy and usability, effective comparison and consolidation methods are required.

[1] https://datasus.saude.gov.br/informacoes-de-saude-tabnet/

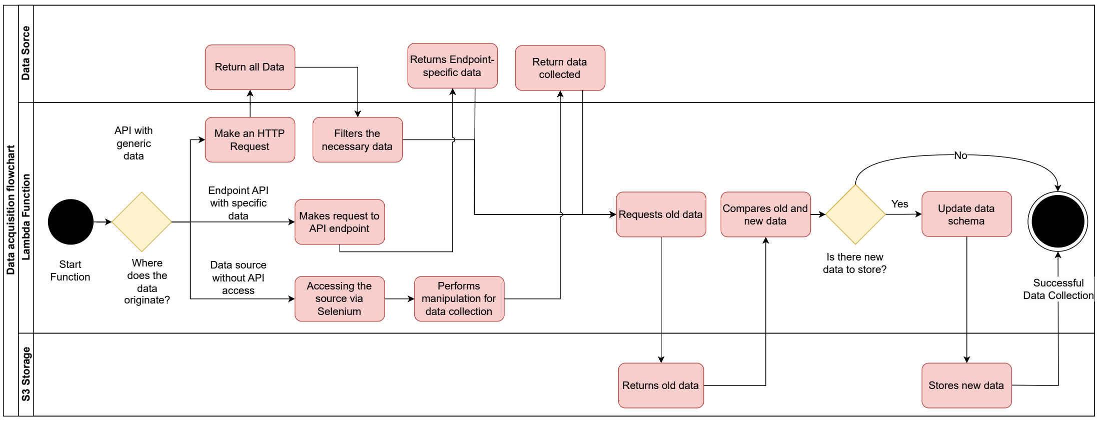
Data acquisition flowchart
Data Sorce
Lambda Function
S3 Storage
Return all Data
Returns Endpoint-specific data
Return data collected
API with generic data
Make an HTTP Request
Filters the necessary data
Endpoint API with specific data
Makes request to API endpoint
Data source without API access
Accessing the source via Selenium
Performs manipulation for data collection
Start Function
Where does the data originate?
Requests old data
Compares old and new data
Is there new data to store?
Yes
No
Update data schema
Successful Data Collection
Returns old data
Stores new data

Figure 3: Data acquisition flowchart.

To address spelling variations, an algorithm was developed using RapidFuzz[2] and InDel distance calculations, optimizing the comparison of key fields such as names and birth dates. The QRatio method was selected due to its balance between computational efficiency and accuracy, with a 90

To handle large-scale data, a sparse matrix approach was used, storing only relevant similarity scores to optimize memory and computational resources. Additionally, an internal comparison process was implemented to detect duplicate or near-duplicate records, ensuring that the most complete and recent version of a record was preserved. The similarity function implementation is illustrated in Figure 4, demonstrating key attributes considered during record comparison.

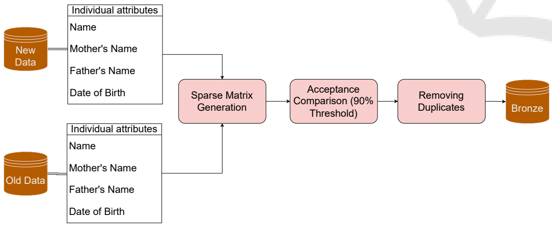

Figure 4: Similarity Flowchart.

This approach provided an efficient and scalable solution for integrating multiple data sources, significantly improving data consistency and reliability. By combining similarity calculation techniques with optimized data structures, this method enhanced data consolidation and quality, ensuring a structured dataset for subsequent analyses.

6.3 Address Data

Geographic data in municipal systems often includes address and neighborhood fields, but manually entered information is prone to inaccuracies such as typographical errors and outdated names due to urban changes like street renaming, neighborhood mergers, and administrative boundary shifts. These inconsistencies challenge data consistency.

To ensure accurate data representation, the platform developed choropleth maps, requiring neighborhood names to match exactly with those in a GeoJSON file provided by the City's Planning Institute. This file adhered to the latest municipal decree, which restructured several neighborhoods. Consequently, non-standard names had to be corrected for proper data aggregation and visualization.

To address this issue, a dedicated Python package was created, using the GeoJSON file as a reference to standardize geographic data throughout the pipeline. AWS Glue dynamically installed this library, enabling automated neighborhood name corrections and ensuring consistent data in the Silver and Gold layers. Among the main functions provided by the library, the following stand out:

- **get_nome_canonico (get_canonical_name).** Returns the official neighborhood name according to the latest municipal decree. The algorithm first ”simplifies” the neighborhood name (removes accent marks and converts it to uppercase) and searches for a match with neighborhoods in the reference GeoJSON file. If an exact match is not found, it performs a fuzzy search and returns the official neighborhood name that is most similar. For example, a neighborhood named ”Centro” could appear as ”Centr,” ”Cntro,” or ”Cêtro”. Taking the last example, the function would initially convert ”Cêtro” to ”CETRO” and subsequently

[2] https://github.com/rapidfuzz/RapidFuzz

employ a fuzzy search, which would probably identify ”Centro” as the canonical name.

- **get_bairro (get_neighborhood).** Returns the neighborhood name given a latitude and longitude pair. This function uses the Shapely library[3] to create a *Point* geometry and checks, for each neighborhood, if that point is contained within its polygon.
- **get_localizacao (get_location).** Returns the location (latitude and longitude pair) given a free-text address. This method uses the GeoPy library[4], specifically with the Nominatim[5] (OpenStreetMap data) and GoogleV3 (Google Maps data) geocoders, to perform geocoding. The user can choose the desired geocoder. During the project, Google Maps data proved to be more accurate for the region of interest. However, this API requires authorization via a token and may incur costs if numerous queries are performed. This technique proved to be more cost-effective than AWS alternatives, which rely on HERE Technologies and Esri data[6].
- **get_endereco (get_address).** Returns address data given a latitude and longitude pair. This method is useful in cases where the exact location is available, but the textual address data is missing.

Isolating utility functions like those described above into a dedicated library proved advantageous for the project, as it centralized neighborhood name handling and other geolocation functions within a single repository, thereby avoiding code duplication and potential maintenance costs associated with updating code across multiple ETL jobs. These functions retained their conceptual integrity throughout the project, with no changes to their method signatures, demonstrating good cohesion and serving as an acceptable point of coupling between the library and the ETL jobs that utilized it.

7 APPLICABILITY, TECHNICAL AND OPERATIONAL LIMITATIONS

This section presents one of the significant results that were obtained from the implementation of the system, despite the contributions presented, it is essential to recognize the technical and operational limitations encountered during its development and implementation.

7.1 Vaccine Monitoring and Updating

TThe system’s applicability was demonstrated through a case study on childhood vaccination monitoring in public daycare centers. By integrating health and education data, the platform provided real-time insights into vaccination gaps among children aged 0 to 3 years, enabling targeted interventions.

Cross-referencing data identified 11,854 overdue vaccinations among 4,657 children, highlighting the need for urgent action. Targeted campaigns based on this analysis resulted in over 2,000 children updating their vaccination records within a month, demonstrating the platform’s efficacy in guiding public health strategies.

Continuous monitoring enabled public managers to track progress and measure intervention effectiveness. Subsequent data collection showed a significant reduction in overdue vaccinations, with 637 children achieving full immunization compliance. However, new enrollments revealed recurring vaccination gaps, reinforcing the need for ongoing and adaptive public health initiatives.

Beyond improving immunization coverage, the platform optimized resource allocation, ensuring that funding and efforts were focused on high-risk areas. Additionally, it enhanced collaboration between health and education departments, fostering a holistic approach to policy planning.

The case study highlights the transformative power of data-driven decision-making in public administration. By integrating and analyzing multi-sectoral data, municipal governments can improve service quality, optimize resources, and implement evidence-based policies that effectively address critical public health challenges.

7.2 Operational Challenges

The system was designed to process large-scale heterogeneous data, but scalability constraints may arise as data sources and records grow. Expanding cloud resources can increase execution time and operational costs, while AWS Lambda execution limits restrict real-time processing. Despite AWS’s advanced tools and scalability, reliance on proprietary cloud services introduces challenges. Costs for S3, Glue, and Athena may escalate in data-intensive scenarios, and pipeline interoperability is limited in organizations using alternative cloud or on-premises solutions. Cost manage-

[3] https://github.com/shapely/shapely

[4] https://github.com/geopy/geopy

[5] https://nominatim.openstreetmap.org/

[6] https://aws.amazon.com/location/

ment is crucial, particularly for government institutions with budget constraints, as on-demand services may lead to unpredictable expenses without proper strategies. Additionally, maintaining the pipeline in a rapidly evolving technology landscape requires frequent updates and specialized expertise, making long-term scalability and sustainability a challenge.

8 FINAL REMARKS

This work presented an architecture to address heterogeneity and inconsistency in large datasets using cloud computing. The proposed framework improves data analysis quality for decision-making and is adaptable to various applications. Future research may explore Infrastructure as Code (IaC) to enhance automation, scalability, and resource management, optimizing data processing efficiency.

ACKNOWLEDGMENTS

Finally, we would like to thank FUNCAP for the project financial support and CNPq for the productivity grant awarded to Rossana M. C. Andrade (306362/2021-0).

REFERENCES

Behringer, M., Treder-Tschechlov, D., Voggesberger, J., Hirmer, P., and Mitschang, B. (2023). Sdrank: A deep learning approach for similarity ranking of data sources to support user-centric data analysis. In *Proceedings of the 25th International Conference on Enterprise Information Systems*, page 419–428. SCITEPRESS - Science and Technology Publications.

Freitas, J. B., Clarindo, J., and Aguiar, C. (2023). Ambiente de data warehousing espacial para tomada de decisão sobre dados de crimes. In *Anais Estendidos do XXXVIII Simpósio Brasileiro de Bancos de Dados*, pages 36–42, Porto Alegre, RS, Brasil. SBC.

Fugini, M. and Finocchi, J. (2020). Quality evaluation for documental big data. In *Proceedings of the 22nd International Conference on Enterprise Information Systems*, page 132–139. SCITEPRESS - Science and Technology Publications.

Furtado, L. S., da Silva, T. L. C., Ferreira, M. G. F., de Macedo, J. A. F., and Moreira, J. K. M. L. C. (2023). A framework for digital transformation towards smart governance: using big data tools to target sdgs in ceará, brazil. In *TEMPLATE'06, 1st International Conference on Template Production*. Journal of Urban Management.

Gupta, B., Mittal, P., and Mufti, T. (2021). A review on amazon web services (aws), microsoft azure & google cloud platform (gcp) services. In *Proc. of the 2nd International Conference on ICT for Digital, Smart, and Sustainable Development*.

Kumar, A., Mishra, A., and Kumar, S. (2023). Data lake, lake house, and delta lake. In *Architecting a Modern Data Warehouse for Large Enterprises: Build Multi-cloud Modern Distributed Data Warehouses with Azure and AWS*, pages 95–160. Springer.

Rautenberg, S. and do Carmo, P. R. V. (2019). Big data e ciência de dados: complementariedade conceitual no processo de tomada de decisão. *Brazilian Journal of Information Science*, 13(1):56–67.

Sandhu, A. K. (2021). Big Data with Cloud Computing: Discussions and Challenges. *Big Data Mining and Analytics*, 5(1):32–40.

Sarker, I. H. (2021). Data science and analytics: An overview from data-driven smart computing, decision-making and applications perspective. In *TEMPLATE'06, 1st International Conference on Template Production*. Springer Science and Business Media LLC.

Silva, M. E., Pinheiro, F., Bezerra, C., and Coutinho, E. (2023). Modelagem de Ecossistemas de Software das Plataformas de Computação em Nuvem AWS e GCP. In *Anais Estendidos do XIX Simpósio Brasileiro de Sistemas de Informação*, pages 172–177, Porto Alegre, RS, Brasil. SBC.

Vines, A. and Tanasescu, L. (2023). An Overview of ETL Cloud Services: An Empirical Study Based on User's Experience. In *Proceedings of the International Conference on Business Excellence*, volume 17, pages 2085–2098.

Wu, Y., Zhang, Z., Kou, G., Zhang, H., Chao, X., Li, C.-C., Dong, Y., and Herrera, F. (2021). Distributed linguistic representations in decision making: Taxonomy, key elements and applications, and challenges in data science and explainable artificial intelligence. Journal of Urban Management.