International Journal of
DATA SCIENCE AND IOT MANAGEMENT SYSTEM
Peer Reviewed, Referred & Indexed Journal
ISSN: 3068-272X www.ijdim.com Original Research Paper

Dynamic IPL Analysis Dashboard with Automated Data Refresh

Lingam Dhanya Sri
PG MCA Scholar
Department of Computer Science
Sir C. R. Reddy College, Eluru
Andhra Pradesh, India - 534005
dhanyalingam@gmail.com

Katragadda Sandhya Rani
Assistant Professor
Department of Computer Science
Sir C. R. Reddy College, Eluru
Andhra Pradesh, India - 534005
sandhyakiran@sircrreddycollege.ac.in

Musunuru Ratnakar
HOD
Department of Computer Science
Sir C. R. Reddy College, Eluru
Andhra Pradesh, India - 534005

Abstract— The Indian Premier League produces large volumes of ball-by-ball and match-level data, but raw files and fixed statistics pages do not provide flexible, cross-filtered analysis. This paper presents a dynamic Power BI dashboard covering IPL seasons from 2008 to 2026. Historical delivery and match records from Cricsheet are combined with player and team dimensions, while the CricketData Series Search REST endpoint supports automated acquisition of current series metadata. Power Query performs extraction, cleaning, standardization, and merging; a related semantic model supports more than forty Data Analysis Expressions measures; and three linked pages provide overview, batter, and bowler analysis. The report includes season KPIs, cap-holder cards, a points table, player trends, boundary distributions, dismissal distributions, venue and opposition comparisons, page navigation, and scheduled refresh. Twelve functional, integrity, interactivity, and refresh tests passed, including a manual API-driven update completed in under two minutes. The system demonstrates an end-to-end self-service sports-analytics workflow that converts granular cricket data into an accessible and maintainable decision-support dashboard.

Keywords— IPL analytics, Power BI, Power Query, DAX, business intelligence, automated refresh, sports dashboard.

I. INTRODUCTION

The Indian Premier League (IPL) has generated a continuously expanding record of deliveries, innings, matches, players, teams, venues, and results since 2008. Although the data has substantial analytical value, it is commonly distributed as CSV, JSON, or website tables that require manual interpretation. A user who wants to compare a batter across seasons, inspect a bowler by venue, or verify a points table often moves among several websites and static pages rather than using one coherent analytical model [1], [5]–[7].

The project addresses this limitation by combining historical ball-by-ball depth with an automated refresh path. Power Query is used to prepare the data, a star-schema-like model relates delivery events to match, player, and team context, and DAX measures calculate batting, bowling, and team indicators. The report pages expose these calculations through slicers, cards, charts, tables, buttons, and cross-highlighting, enabling non-technical users to obtain results without writing SQL, Python, or DAX [3], [4], [8]–[11].

The principal contributions are fourfold: (i) consolidation of four complementary IPL datasets; (ii) reusable measures for strike rate, averages, economy, wickets, boundaries, points, and season summaries; (iii) a consistent three-page interface for overview, batter, and bowler analysis; and (iv) an automated REST-based refresh mechanism that preserves the last successfully loaded model when the external service is unavailable.

II. RELATED WORK

ESPNcricinfo Statsguru, Cricbuzz, and the official IPL portal provide scorecards, leaderboards, points tables, and historical records. These services are authoritative and convenient, but their layouts are predefined and they do not permit users to redesign measures, connect custom dimensions, or cross-filter all visuals through a self-service semantic model [5]–[7]. Cricsheet provides reproducible structured match files and is therefore suitable for event-level analysis, while CricketData provides hosted JSON endpoints that complement historical files with current series and match information [1], [2].

Dimensional modelling separates granular facts from descriptive dimensions, reducing duplication and producing predictable filter propagation [8], [13]. Power BI extends this approach with Power Query, DAX, interactive visuals, bookmarks, scheduled refresh, and cloud publication [3], [4], [10]–[16]. Dashboard-design literature also emphasizes visual hierarchy, direct comparison, context-preserving interaction, and limited clutter [17]–[21]. These principles guided the repeated navigation panel, top-level slicers, KPI grouping, and coordinated charts in the proposed dashboard.

Cricket analytics research has developed statistical measures for batting and bowling, player ranking, optimal batting order, in-play forecasting, and machine-learning prediction [23]–[30]. The present work is different from a match-outcome classifier: it focuses on descriptive and diagnostic business intelligence. Consequently, accuracy, precision, recall, and F1-score are not applicable to the current implementation; correctness is established through value reconciliation, functional tests, interaction response, and refresh reliability.

III. MATERIALS AND METHODS

The proposed system follows an acquisition–transformation–model–presentation architecture. Historical match files and live series metadata enter Power Query, are standardized into related tables, and are summarized by DAX before being displayed through interactive report pages. Fig. 1 shows the end-to-end pipeline reproduced from the project report.

International Journal of
DATA SCIENCE AND IOT MANAGEMENT SYSTEM
Peer Reviewed, Referred & Indexed Journal
ISSN: 3068-272X www.ijdim.com Original Research Paper

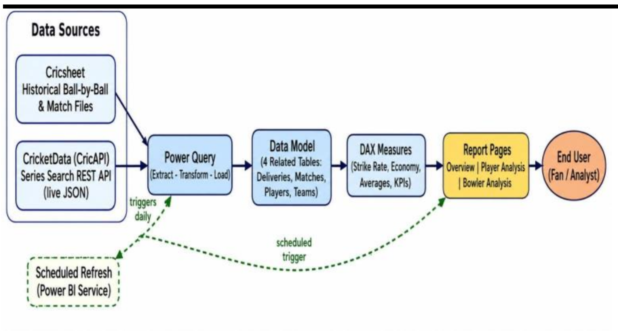

Fig. 1. System architecture and automated refresh pipeline.

A) Data Collection

i) Historical Match Data: Cricsheet files provide ball-level records such as match identifier, innings, over, ball, batter, bowler, non-striker, runs off the bat, extras, wicket type, dismissed player, and fielder. Match metadata supplies season, date, venue, city, teams, toss, result, and winner [1]. ii) Live Series Data: The CricketData Series Search endpoint returns JSON metadata for the latest IPL series and match list. This source is queried through the Power BI Web connector and is used to keep the current-season layer from becoming stale [2], [11].

iii) Player Dimension: Player identifiers, full names, short names, role attributes, team mappings, and image URLs enable profile cards and player slicers. iv) Team Dimension: Team identifiers, official names, short codes, logos, and branding fields support champion, runner-up, cap-holder, and points-table visuals.

Table 1. Dataset Structure and Analytical Role

Dataset	Representative fields	Role
Deliveries	match, ball, batter, bowler, runs, wicket	Fact
Matches	season, date, venue, teams, winner	Dimension
Players	player id, name, team, image URL	Dimension
Teams	team id, code, logo, colour	Dimension

B) Data Pre-processing

Power Query applies a repeatable ETL sequence. File schemas are aligned, data types are assigned, date fields are standardized, renamed franchises are mapped to common labels, blank values are inspected, duplicate rows are removed, and player or team identifiers are normalized. The live JSON response is expanded into tabular form and merged with the historical match table. Image fields are assigned a placeholder rule so that a missing URL does not break a report page.

The data-quality controls include uniqueness checks for match identifiers, row-count comparisons before and after transformations, validation of six-ball overs after excluding wides and no-balls, verification that bowler wickets exclude run-outs, and reconciliation of season-level totals with dashboard cards. The transformation steps are retained in Power Query so that a future season can be loaded by refreshing rather than redesigning the report.

C) Data Model

The semantic model uses one-to-many relationships between descriptive tables and the delivery fact table. Match filters propagate to deliveries by match_id; player filters identify the selected batter or bowler; and team context supports logos, points, opposition analysis, and franchise summaries. The live API table enriches the match layer, while a small boundary-type table supports controlled category ordering. Fig. 2 shows the implemented relationship view.

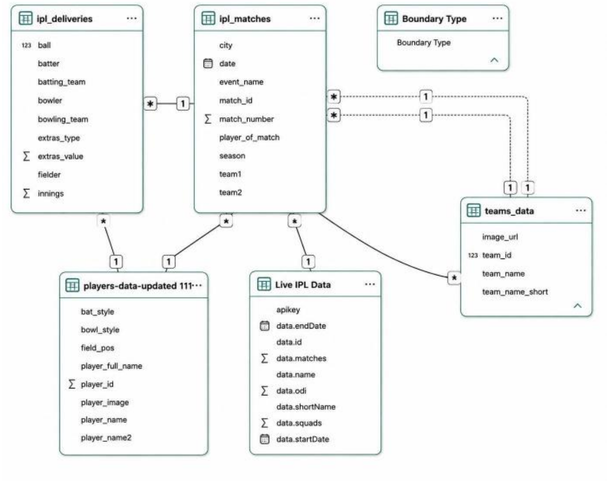

Fig. 2. Power BI data model and table relationships.

D) Data Flow and Process Logic

The Level-1 data-flow view separates external sources, transformation processes, persistent stores, model computation, and user-facing visualization. Cricsheet files enter Process 1.0 for extraction and cleaning, while the live REST response enters Process 2.0 for current-series refresh. The resulting deliveries, matches, players, and teams are stored independently and then consumed by Process 3.0, where relationships and DAX measures are evaluated. Process 4.0 renders the report and returns filtered results to the end user.

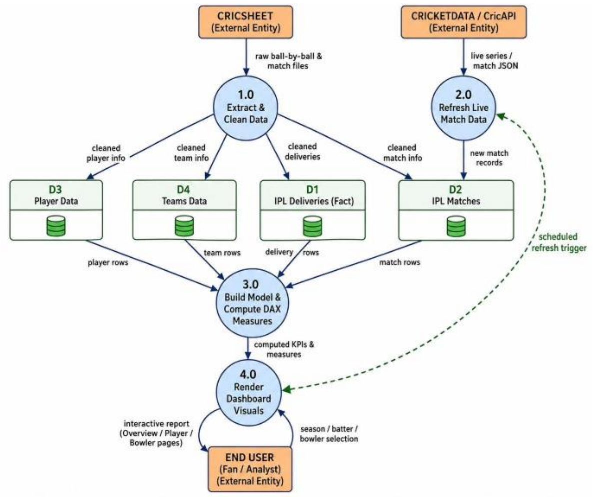

Fig. 3. Level-1 data-flow diagram of the dashboard system.

This separation supports reliability and traceability. A failure in the live-source branch does not invalidate historical tables, while a transformation error can be isolated before model loading. User selections travel only between the presentation layer and the loaded semantic model; they do not repeatedly call the external API. The design therefore reduces latency, avoids unnecessary requests, and makes the refresh pipeline easier to test.

E) Analytical Measures

The dashboard contains more than forty headline and helper measures. The following editable equations define the principal cricket indicators used in the visual layer:

$$\textit{Strike Rate} = (\textit{Total Runs} / \textit{Balls Faced}) \times 100 \quad (1)$$

$$Batting\ Average = Total\ Runs\ /\ Times\ Out \quad (2)$$

$$Economy\ Rate = Runs\ Conceded\ /\ Overs\ Bowled \quad (3)$$

$$Bowling\ Average = Runs\ Conceded\ /\ Wickets \quad (4)$$

$$Bowling\ Strike\ Rate = Balls\ Bowled\ /\ Wickets \quad (5)$$

$$NRR = Run\ Rate\ Scored - Run\ Rate\ Conceded \quad (6)$$

DAX DIVIDE is used for ratio measures to return a controlled result when the denominator is zero [4], [9]. Measures are evaluated in the current filter context, so the same formula supports an all-season career view, a selected season, a venue, an opposition team, or a combination of slicers.

IV. SYSTEM IMPLEMENTATION

A) ETL and DAX Implementation

The historical match files are consolidated into one delivery table and one match table. Player and team references are loaded as dimensions, and the CricketData response is requested through the Web connector. Power Query M steps expand records and lists, select required fields, cast dates and numeric columns, standardize franchise names, and append or merge compatible data. After loading, DAX measures calculate totals and rates from the ball-by-ball records rather than relying on pre-aggregated statistics.

Representative measures include Total Runs, Balls Faced, Fours, Sixes, Highest Score, Fifties, Hundreds, Times Out, Wickets, Runs Conceded, Balls Bowled, Overs Bowled, Economy Rate, Bowling Average, Bowling Strike Rate, Dot Balls, Matches Played, Team Wins, Team Ties, Total Points, Net Run Rate, season centuries, and cap-holder ranks. Helper measures reduce repeated logic and make the semantic model easier to maintain.

Table 2. Representative DAX Measures

Measure	Simplified DAX logic	Purpose
Total Runs	SUM(runs_off_bat)	Batter runs
Strike Rate	DIVIDE(runs, balls) x 100	Runs/100 balls
Wickets	COUNTROWS(wickets excluding run-out)	Bowler wickets
Economy	DIVIDE(runs conceded, overs)	Runs per over
Total Points	wins x 2 + ties	Points table
Dot Balls	runs=0 and extras=0	Pressure metric

B) Automated Refresh

A scheduled refresh job initiates the Power Query pipeline. The query requests the latest series metadata, transforms the response, loads updated tables, and triggers recalculation of dependent DAX measures. User interaction is deliberately decoupled from the live API: report pages always query the most recently loaded model, so a temporary API delay affects the next refresh but does not block slicers or navigation. Fig. 4 presents the sequence of background refresh and foreground interaction.

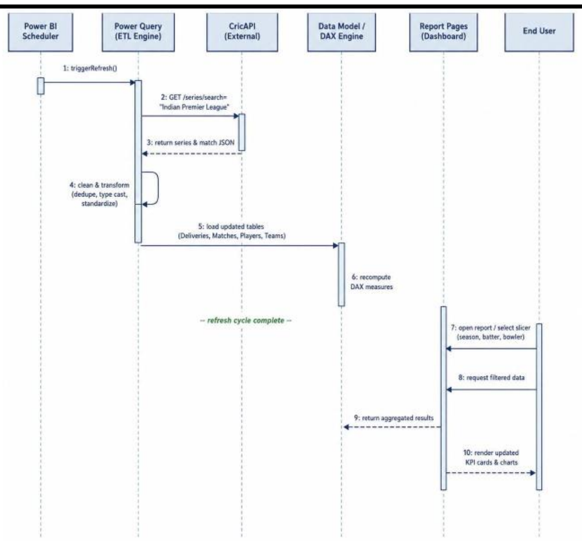

Fig. 4. Automated refresh and user-interaction sequence.

C) Dashboard and Navigation Design

The home page provides three direct actions—Overview, Player Analysis, and Bowler Analysis—through large navigation buttons. A consistent dark-blue IPL theme, repeated left-side navigation, aligned slicers, and standard card styling make the interface predictable. External resource icons open the official IPL site, Cricbuzz, and social platforms in a new browser tab. The landing page shown in Fig. 5 separates navigation from analysis and gives the report an application-like entry point.

Fig. 5. Dashboard home page and navigation interface.

D) Report Pages

The Overview page combines champion and runner-up cards, season totals, Orange and Purple Cap panels, highest-four and highest-six cards, and a sortable points table. The Batter page provides career or season totals, strike rate, average, boundaries, highest score, fifties, hundreds, season trend, scoring-shot distribution, opposition comparison, and venue comparison. The Bowler page provides wickets, dot balls, economy, average, strike rate, season trend, dismissal distribution, opposition comparison, and venue comparison. All visuals respond to the active slicer and to selections made in other charts.

Power BI cards are used for single-value KPIs; line charts show runs or wickets by season; donut charts show boundary and dismissal composition; clustered bars compare opposition and venue results; and table visuals display points and standings. Image URLs are categorized as image fields so that player and team photographs change automatically with filter context.

International Journal of
DATA SCIENCE AND IOT MANAGEMENT SYSTEM
Peer Reviewed, Referred & Indexed Journal
ISSN: 3068-272X www.ijdim.com Original Research Paper

E) Requirement-to-Implementation Mapping

The implementation was checked against the functional and non-functional requirements defined in the project report. Each requirement was linked to a visible component, a measure or query dependency, and a test condition. This mapping prevents a visually attractive dashboard from being accepted without verifying data integrity, reliability, and maintainability.

Table 3. Requirement-to-Implementation Mapping

Requirement	Implemented feature	Verification
Season analysis	Season slicer and KPI cards	TC-01/02
Batter analysis	Batter slicer and trends	TC-03/04
Bowler analysis	Bowler slicer and dismissals	TC-05/06
Interactivity	Cross-filtering and buttons	TC-07/09
Automated refresh	Web query and schedule	TC-10/11
Fault handling	Image fallback and cached model	TC-12

Usability was addressed through repeated placement of slicers and navigation controls, performance through a related model and reusable measures, maintainability through parameterized queries and dimension mappings, and reliability through cached data and fallback behavior. The resulting design connects system requirements directly to observable test evidence rather than relying on subjective visual inspection alone.

V. EXPERIMENTAL RESULTS

A) Season Overview Output

The selected 2026 dashboard snapshot identifies Royal Challengers Bangalore as champion and Gujarat Titans as runner-up. It displays 1,426 sixes, 2,334 fours, 74 matches, 16 teams, 15 centuries, 154 half-centuries, and 13 venues. The same page shows V. Suryavanshi as Orange Cap holder with 776 runs, K. Rabada as Purple Cap holder with 29 wickets, B. Sai Sudharsan with 75 fours, and V. Suryavanshi with 72 sixes. These values are presented as project output from the attached dashboard dataset and can be changed through the season slicer.

Fig. 6. Overview page with KPIs, cap holders, and points table.

B) Batter Analysis Output

For V. Kohli with Season set to All, the dashboard reports 9,346 runs from 275 matches, a strike rate of 131.14, 847 fours, 317 sixes, a highest score of 113, 69 fifties, 9 hundreds, and a batting average of 41. The opposition comparison shows 1,174 runs against Chennai Super Kings, while the venue comparison shows 1,874 runs at M. Chinnaswamy Stadium. The scoring distribution totals approximately 100% after rounding: 64.17% singles, 18.22% fours, 10.80% doubles, and 6.82% sixes.

Fig. 7. Batter analysis page for V. Kohli.

C) Bowler Analysis Output

For Mohammed Siraj with Season set to All, the report displays 132 wickets, 1,111 dot balls, an economy rate of 8.48, a bowling average of 30.33, and a bowling strike rate of 21.47. The visual breakdown identifies caught as the dominant dismissal type, followed by bowled and leg-before-wicket. Opposition and venue charts show 19 wickets against Kolkata Knight Riders and 23 wickets at M. Chinnaswamy Stadium.

Fig. 8. Bowler analysis page for Mohammed Siraj.

D) Consolidated KPI Summary

Table 4. Selected Dashboard Outputs

Context	Indicator	Displayed value
2026	Matches / teams	74 / 16
2026	Fours / sixes	2334 / 1426
V. Kohli	Runs / strike rate	9346 / 131.14
V. Kohli	100s / 50s	9 / 69
M. Siraj	Wickets / dot balls	132 / 1111
M. Siraj	Economy / average	8.48 / 30.33

The consolidated values show the range of the semantic model: one report context produces tournament totals, another produces long-term batting measures, and another produces bowling efficiency measures. The same tables and formulas are reused; only the active filter context changes.

E) Result Interpretation

The three outputs demonstrate that one semantic model can answer season-level, batting, and bowling questions without duplicating datasets or calculations. Filter context changes both headline cards and supporting visuals, while cross-highlighting provides a direct path from a broad summary to a venue or opposition-specific explanation. The project therefore converts event-level records into a compact analytical narrative rather than displaying isolated statistics.

International Journal of
DATA SCIENCE AND IOT MANAGEMENT SYSTEM
Peer Reviewed, Referred & Indexed Journal
ISSN: 3068-272X www.ijdim.com Original Research Paper

VI. VALIDATION AND DISCUSSION

A) Functional and Data Validation

Twelve tests were executed on the final PBIX build and, where applicable, on the Power BI Service version. The tests covered overview KPIs, points-table row count and sorting, batter values, boundary-distribution totals, bowler values, dismissal totals, cross-filtering, navigation, external links, API-driven refresh, scheduled refresh, and missing-image fallback. Every test passed. Fig. 9 summarizes the four test groups.

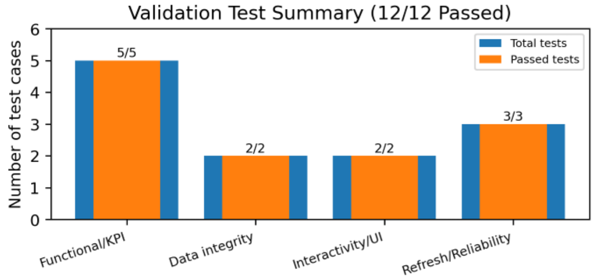

Fig. 9. Combined validation chart: 12 of 12 tests passed.

Table 5. Representative Validation Results

B) Refresh and Interaction Performance

With the complete deliveries table loaded for seasons 2008–2026, a full refresh completed within the Power BI Service allowance. The external API usually returned within a few seconds, while the larger share of elapsed time was spent transforming the delivery fact table. No failures were observed during the reported testing window, and refresh history consistently showed Success. Cross-filter actions and page navigation produced no visible delay during manual testing; the design target was approximately one to two seconds for slicer-driven visual updates.

C) Discussion

The dashboard achieves its primary objective: a single interactive report now combines historical depth, current-series enrichment, reusable measures, and multi-page navigation. The related model avoids calculating the same metric independently on every page, and the API is isolated from direct user interaction. This architecture improves maintainability because the next season can be incorporated by refreshing the query and extending dimension mappings rather than recreating visuals.

A classifier would be judged by predictive metrics, but this project is judged by whether filters return correct values, distributions reconcile, navigation works, refresh completes, and failures are handled gracefully. The test evidence shows that these requirements were met. The use of project-supplied 2026 outputs should nevertheless be understood as a dashboard snapshot; independent real-world verification depends on the quality and timing of upstream data.

D) Limitations and Future Work

The free API tier limits request frequency; player and team images require mapping maintenance; older source files may contain missing location metadata; and public sharing depends on Power BI licensing. The present system is descriptive and does not estimate match outcome, player form, or win probability. Future work can introduce incremental refresh, data-quality alerts, automated image lookup, a side-by-side player comparison page, fantasy-point calculation, ball-by-ball replay, mobile layouts, natural-language Q&A, and expansion to other T20 leagues.

A predictive extension may add logistic regression, gradient boosting, or LSTM models trained with season-wise holdout validation. Such a model should report accuracy, precision, recall, F1-score, calibration, and temporal generalization separately from the current descriptive dashboard, so that predictive uncertainty is not confused with verified historical statistics [27]–[30].

E) Scalability and Maintainability

The solution is designed to grow by appending new match rows rather than redesigning the report. Delivery records remain in the fact table, while teams, players, venues, and seasons are handled through reusable dimensions. New-season support therefore requires refresh and mapping updates, not a new set of visuals. Central DAX measures also prevent formula drift: a correction to strike rate, economy, or points logic is made once and is inherited by every card, table, and chart that uses the measure. Query steps are arranged in a repeatable order so that schema alignment, type conversion, duplicate removal, and API expansion can be inspected independently. This separation reduces maintenance effort and makes failures easier to localize.

For larger historical archives, incremental refresh and partitioned storage can restrict processing to recent seasons while preserving older records. The same semantic model can also be published as a shared dataset for mobile reports, team-comparison pages, or management summaries. Because user interaction queries the loaded model instead of the live API, report responsiveness remains stable even when the external service is slow or temporarily unavailable.

F) Practical Significance and Use Cases

The dashboard supports several practical user groups. Fans can compare players and teams without manually combining statistics from multiple websites. Sports journalists can obtain season summaries, opposition breakdowns, and venue trends for rapid story preparation. Coaches and analysts can identify long-term scoring patterns, boundary dependence, bowling efficiency, and dismissal composition. Fantasy-sports users can inspect recent and historical player output before team selection, while students can study a complete business-intelligence workflow that links public data, ETL, modelling, calculation, visualization, testing, and deployment.

The project is also a reusable template for other event-based domains. Replacing cricket deliveries with transactions, sensor readings, or service events would preserve the same architectural pattern: acquire detailed records, standardize them, connect them to descriptive dimensions, define governed measures, and expose interactive views. Thus, the contribution is not limited to IPL statistics; it demonstrates an end-to-end method for converting continuously growing operational data into maintainable self-service analytics.

G) Threats to Validity and Reproducibility

The reported values depend on the completeness and consistency of Cricsheet files, CricAPI responses, and manually maintained player or team mappings. A delayed upstream update, renamed franchise, missing venue, or inconsistent player spelling can temporarily affect totals until the transformation rules are updated. Visual agreement with external websites is useful but is not by itself proof of

International Journal of
DATA SCIENCE AND IOT MANAGEMENT SYSTEM
Peer Reviewed, Referred & Indexed Journal
ISSN: 3068-272X www.ijdim.com Original Research Paper

correctness because different sources may apply different definitions for legal balls, run-outs, retired-hurt innings, or historical franchise identities. The project therefore relies on reconciled totals, explicit wicket exclusions, test cases, and repeatable DAX formulas rather than visual inspection alone.

Reproducibility is improved by keeping source tables, Power Query steps, relationships, measures, test inputs, and expected outputs within the documented project workflow. A future release should additionally preserve dataset versions, API timestamps, PBIX version identifiers, and refresh logs so that a reported snapshot can be regenerated exactly. Independent validation against official season records would further strengthen external validity, especially for the 2026 snapshot contained in the submitted project report.

H) Comparison with Existing Platforms

Conventional cricket portals are optimized for publishing scores, commentary, news, and predefined record tables. They provide authoritative coverage, but their analytical path is largely fixed by the publisher. The proposed dashboard does not replace those services; it complements them with a user-controlled semantic layer. A single selection can simultaneously update headline KPIs, trend lines, venue comparisons, opposition comparisons, cap-holder cards, and the points table. This coordinated filtering is the principal advantage over navigating separate static pages. In addition, the underlying measures remain visible and editable, allowing an analyst to audit or adapt the calculation logic rather than accepting an opaque total.

Compared with spreadsheet-only analysis, Power BI offers stronger relationship management, filter propagation, reusable measures, and report navigation. Compared with a custom-coded web dashboard, it reduces development effort for visual interaction and scheduled refresh, although it introduces licensing and platform-dependency constraints. The selected approach is therefore appropriate for an academic self-service analytics project whose priorities are rapid development, traceable calculations, interactive exploration, and maintainable refresh.

I) Data Governance and Responsible Use

The datasets describe public sporting events and publicly presented player statistics, but responsible use still requires source attribution, version control, and clear distinction between observed data and inferred claims. Dashboard values should display the active season and filter context so that users do not quote a career total as a single-season result. Refresh timestamps should be visible when current-series information is presented, and missing or incomplete records should be flagged rather than silently interpreted as zero. Image URLs should be sourced from permitted locations and replaced with neutral placeholders when unavailable.

Any future predictive feature should avoid presenting probability as certainty. Fantasy selection, coaching assessment, and player comparison can be influenced by injuries, role changes, pitch conditions, and selection decisions that are not represented in historical ball-by-ball data. Model outputs should therefore include uncertainty, evaluation period, and known limitations. These governance practices preserve the dashboard as a decision-support tool rather than an unquestioned decision maker.

J) Deployment Recommendations

For reliable deployment, the PBIX file, source schema, query parameters, DAX catalogue, and validation checklist should be versioned together. Refresh should run after upstream match publication, retain the last successful model, and generate failure notifications. Schema changes should be tested in a development workspace before public release, while API keys and gateway credentials remain outside visible report content. Periodic reconciliation against official season totals will help detect upstream changes and maintain user trust.

VII. CONCLUSION

This paper presented an IPL analytics solution combining data acquisition, Power Query ETL, dimensional modelling, DAX calculation, interactive reporting, and automated refresh. The common semantic model supports overview, batter, and bowler analysis, contains more than forty measures, and passed all twelve reported tests. The resulting self-service dashboard reduces manual preparation and enables direct visual exploration of cricket data.

REFERENCES

[1] Cricsheet, “Ball-by-ball data for international and T20 franchise cricket matches,” open cricket-data repository, 2026. [Online]. Available: https://cricsheet.org. Accessed: Jul. 23, 2026.

[2] CricketData.org, “CricAPI cricket data API documentation: series search, match lists, scorecards, and player information,” 2026. [Online]. Available: https://cricketdata.org. Accessed: Jul. 23, 2026.

[3] Microsoft, “Power BI documentation: data acquisition, Power Query, semantic modelling, visualization, publishing, and administration,” Microsoft Learn, 2026. [Online]. Available: https://learn.microsoft.com/power-bi/. Accessed: Jul. 23, 2026.

[4] Microsoft, “Data Analysis Expressions (DAX) function reference,” Microsoft Learn, 2026. [Online]. Available: https://learn.microsoft.com/dax/. Accessed: Jul. 23, 2026.

[5] Indian Premier League, “Official IPL statistics: points table, batting records, Orange Cap, Purple Cap, and season summaries,” 2026. [Online]. Available: https://www.iplt20.com/stats/. Accessed: Jul. 23, 2026.

[6] ESPNcricinfo, “Statsguru: advanced cricket records and statistical query engine,” 2026. [Online]. Available: https://stats.espncricinfo.com. Accessed: Jul. 23, 2026.

[7] Cricbuzz, “Indian Premier League live scores, schedules, news, scorecards, and player statistics,” 2026. [Online]. Available: https://www.cricbuzz.com. Accessed: Jul. 23, 2026.

[8] R. Kimball and M. Ross, The Data Warehouse Toolkit: The Definitive Guide to Dimensional Modeling, 3rd ed. Hoboken, NJ, USA: Wiley, 2013. This work defines fact tables, dimensions, star schemas, and reusable analytical models.

[9] A. Ferrari and M. Russo, The Definitive Guide to DAX: Business Intelligence for Microsoft Power BI, SQL Server Analysis Services, and Excel, 2nd ed. Redmond, WA, USA: Microsoft Press, 2019.

[10] Microsoft, “Data refresh in Power BI, scheduled refresh, refresh history, credentials, and gateway configuration,” Microsoft Learn, 2026. [Online]. Available: https://learn.microsoft.com/power-bi/connect-data/refresh-data. Accessed: Jul. 23, 2026.

[11] Microsoft, “Web connector and JSON data handling in Power Query M,” Microsoft Learn, 2026. [Online]. Available: https://learn.microsoft.com/power-query/connectors/web/web. Accessed: Jul. 23, 2026.

[12] Board of Control for Cricket in India and Indian Premier League, “Official tournament rules, teams, fixtures, results, and

International Journal of
DATA SCIENCE AND IOT MANAGEMENT SYSTEM
Peer Reviewed, Referred & Indexed Journal
ISSN: 3068-272X www.ijdim.com Original Research Paper

records," 2026. [Online]. Available: https://www.iplt20.com. Accessed: Jul. 23, 2026.

[13] Microsoft, "Understand star schema and its importance for Power BI semantic models," Microsoft Learn, 2026. [Online]. Available: https://learn.microsoft.com/power-bi/guidance/star-schema. Accessed: Jul. 23, 2026.

[14] Microsoft, "Create and manage relationships in Power BI Desktop," Microsoft Learn, 2026. [Online]. Available: https://learn.microsoft.com/power-bi/transform-model/desktop-create-and-manage-relationships. Accessed: Jul. 23, 2026.

[15] Microsoft, "Best practices when working with Power Query," Microsoft Learn, 2026. [Online]. Available: https://learn.microsoft.com/power-query/best-practices. Accessed: Jul. 23, 2026.

[16] Microsoft, "Query folding guidance and transformation optimization in Power Query," Microsoft Learn, 2026. [Online]. Available: https://learn.microsoft.com/power-query/power-query-folding. Accessed: Jul. 23, 2026.

[17] Microsoft, "Optimization guidance for Power BI data models, DAX queries, visuals, and report performance," Microsoft Learn, 2026. [Online]. Available: https://learn.microsoft.com/power-bi/guidance/power-bi-optimization. Accessed: Jul. 23, 2026.

[18] S. Few, Information Dashboard Design: Displaying Data for At-a-Glance Monitoring, 2nd ed. Burlingame, CA, USA: Analytics Press, 2013. The book presents principles for compact, readable, and decision-oriented dashboard design.

[19] T. Munzner, Visualization Analysis and Design. Boca Raton, FL, USA: CRC Press, 2014. The text provides a systematic framework for matching data abstractions, analytical tasks, visual encodings, and interaction techniques.

[20] B. Shneiderman, "The eyes have it: A task by data type taxonomy for information visualizations," in Proc. IEEE Symp. Visual Languages, Boulder, CO, USA, 1996, pp. 336–343, doi: 10.1109/VL.1996.545307.

[21] J. Heer, M. Bostock, and V. Ogievetsky, "A tour through the visualization zoo," Commun. ACM, vol. 53, no. 6, pp. 59–67, Jun. 2010, doi: 10.1145/1743546.1743567.

[22] D. A. Keim, F. Mansmann, J. Schneidewind, J. Thomas, and H. Ziegler, "Visual analytics: Scope and challenges," in Visual Data Mining, S. J. Simoff, M. H. Böhlen, and A. Mazeika, Eds. Berlin, Germany: Springer, 2008, pp. 76–90.

[23] A. C. Kimber and A. R. Hansford, "A statistical analysis of batting in cricket," J. Roy. Stat. Soc. A, vol. 156, no. 3, pp. 443–455, 1993. The study illustrates statistical treatment of batting performance and scoring variability.

[24] G. R. Amin and S. K. Sharma, "Measuring batting parameters in cricket: A two-stage regression-OWA method," Measurement, vol. 53, pp. 56–61, Jul. 2014, doi: 10.1016/j.measurement.2014.03.027.

[25] M. J. Bailey and S. R. Clarke, "Predicting the match outcome in one day international cricket matches, while the game is in progress," J. Sports Sci. Med., vol. 5, no. 4, pp. 480–487, Dec. 2006.

[26] T. B. Swartz, P. S. Gill, D. Beaudoin, and B. M. de Silva, "Optimal batting orders in one-day cricket," Comput. Oper. Res., vol. 33, no. 7, pp. 1939–1950, Jul. 2006, doi: 10.1016/j.cor.2004.09.015.

[27] K. Passi and N. Pandey, "Increased prediction accuracy in the game of cricket using machine learning," arXiv:1804.04226, Apr. 2018. The study evaluates supervised learning for cricket outcome prediction using structured match attributes.

[28] R. Lamsal and A. Choudhary, "Predicting outcome of Indian Premier League matches using machine learning," arXiv:1809.09813, Sep. 2018. The work examines IPL match features and comparative classification approaches.

[29] K. C. Srikantaiah, A. Khetan, B. Kumar, D. Tolani, and H. Patel, "Prediction of IPL match outcome using machine learning techniques," arXiv:2110.01395, Oct. 2021. The paper compares learning methods for pre-match outcome estimation.

[30] R. Chakwate and M. R. A., "Analysing long short-term memory models for cricket match outcome prediction," arXiv:2011.02122, Nov. 2020. The study investigates sequence modelling and temporal information for cricket prediction.