2024 IEEE/ACM International Conference on Big Data Computing, Applications and Technologies (BDCAT) | 979-8-3503-6730-0/24/$31.00 ©2024 IEEE | DOI: 10.1109/BDCAT63179.2024.00028

PlayField: An Adaptable Framework for Integrative Sports Data Analysis

Filipe Pinto
Faculty of Engineering of the University of Porto
Portugal
up201907747@edu.fe.up.pt

Bruno Lima
University of Maia,
Faculty of Engineering of the University of Porto,
and LIACC
Portugal
brunomclima@gmail.com

***Abstract*—As sports analytics evolve to include a broad spectrum of data from diverse sources, the challenge of integrating heterogeneous data becomes pronounced. Current methods struggle with flexibility and rapid adaptation to new data formats, risking data integrity and accuracy. This paper introduces PlayField, a framework designed to robustly handle diverse sports data through adaptable configuration and an automated API. PlayField ensures precise data integration and supports manual interventions for data integrity, making it essential for accurate and comprehensive sports analysis. A case study with ZeroZero demonstrates the framework's capability to improve data integration efficiency significantly, showcasing its potential for advanced analytics in sports.**

***Index Terms*—Data Integration, Data Mapping, Heterogeneous Data**

I. Introduction

Sports analytics has progressed from simple statistical analysis to advanced metrics covering everything from major leagues to amateur competitions. This broadened scope includes player tracking and in-depth analysis at all levels, reflecting the increased complexity and diversity of data sources in sports. Modern sports analytics now integrates varied data formats—structured and unstructured, like JSON, XML, and CSV—to provide deeper insights, which are crucial for strategic decisions in sports management and athlete performance.

However, traditional data integration methods struggle with the volume, velocity, and variety of contemporary sports data, emphasizing the need for a robust, adaptable ETL process tailored to these specific requirements.

This research was conducted in partnership with ZeroZero (www.zerozero.pt/), the world's largest sports-related information database, focusing on football. By collaborating closely with ZeroZero, we developed and tested a framework that addresses the unique challenges of modern sports data integration, ensuring our solutions were practical and effective.

II. PlayField

The current data integration and API management solutions are advanced and comprehensive but do not entirely address the issue of converting different sports data into a standard format. They are limited in handling diverse and complex data sources, integrating real-time data efficiently, and creating precise mappings to maintain data consistency and integrity.

To address this problem, the proposed solution focuses on data mapping, harmonizing different data structures and formats, making integration and analysis easier. This solution allows users to upload files and transform them into a specified output structure using manually or automatically created rules. This user-centric approach ensures flexibility and accuracy in data transformation, catering to the specific needs of sports data.

Additionally, the solution includes an API that automates the transformation process by applying the previously created rules. This automation simplifies data integration, allowing for real-time processing and continuous updates. By focusing on data mapping and using an API for automation, the solution guarantees accurate integration of diverse data sources, providing a standardized dataset ready for analysis.

A. Requirements

The PlayField framework has several core features to facilitate efficient data handling, transformation, and visualization. These features provide a seamless user experience and ensure the framework meets the needs of various stakeholders. Below are the detailed descriptions of the main features:

User Interface and Interaction: The user interface (UI) of PlayField is designed to be intuitive and user-friendly, enabling users to perform complex data integration tasks with ease. Streamlit, the Python library used to build the UI, allows for rapid development and deployment of interactive web applications.

Dropdown Menus for Data Selection: The PlayField interface features drop-down menus to select data types and providers. This intuitive design allows users to easily choose from predefined data types and providers or add new ones. Users can select the type of information they are dealing with, such as match feeds, competitions, or odds data. This helps in filtering the data and applying the correct transformation rules. Additionally, the provider selection dropdown allows users to select an existing provider or add a new one using the "Add New Provider" option. When adding a new provider, a text

979-8-3503-6730-0/24/$31.00 ©2024 IEEE
DOI 10.1109/BDCAT63179.2024.00028

input field appears where users can enter the provider's name. Upon submission, the new provider is saved in the database and becomes the pre-selected option in the dropdown.

File Uploader: After selecting the type of information and the corresponding provider, users can upload the input file they wish to convert. This functionality ensures that the framework can handle various file types and sizes, making it adaptable to different data sources. The file uploader component allows users to drag and drop files or select files from their computer. Supported file types include CSV, JSON, and XML, among other text files. This flexibility is essential for dealing with diverse data formats from different providers.

Rule Management: Effective rule management is a fundamental aspect of the PlayField framework, enabling users to specify the transformation of data from its original format to the standardized format required by the target database.

Preview and Edit Rules: The framework provides robust rule management features to ensure accurate data transformation. A button opens a pop-up, displaying all existing rules for the selected provider. Users can review these rules and delete any that are no longer needed. This feature is crucial for maintaining data integrity and ensuring that outdated or incorrect rules do not affect the transformation process. Rules are specific to each provider, and users can only preview the rules associated with the currently selected provider. Additionally, this feature allows users to see the important/mandatory fields in the output file. By toggling checkboxes, users can include or exclude fields, tailoring the output to their needs. This customization ensures that only relevant data is included in the final output, optimizing both storage and processing efficiency.

Data Mapping and Rule Creation: The data mapping and rule creation process is a critical step in transforming the input data into a standardized output format. The interface displays two columns, representing the input and output nodes with their respective fields. Users can map these fields to create rules that define how data from the input should be transformed to fit the output structure. Input nodes represent the fields present in the uploaded input file, which need to be transformed. Output nodes represent the fields required in the final output file, adhering to the standardized format used by the target database.

Dropdown for Node Mapping: Each column contains dropdowns with the names of the nodes, enabling users to select and match nodes from the input to the output. This mapping process is crucial for establishing a clear and accurate data transformation path. Users can select the appropriate input field from a dropdown list and the corresponding output field from a dropdown list.

Rule Storage and Application: Once rules are created, they are stored in the PostgreSQL database. This storage mechanism ensures that all transformations are recorded and can be reused or modified as needed. Rules are stored in a structured format, enabling quick retrieval and application during the data transformation process. The framework supports exact matches between input and output fields and hierarchical rule creation for nested structures like arrays and dictionaries. This flexibility is essential for handling complex data formats and ensuring that all relevant information is accurately transformed. Exact matches ensure that fields with identical names are mapped directly, simplifying the rule creation process. Hierarchical rule creation allows users to define rules for nested structures, such as arrays within dictionaries, which is particularly useful for complex data sets containing multiple levels of nested information.

Exporting Transformed Data: After the rules are applied, users can export the transformed input data into a standardized format, typically a JSON file. This export feature ensures that the data is ready for integration into the target database or other systems. An ”Export Transformed Input” button triggers the export process, converting the transformed data into a JSON file.

Automation through API: To further enhance the framework's capabilities, an API was developed to automate the data integration process. The PlayField framework includes two API endpoints to facilitate automated data transformation and integration. The ”List Providers” endpoint returns a list of all registered providers, allowing users to see which providers are available for data transformation. The ”Transform Input” endpoint accepts the provider's name and the input file in the request body, returning the transformed data. This allows for seamless integration with external systems and automated workflows.

In summary, the PlayField framework offers a comprehensive set of features designed to streamline data handling, transformation, and visualization. Its user-friendly interface, effective rule management, robust data mapping capabilities, and API integration make it a powerful tool for efficient and accurate data integration.

B. Architecture

To provide a comprehensive understanding of the PlayField framework, this section delves into its technical architecture, the technologies employed, and the overall data flow. The PlayField framework is constructed on a modular architecture designed to ensure scalability, flexibility, and maintainability, as shown in Figure 1. This section will detail the critical components of the framework, including the frontend, backend, database, and API, highlighting their roles and interactions. We will explore the specific technologies utilized, such as Streamlit for the frontend and PostgreSQL for the database, and examine the domain model that defines the core entities and relationships within the system. Furthermore, the workflow subsection will present a detailed step-by-step process of data transformation and integration, showcasing how the framework efficiently manages and processes complex data sets. Through this detailed examination, the architecture section aims to illustrate the robustness and versatility of the PlayField framework in real-world applications. The key components include:

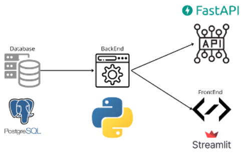

Fig. 1. PlayField Architecture

- **Frontend:** Developed using Streamlit (streamlit.io), providing an interactive and responsive user interface.
- **Backend:** Implemented in Python, handling the logic for data transformation and rule management.
- **Database:** PostgreSQL, used for storing rules, providers, and the input and output data.
- **API:** RESTful API built with Flask, enabling automated data processing and integration.

C. *Workflow*

The PlayField framework follows a structured workflow to ensure efficient and accurate data transformation. This workflow provide a step-by-step guide from accessing the application to exporting the transformed data. Below is a detailed description of each step involved in the process:

1) **Access the PlayField Interface:** The user opens the PlayField application built with Streamlit.
2) **Select Data Type:** The user selects the type of data to be processed (e.g., player statistics, match results, team data) from the dropdown menu.
3) **Select Provider:** The user selects the data provider from the dropdown menu. If the provider is not listed, the user can add a new provider by selecting the “Add New Provider” option and entering the provider’s name.
4) **Upload Input File:** The user uploads the input file (CSV, JSON, XML, etc.) via the file uploader component. This file contains the raw data from the selected provider.
5) **Define Transformation Rules:** The user defines transformation rules through the interactive user interface. This involves mapping input fields to the corresponding output fields.
6) **Set Hierarchical Rules:** If the input data contains nested structures (e.g., arrays within dictionaries), the user defines hierarchical rules to ensure all nested data is correctly transformed.
7) **Automatic Rule Saving:** As the user defines rules, they are automatically saved in the PostgreSQL database. This ensures that all rules are recorded and can be retrieved or modified as needed.
8) **Initiate Data Transformation:** The backend processes the input data according to the defined rules. The system applies each rule to transform the raw data into a standardized format previously defined.
9) **Export Transformed Data:** Once the data transformation is complete, the user can export the transformed data. This can be done in two ways: **Download as JSON File:** The user clicks the ”Export Transformed Input” button to download the transformed data as a JSON file or through the **API:** The user can use the *transform* endpoint to retrieve the transformed data programmatically.

III. Validation

Validating the framework is crucial to ensure its effectiveness, usability, and relevance. We employed a dual approach for comprehensive validation.

First, an experimental validation was conducted in a controlled environment to rigorously test the framework’s functionalities and performance. This phase helped identify and address potential issues, ensuring robustness and reliability.

Following this, we carried out an extensive case study with ZeroZero. In this real-world application, ZeroZero’s content managers and software developers tested the framework and provided feedback through a detailed questionnaire. This practical testing validated the framework’s usability and effectiveness under actual usage conditions.

This two-pronged strategy—combining experimental testing with real-world application—provides a thorough evaluation, confirming the framework’s technical soundness and practical benefits for end-users.

A. *Experimental Validation*

To rigorously evaluate the robustness and adaptability of the framework, comprehensive tests were conducted using data from multiple providers, including various iterations of the same providers. This approach assessed the framework’s ability to process and transform diverse input file formats and structures accurately. By introducing data from different sources and newer versions, we analyzed the framework’s effectiveness in managing variations in data schemas, ensuring consistency in transformation processes, and maintaining data integrity across disparate datasets. Initially, the framework displayed raw input and output files, requiring users to manually input the full path of fields for mapping. This approach proved too complex for non-technical content managers. Feedback from detailed meetings highlighted the need for a more intuitive solution, leading to significant changes. We developed a user-friendly interface with dropdown menus and visual aids, simplifying the rule creation process and improving user experience. **Testing with EnetPulse Files:** EnetPulse data was used to create and refine transformation rules. Initial manual rule creation was found cumbersome, prompting the introduction of an automation feature. This feature scans input and output files, automatically creating rules for matching fields, significantly streamlining the process. **Testing with Updated Versions:** Further tests with newer versions of data from the same provider, such as Monks files, evaluated the framework's adaptability to changes in data schemas. The automated rule creation feature ensured accurate transformations, maintaining consistency. **Enhancements Based on User Feedback:** Initially, the framework restricted rule creation to fields at the same depth level. Feedback indicated that this limitation hindered flexibility. Enhancements supported rule creation across different hierarchical levels, allowing users to map fields regardless of depth. Additional dropdown menus facilitated this process, improving usability and flexibility.

B. *Use Case*

To assess the importance and usability of the framework in transforming input files to a common structure, we conducted an evaluation with real users at ZeroZero, specifically targeting content managers. This rigorous validation process was designed to ensure the framework's effectiveness and usability in real-world scenarios, providing valuable insights and confirming its suitability for practical applications. For the validation purpose, we reached out to seven members of the content management department and six software developers from ZeroZero to participate in this validation phase.

Initially, we provided a detailed explanation of the main concept of the framework. This introduction was crucial for ensuring that all participants had a clear understanding of the framework's objectives and functionalities.

C. *Pre-Test Questionnaire*

The pre-test questionnaire aimed to collect baseline information about the participants to understand the demographic and professional background of the users. This section was essential for contextualizing the feedback and understanding the diversity of the participant pool.

- **Language Selection:** Participants were given the option to select their preferred language for the questionnaire (Portuguese or English) to ensure clarity and comfort in responding.
- **Consent:** Participants had to agree to provide personal information and details about their background. This consent was crucial for ethical research practices.
- **Demographic Information:** Basic information such as gender and age was collected to categorize the participants.
- **Professional Background:** Participants were asked about their current role, years of experience, as illustrated in Figure 2, and their level of expertise in data manipulation, as observed in Figure 3. These details helped in understanding the user profiles and their relevance to the framework.

D. *Task Execution*

This section focused on practical tasks that participants had to perform using the framework. The tasks were designed to evaluate the ease of use, intuitiveness, and effectiveness of the framework.

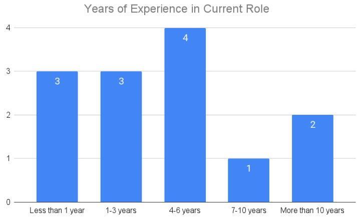

Fig. 2. Years of Experience in the Current Role

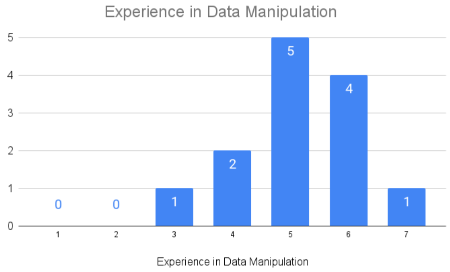

Fig. 3. Experience of each user in Data Manipulation

- **Task 1:** Participants were required to integrate and manipulate data from a specified provider using the framework. The objectives included ensuring proper mapping and transformation of data fields and refining the dataset by removing unnecessary fields.
- **Task 2:** After the initial data manipulation, participants were asked to automate the transformation process using the provided API. This task aimed to test the framework's ability to handle automated workflows and integration with other systems.

The tasks were designed to be completed within a 20-minute timeframe, ensuring that they were challenging yet achievable. Participants were encouraged to solve the tasks independently to maintain an unbiased evaluation.

E. *Post-Test Questionnaire*

Following the task execution, the post-test questionnaire aimed to gather detailed feedback on the framework's usability and overall user experience.

- **Usability Evaluation:** Participants rated their agreement with various statements about the system's usability on a scale from 'Strongly Disagree' to 'Strongly Agree.' The statements covered aspects such as the intuitiveness of the solution, the need for technical support, the complexity of the solution, and the learning curve.
- **Feedback on Specific Features:** Participants provided their subjective evaluations on whether the solution could

accelerate the data collection process and whether they would recommend the solution to colleagues.

- **Additional Comments:** An open-ended section allowed participants to share any additional feedback or comments about their experience with the framework. This section was vital for capturing qualitative data that could provide deeper insights into user perceptions and potential areas for improvement.

The usability evaluation statements used were:

Q1) I think that the solution is intuitive.

Q2) I think that I would need the support of a technical person to be able to use this solution.

Q3) I think that I would like to use this solution in my day-to-day work.

Q4) I found the solution unnecessarily complex.

Q5) I would imagine that most people would learn to use this solution very quickly.

Q6) I think this solution is able to accelerate the process of data collection.

Q7) I think that the learning curve for this solution is smooth and easy to navigate.

Q8) I would recommend this solution to a colleague.

The results are presented in Figure 4:

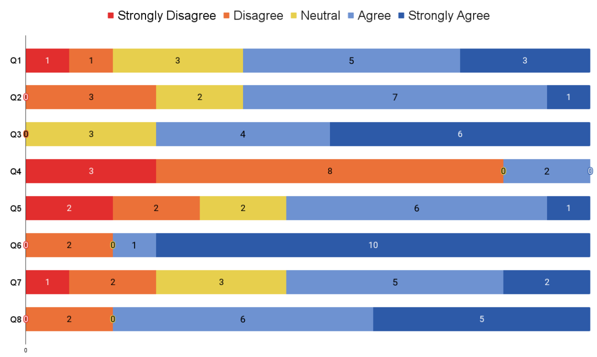

Fig. 4. SUS results from the questionnaire

The feedback from the validation process highlights several critical aspects of the framework's performance and user experience. Overall, the framework is perceived positively, with many users finding it intuitive and effective for their needs. The solution's utility in daily work is particularly appreciated, indicating that it meets a significant need for efficient data integration and transformation. However, some areas require attention to enhance the user experience further. Some users indicated a need for technical support, suggesting that the framework could benefit from additional user documentation, training sessions, and in-app support features. Simplifying the user interface and making it more intuitive will also help accommodate users with varying levels of technical expertise. The feedback also pointed out the importance of a smoother learning curve. Enhancing onboarding materials, such as interactive tutorials and mentorship programs, will facilitate quicker and easier adoption by new users. Ensuring the solution remains easy to use and accessible to non-technical users will be crucial for broader adoption and satisfaction. Efficiency in data collection is a strong point for the framework, with many users acknowledging its capability to streamline this process. To capitalize on this, further optimization through automation and advanced data processing algorithms can enhance productivity even more. Lastly, the overall satisfaction and willingness to recommend the framework to colleagues reflect a high level of user endorsement. Continuous engagement with users to gather feedback and implement improvements will be essential in maintaining this positive momentum and ensuring the framework remains relevant and valuable to its users.

In summary, the PlayField framework demonstrates considerable promise and utility, particularly in enhancing the efficiency of data collection and transformation processes. Addressing its limitations by enhancing user intuitiveness, simplifying complex aspects, and improving support resources will be crucial for maximizing its effectiveness and user satisfaction. By regularly engaging with users and iteratively improving the framework based on their feedback, PlayField can continue to be a valuable tool for data transformation and integration tasks.

IV. Related Work

Existing ETL tools, while rich in features, have limitations that restrict their applicability to our project's specific needs. This section briefly reviews notable ETL tools, such as Fivetran (www.fivetran.com/), Dataddo (www.dataddo.com/), Hevo Data (hevodata.com), Alooma (www.alooma.com), Talend (www.talend.com/), Informatica (docs.informatica.com/), IBM DataStage (www.ibm.com/products/datastage), and AWS Glue (aws.amazon.com/glue/), focusing on their constraints.

A. Limitations of Existing ETL Tools

Most commercial ETL tools, including Fivetran and Dataddo, excel in ease of use and scalability but lack the flexibility for complex data transformations. Tools like Hevo Data and Alooma offer real-time data processing but can be complex to configure and may struggle with large volumes of historical data. Talend and Informatica provide extensive customization options, yet their high setup and maintenance demands make them less ideal for rapid deployment. IBM DataStage, known for robust parallel processing, is prohibitively costly for smaller projects, while AWS Glue's reliance on the AWS ecosystem limits multi-cloud integration options.

B. Feature Comparison

We conducted a comparative analysis of key ETL tools based on features essential for successful ETL operations, such as Cloud Support, Parallel Processing, and Real-Time

Integration, summarized in Table I ('Y' indicates the presence of a feature in the ETL tool, and 'N' is its absence).

We established key features for a successful ETL solution based on [17], [18]. The key features are:

- Accept/Ignore Mechanisms (A/I);
- Real-Time Integration (RI);
- Real-Time Analysis (RA);
- Web-based UI (UI);
- Cloud Support (C);
- Non-RDBMS Connections (N);
- Metadata Management (M);
- Automation (A);
- Horizontal Scalability (S);
- Parallel Processing (P);
- Data Transformation (T);
- Large Volume Performance (VP);
- Human workflow of Error Handling (HE);
- Join Multiple Sources (MS);
- Data Partitioning (DP);

Feature	Informatica	IBM DataStage	AWS Glue	Talend	Alooma	Hevo Data	Fivetran	Dataddo
A/I	N	N	N	N	N	N	N	N
T	Y	Y	Y	Y	Y	Y	Y	Y
RI	Y	Y	Y	Y	Y	Y	Y	Y
RA	Y	Y	Y	Y	Y	Y	Y	Y
UI	Y	Y	Y	Y	N	Y	Y	Y
C	Y	Y	Y	Y	Y	Y	Y	Y
S	N	Y	N	N	N	Y	Y	Y
N	N	Y	Y	N	N	N	Y	Y
P	Y	Y	N	Y	N	N	N	Y
M	Y	Y	Y	N	N	Y	Y	Y
A	N	Y	Y	N	N	Y	Y	Y
VP	Y	Y	Y	Y	N	Y	Y	Y
HE	N	N	N	N	N	N	N	N
MS	Y	Y	Y	Y	Y	Y	Y	Y
DP	Y	Y	N	Y	Y	Y	Y	Y

TABLE I
COMPARISON OF ETL TOOLS

No single tool meets all our project requirements. To bridge these gaps, we propose a custom ETL framework that combines the automation strengths of Fivetran and Dataddo with advanced data integration features. This tailored solution aims to meet the diverse and dynamic needs of modern data integration.

V. CONCLUSIONS

The landscape of sports analytics has evolved significantly, transitioning from basic statistical analysis to a more comprehensive use of intricate data, including detailed metrics of player performance and extensive information from amateur leagues. This expansion has introduced increased complexity and diversity in data sources, necessitating advanced integration methods to maintain coherence and value in sports analytics. Traditional data integration methods often fall short in handling the volume, velocity, and variety of modern sports data, highlighting the need for a robust and adaptable ETL process tailored to these unique demands.

In response to these challenges, we present in this paper Playfield, a framework designed to robustly handle diverse sports data through adaptable configuration and an automated API. PlayField ensures precise data integration and supports manual interventions for data integrity, making it essential for accurate and comprehensive sports analysis.

To validate the framework a case study was conducted with ZeroZero, the world's largest sports-related information database, focusing on football. The results demonstrated the framework's capability to improve data integration efficiency significantly, showcasing its potential for advanced analytics in sports.

Future work should enhance the tool's capabilities by incorporating advanced machine learning algorithms for automatic data mapping, reducing manual effort and improving accuracy. Developing sophisticated data transformation algorithms and enhancing the user interface will further improve user experience. Continuous feedback and research will ensure the tool remains valuable for both academic and industry applications, driving ongoing improvement and innovation.

REFERENCES

[1] Seref Sagiroglu and Duygu Sinanc. Big data: A review. Proceedings of the 2013 Inter- national Conference on Collaboration Technologies and Systems, 2013.

[2] Shaikh Abdul Hannan. An overview on big data and hadoop. International Journal of Computer Applications, 2016.

[3] Sreemathy, K Naveen Durai, E Lakshmi Priya, R Deebika, K Suganthi, and PT Aisshwarya. Data integration and etl: A theoretical perspective. In 2021 7th International Conference on Advanced Computing and Communication Systems (ICACCS), 2021.

[4] Marek Macura. Integration of data from heterogeneous sources using etl technology. Com- puter Science, 2014.

[5] Shaker H. Ali El-Sappagh, Abdeltawab M. Ahmed Hendawi, and Ali Hamed El Bastawissy. A proposed model for data warehouse etl processes. Journal of King Saud University - Computer and Information Sciences, 2011.

[6] J. Sreemathy, Infant Joseph V., S. Nisha, Chaaru Prabha I., and Gokula Priya R.M. Data integration in etl using talend. In 2020 6th International Conference on Advanced Computing and Communication Systems (ICACCS), 2020.

[7] Wayne Yaddow. The process of data mapping for data integration projects. 10 2019.

[8] George P. Fletcher. The data mapping problem: Algorithmic and logical characterizations, 2005.

[9] Qingzhao Tan, Prasenjit Mitra, and C. Lee Giles. Metadata extraction and indexing for map search in web documents. In Proceedings of the 17th ACM Conference on Information and Knowledge Management, CIKM '08, 2008.

[10] Elizabeth M. Silvers. The Data Asset: How Smart Companies Govern Their Data for Busi- ness Success, 2012.

[11] Thomas C. Redman. Data Driven: Profiting from Your Most Important Business Asset.

[12] Harvard Business Review Press, 2008. David Loshin. Master Data Management. Morgan Kaufmann, 2010.

[13] Ralph Kimball and Margy Ross. The Data Warehouse Toolkit: The Definitive Guide to

[14] Dimensional Modeling. John Wiley & Sons, 2013. William H Inmon, Derek Strauss, and Genia Neushloss. DW 2.0: The Architecture for the Next Generation of Data Warehousing. Morgan Kaufmann, 2010.

[15] Thomas H Davenport and Jeanne G Harris. Competing on Analytics: The New Science of Winning. Harvard Business Review Press, 2007.

[16] Wayne W Eckerson. Performance Dashboards: Measuring, Monitoring, and Managing Your Business. John Wiley & Sons, 2011.

[17] Asma Qaiser, Muhamamd Umer Farooq, Syed Muhammad Nabeel Mustafa, and Nazia Abrar. Comparative analysis of etl tools in big data analytics. Pakistan Journal of Engi- neering and Technology, 2023.

[18] Rajendrani Mukherjee and Pragma Kar. A comparative review of data warehousing etl tools with new trends and industry insight. In 2017 IEEE 7th International Advance Computing Conference (IACC), 2017.