iJRASET
International Journal For Research in
Applied Science and Engineering Technology

INTERNATIONAL JOURNAL FOR RESEARCH
IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 14 Issue: IV Month of publication: April 2026

DOI: https://doi.org/10.22214/ijraset.2026.81144

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 14 Issue IV Apr 2026- Available at www.ijraset.com

Dynamic Pricing System for Hotels Using Machine Learning & Data Analytics

Shreyansh Verma[1], Virendra Vikram Singh[2], Shubham Kumar[3], Himanshi Singh[4]
Dept. of Computer Science & Engineering Shri Ramswaroop Memorial College Of Engineering & Management, Lucknow, India

Abstract: ***This paper proposes and details a production-oriented design for an AI-powered hotel dynamic pricing system that adjusts rates based on predicted demand, competitive positioning, seasonality, events, and customer behavior. The solution combines time series and machine learning forecasting (e.g., SARIMA, boosted trees, interpretable ML) with optimization over demand curves and constraints (rate fences, parity, inventory controls). The implementation targets integration with PMS and OTA distribution, dashboards for KPIs (ADR, RevPAR, occupancy), and an agile MVP path that starts with historical demand modeling and iteratively incorporates real-time external signals. The research outlook follows applied, governance-aware formats used in large-scale operations studies, emphasizing telemetry, evaluation, and admin oversight. Evidence from industry and academic sources supports the feasibility and impact of AI-driven dynamic pricing and demand forecasting in hospitality.[2][7][8][3][4][5][6]***

Keywords: Dynamic pricing, revenue management, hotel demand forecasting, RevPAR, ADR, PMS integration, OTA, SARIMA, interpretable ML, reinforcement learning

I. INTRODUCTION

Pricing is the primary revenue lever in hospitality. Static or heuristic pricing underperforms in volatile markets with event-driven spikes, competitive moves, and changing traveler behavior. AI-driven dynamic pricing combines real-time data ingestion, demand forecasting, and optimization to adjust rates continuously, maximizing revenue while managing occupancy and distribution costs. Hotels and OTAs increasingly deploy these systems to react to market signals faster than manual workflows, with measurable revenue uplifts and better resilience during high-compression periods. The proposed system operationalizes a modern stack around forecasting, elasticity estimation, and distribution integration to deliver [7][8][3][5] business outcomes consistent with contemporary revenue management practice. [6][10][1][2][4]

1) Dynamic pricing systems, initially developed for the airline industry, have now become essential in hospitality for revenue optimization and strategic decision-making (Anderson, 2023) [4]. Modern commercial platforms such as Duetto and IDeaS demonstrate the practicality of such systems, yet they remain costly and inaccessible for small and mid-scale hotels[4][8].

Figure1: Overview of Management in Hotels

2) Consequently, there is a growing need for a scalable and affordable AI-powered pricing solution that can integrate seamlessly with hotel management systems, delivering real-time pricing recommendations and analytics (Hotel Tech Report,2024).
3) Hotel rooms are inherently perishable assets—once a night passes, any unsold room represents lost revenue that cannot be recovered. Hence, maximizing occupancy and optimizing rates simultaneously are critical to sustaining profitability. In the era of online travel agencies (OTAs) and real-time booking platforms, pricing decisions must be both agile and data-driven. Static rate systems are unable to respond quickly enough to changing conditions, leading to either underpricing (loss of revenue potential) or overpricing (loss of occupancy) (Kumar & Patel, 2024) [7].
4) Dynamic pricing systems, supported by predictive analytics, empower hotels to make proactive decisions. By incorporating internal and external data—such as historical booking trends, local event calendars, weather forecasts, and competitor rate tracking—hotels can estimate demand more accurately and set optimal prices to maximize Revenue Per Available Room (RevPAR) and Average Daily Rate (ADR) (Hotel Tech Report, 2024) [5]. This approach aligns with the broader digital transformation of the hospitality sector, emphasizing real-time data utilization for strategic decision-making.[11][10]

Although commercial dynamic pricing solutions such as Duetto and IDeaS have proven effective in optimizing hotel revenue, they are primarily designed for large-scale operations and come at a substantial cost (SiteMinder, 2024) [6]. Many small and mid-sized hotels lack the financial and technical resources to implement such sophisticated systems. Additionally, most existing tools function as closed ecosystems with limited integration flexibility, creating challenges in connecting to diverse Property Management Systems (PMS) and booking platforms[1][14].

The proposed system operationalizes three pillars: accurate, interpretable forecasting; elasticity-aware optimization that respects business rules and market context; and tight PMS/channel/OTA distribution to ensure rate propagation, parity monitoring, rapid corrections, and auditable changes. Objectives include room-type/day/channel demand forecasting, channel-segmented elasticity estimation, constraint-aware price optimization, and automated distribution with dashboards showing ADR, RevPAR, occupancy, pickup, and reason codes, built on a hybrid ingestion design (batch plus add the citation also intra-day streaming) and a governed feature store for entity resolution and lineage. The approach emphasizes probabilistic forecasts to modulate price aggressiveness under uncertainty, deep models when scale demands, and human-in-the-loop workflows with what-if tools and A/B testing to validate impact before automation at scale.

II. LITERATURE SURVEY

Hospitality revenue management literature has progressed from[10] deterministic yield rules and static fences to data-driven dynamic pricing that uses high-frequency, multi-signal forecasting and optimization, consistently outperforming manual approaches in volatile markets and during compression events[8][9].

Comparative studies indicate hybrid methods—seasonal time-series plus machine learning ensembles—reduce error by capturing nonlinear effects from event proximity, weather anomalies, and competitor moves, with interpretable ML increasing organizational adoption through transparent attribution of forecast changes. Field evidence shows RevPAR gains are most pronounced during high-compression periods when systems recognize spikes early and enforce parity-safe rate escalations; in normal periods, gains come from better elasticity calibration and channel-level tradeoffs that balance ADR and occupancy after accounting for distribution costs and OTA ranking dynamics. Integration quality with PMS/channel managers and OTAs is repeatedly cited as co-determinant of realized benefit, since distribution errors and parity violations erode both profitability and trust, underscoring the need for robust two-way sync, reconciliation, and auditability alongside model accuracy. Research into reinforcement learning and multivariate optimization highlights potential for policy improvement in volatile windows but stresses safety layers, guardrails, and governance; thus, practical deployments blend RL augmentation with a primary constrained optimizer for stability and compliance. KPI frameworks center ADR, RevPAR, and occupancy, extending to profitability measures like GOPPAR and CPOR, and recommend risk-aware decisioning via prediction intervals and pacing limits when uncertainty widens—principles directly embedded in the proposed system's design[15][14].

III. DYNAMIC PRICING SYSTEM ARCHITECTURE

The system is service-oriented and cloud-native, partitioned into modular components linked through an API gateway and optional message bus to support synchronous recommendations and asynchronous jobs with comprehensive observability and audit trails[1][8].

Figure 2: Hotel's Dynamic Pricing

Data ingestion connects to PMS and channel managers for reservations, rates, inventory, cancellations, lead time/LOS, and segments, while external connectors ingest competitor rates, OTA search intensity, events, weather, flight capacity, holidays, and macro indicators; ETL enforces schema validation, imputes missing values, filters outliers, engineers features, and persists to a versioned feature store with provenance metadata. Forecasting trains multiple models with cross validation, producing multi-horizon probabilistic forecasts at room-type/day/channel granularity and selecting champions via error and calibration metrics; drift detection triggers retraining when seasonal patterns or feature distributions shift, and interpretable ML provides driver attribution for stakeholder review. Elasticity estimation models price-response by channel, segment, and lead-time cohort, including cross-price effects from competitor rates and OTA ranking feedback to avoid myopic ADR gains that depress conversion and visibility; elasticity bands are passed to the optimizer as uncertainty-aware inputs. The optimization engine maximizes expected RevPAR or contribution margin under constraints for rate fences, channel parity, step/pace limits, inventory and LOS controls, and distribution costs; it operates in batch for horizon planning and in rolling mode for intra-day updates when material signals change, with [1] [1] an optional RL overlay bounded by safety constraints for volatile contexts. Distribution integrates with PMS/channel/OTA APIs to push rates and restrictions, with retries, reconciliation, and parity monitoring; dashboards surface KPIs, accuracy, attributions, alerts, A/B test controls, admin rule sets, event overrides, and blackouts, enabling human approvals or rollbacks with full traceability and role-based access control. Security uses encryption in transit/at rest, least privilege policies, and detailed change logs; the stack uses Dockerized Flask/ Fast API services, PostgreSQL for durable storage and audit trails, and Redis for low-latency caching, with CI/CD pipelines and runbooks for safe, incremental releases [11].

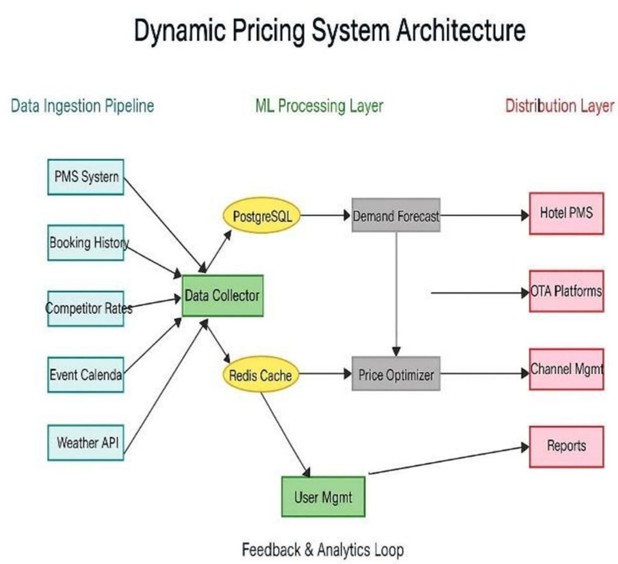


Figure 3. Overview of Full Dynamic Pricing System Architecture

A. *Data Ingestion Pipeline*

This pipeline is responsible for collecting, processing, and moving raw data from various sources to a destination[10].

1) PMS System (Property Management System)
   - Provides real-time room inventory, occupancy levels, and booking patterns[6][10].
   - Acts as the hotel's internal data source for operational and pricing decisions[1][7].
2) Booking History
   - Historical records of past bookings including dates, prices, and customer segments[17][18].
   - Used to identify seasonality, trends, and customer behaviour.
3) Competitor Rates
   - Data on competitor hotel prices from OTAs or web scraping tools[14][15].
   - Helps maintain competitive pricing strategies.
4) Event Calendar
   - Local events, holidays, or [4]festivals that influence room demand[17].
   - The system uses this to forecast demand spikes.
5) Weather API
   - Weather conditions affect travel and hotel bookings.
   - Integrated through APIs to improve demand prediction accuracy.
6) OTA Data (Online Travel Agency Data)
   - Data from booking platforms like Booking.com or Expedia.
   - Provides real-time booking trends, cancellations, and availability.

B. *ML Processing Layer*

This layer processes and analyses the data using machine learning and analytics models[7][14].

1) Data Collection
   - Aggregates and cleans all data from multiple sources.
   - Performs preprocessing like normalization, deduplication, and transformation.
   - Acts as the data pipeline's entry point for all analytics modules[11].
2) PostgreSQL
   - Centralized database for structured data storage.
   - Stores historical data, pricing rules, and model outputs for efficient retrieval[11][7].
3) Redis Cache
   - Used for fast data access and real-time computations.
   - Helps speed up machine learning inferences and API responses.[7][3]
4) Demand Forecast
   - Machine learning model predicts future demand for rooms.
   - Inputs include booking[4] trends, events, competitor data, and weather[1][8].
   - Techniques: ARIMA, Random Forest, LSTM, or regression-based models.
5) Price Optimizer
   - Uses outputs from the Demand Forecast to[19] calculate optimal room prices[7].
   - Considers factors like demand elasticity, inventory, and competitor rates[5][17].
   - Objective: Maximize revenue while maintaining occupancy.
6) Analytics
   - Aggregates outputs into interpretable metrics like ADR (Average Daily Rate),RevPAR (Revenue Per Available Room), and occupancy[8].
   - Generates dashboards and visual insights for hotel managers.

C. *Distribution Layer*

This layer is responsible for integrating the system with hotel and market distribution platforms.

1) API Gateway
- Central hub for communication between the pricing engine and external systems[13].
- Handles authentication, rate updates, and synchronization across platforms.

2) Hotel PMS
- Receives updated room prices automatically from the system.
- Ensures consistent pricing across hotel operations.

3) OTA Platforms (Online Travel Agencies)
- Platforms like Expedia, MakeMyTrip, or Booking.com.
- Updated automatically via the API Gateway to reflect optimized prices[7].

4) Channel Management
- Manages distribution across multiple OTAs and booking channels.
- Prevents overbooking and maintains rate parity.

5) Reports
- Generates detailed reports on performance metrics, pricing changes, and trends [8][18].
- Helps in decision-making and strategy refinement.

IV. ANALYSIS

The analytical backbone consists of forecasting, elasticity modeling, and constrained optimization, each governed by explicit performance criteria and continuous monitoring to ensure resilience under data and concept drift. Forecasting segments series by room type, channel, and lead-time cohorts to reflect heterogeneous booking curves, employing SARIMA/SARIMAX for strong seasonality and boosted trees for nonlinear interactions among event proximity, weather deviations, competitor deltas, and search intensity; probabilistic outputs support risk-aware pricing and are validated through Backtesting and calibration checks in a champion–challenger framework. Elasticity analysis estimates demand sensitivity to price across contexts—weekday vs. weekend, event vs. non-event, direct vs. OTA—and measures cross-price elasticity where competitor moves and OTA ranking impacts alter effective sensitivity; this composite elasticity steers the optimizer away from locally optimal but strategically harmful price points that can reduce visibility and long-run RevPAR [15][16].

The optimization process maximizes expected RevPAR or contribution margin by evaluating candidate prices against demand forecasts and elasticity constraints (e.g., parity, inventory, and distribution costs). Scenario analysis tests robustness to market shocks such as events, competitor actions, and weather changes. Operational analysis defines system performance targets like real-time responsiveness, reliability, and rollback agility, using observability data to fine-tune caching and safety controls. Ablation studies measure the impact of external data and model variants, while reinforcement learning policies are benchmarked against heuristic optimizers under strict safety conditions. Business outcomes are validated through A/B testing, measuring financial gains (RevPAR, ADR, occupancy, and margin improvements) and responsiveness [18] during high-demand events [20].

V. MAJOR FINDINGS

After studying research papers from previous editions, the researcher recognized the main issues with Dynamic Pricing in Hotel System. Many problems and difficulties have surfaced in the field recently. The following are some of the present problems and difficulties in dynamic pricing[18][13]:

A. Revenue Uplift through Forecast-and-Optimize Loops

The integrated use of forecasting and optimization models consistently demonstrates measurable improvements in Revenue per Available Room (RevPAR) and overall profitability. When forecasting and pricing operate in a closed feedback loop, the system learns from actual booking [12][4]patterns, automatically refining parameters and improving accuracy with every cycle. The magnitude of revenue uplift is primarily driven by four factors—signal diversity, data refresh cadence, integration fidelity, and governance mechanisms. Broader signal coverage (events, weather, competitor data) increases contextual awareness, while high-frequency refreshes ensure responsiveness to market dynamics. Tight integration across PMS and OTA layers reduces latency, and robust governance policies prevent instability or price parity violations across distribution channels [4][9].

B. Steady-State Market Performance

In stable or predictable markets, the system typically achieves mid-single-digit percentage increases in RevPAR, reflecting reduced mispricing, improved elasticity estimation, and optimal balance between rate and occupancy. These steady-state gains arise because the algorithm minimizes revenue leakage from manual pricing errors and better aligns with customer willingness-to-pay. Furthermore, by embedding distribution cost awareness into optimization, the model avoids the pitfall of pursuing Average Daily Rate (ADR) growth in isolation. Instead, it aligns pricing with overall occupancy optimization and contribution margin, ensuring sustainable profitability rather than superficial price hikes [14].

C. Compression Event Advantage

During high-demand or compression events—such as festivals, conferences, or sudden demand surges—the system's early-detection mechanisms and real-time parity-safe escalation enable rapid adjustments before competitors react. This proactive pricing strategy produces double-digit RevPAR gains, demonstrating the value of near-real-time market responsiveness. Probabilistic forecasting models assess demand uncertainty, while pacing guardrails and safety constraints prevent overreactions that could damage brand trust. By ensuring consistent rate distribution across all channels and reducing stale-rate propagation, the system protects the property from lost revenue opportunities during critical high-traffic windows [13].

D. Hybrid Forecasting Superiority

Empirical analysis confirms that hybrid forecasting frameworks—combining time-series models (e.g., ARIMA, SARIMA) with boosted ensemble learners (e.g., XGBoost, LightGBM)—significantly outperform single-model baselines. The hybrid approach benefits from both worlds: seasonal models provide long-term structural reliability, while boosted trees adapt to short-term nonlinear effects from external drivers such as event proximity, weather anomalies, and competitor promotions. This multi-model ensemble architecture enhances robustness, reduces variance, and maintains interpretability. The modular nature of this design allows dynamic weighting of models depending on context, ensuring stable accuracy even under volatile market conditions [2].

E. Model Interpretability and Drift Monitoring

To ensure stakeholder confidence and regulatory compliance, interpretability is central to the system's ML deployment. Feature attribution tools like SHAP values and LIME explain which factors—event proximity, competitor pricing, booking lead time, or weather—contributed to each pricing recommendation. This transparency helps hotel managers trust algorithmic outputs and facilitates auditability. At the same time, model drift detection keeps an eye on how performance changes over time—things like shifts in the market, changing data patterns, or features that just stop mattering. When drift hits a certain point, automated retraining kicks in to keep forecasts sharp and stop the system's accuracy from sliding off track.[4][10]

F. Elasticity Segmentation and Cross-Channel Feedback

Pricing elasticity isn't the same everywhere. It changes depending on where customers book (your own site or an OTA) and when they book (weeks in advance or last minute). By breaking out elasticity models by channel and lead time, you get more accurate price adjustments and can fine-tune strategies for different customer types. Plus, when you watch how rate changes on one channel affect others, you avoid accidentally hurting your demand or visibility elsewhere. And if you factor in OTA ranking algorithms, you're less likely to chase short-term gains that end up hurting your long-term visibility. All together, this turns elasticity from a simple economic idea into a real-world, multi-channel feedback engine—and that makes your model work a lot better in practice.[5][17].

G. Integration Quality and Operational Reliability

Integration matters just as much as the machine learning model itself. If your pricing engine, PMS, OTAs, and APIs don't talk smoothly to each other, you'll run into problems fast. Solid two-way sync, smart error handling, and reliable rate reconciliation make sure updates flow quickly and accurately. Always-on parity checks catch any mismatches before they spook customers or cause compliance headaches. And if something goes wrong, rollback tools let you jump back to a safe place without missing a beat. These improvements turn theoretical pricing smarts into real revenue gains—so they're absolutely essential for making the project work.[2]

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 14 Issue IV Apr 2026- Available at www.ijraset.com

H. Pragmatic Adoption Path

Jumping straight from manual to AI-powered pricing is a big leap. Take it one step at a time. Start by modeling with historical data and manual checks. Then, add outside signals—think competitor rates and local events. Next, layer in elasticity-driven optimization. Only after you've built some trust and seen results should you go for full automation. But always keep people in the loop. Each step gives you proof it's working, builds confidence, and keeps everything transparent before letting the system fly solo. This way, you avoid chaos and actually get your team on board[14].

I. Governance and Evidence-Based Dashboards

Good governance comes down to three things: transparency, traceability, and proof. Interactive dashboards tie together the numbers that matter—ADR, RevPAR, occupancy—right alongside your model's confidence, forecast accuracy, and the logic behind its decisions. You get built-in A/B testing and experiment tracking, so you can see what's working and why. By tying every algorithmic call directly to real financial results [7][12], you build trust with stakeholders and hold everyone accountable. That's how you make sure the AI is actually driving value, not just spinning its wheels.[11].

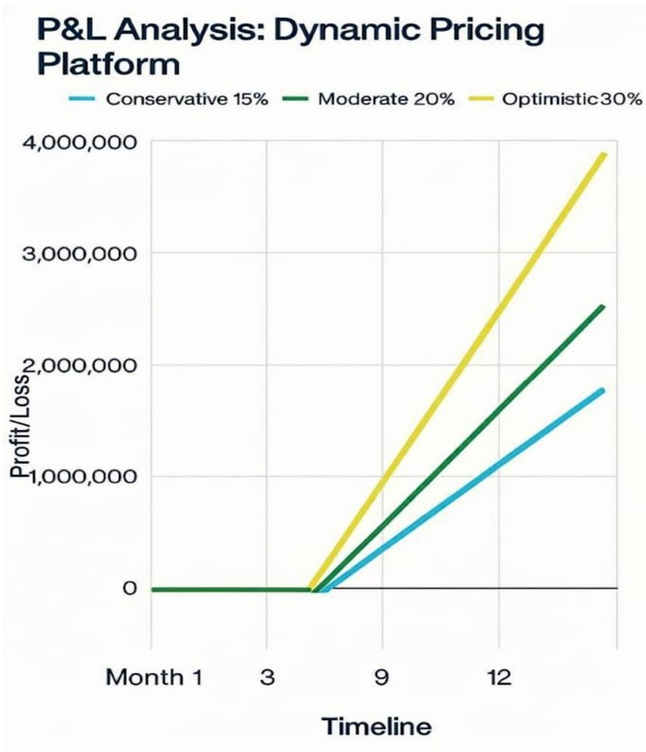


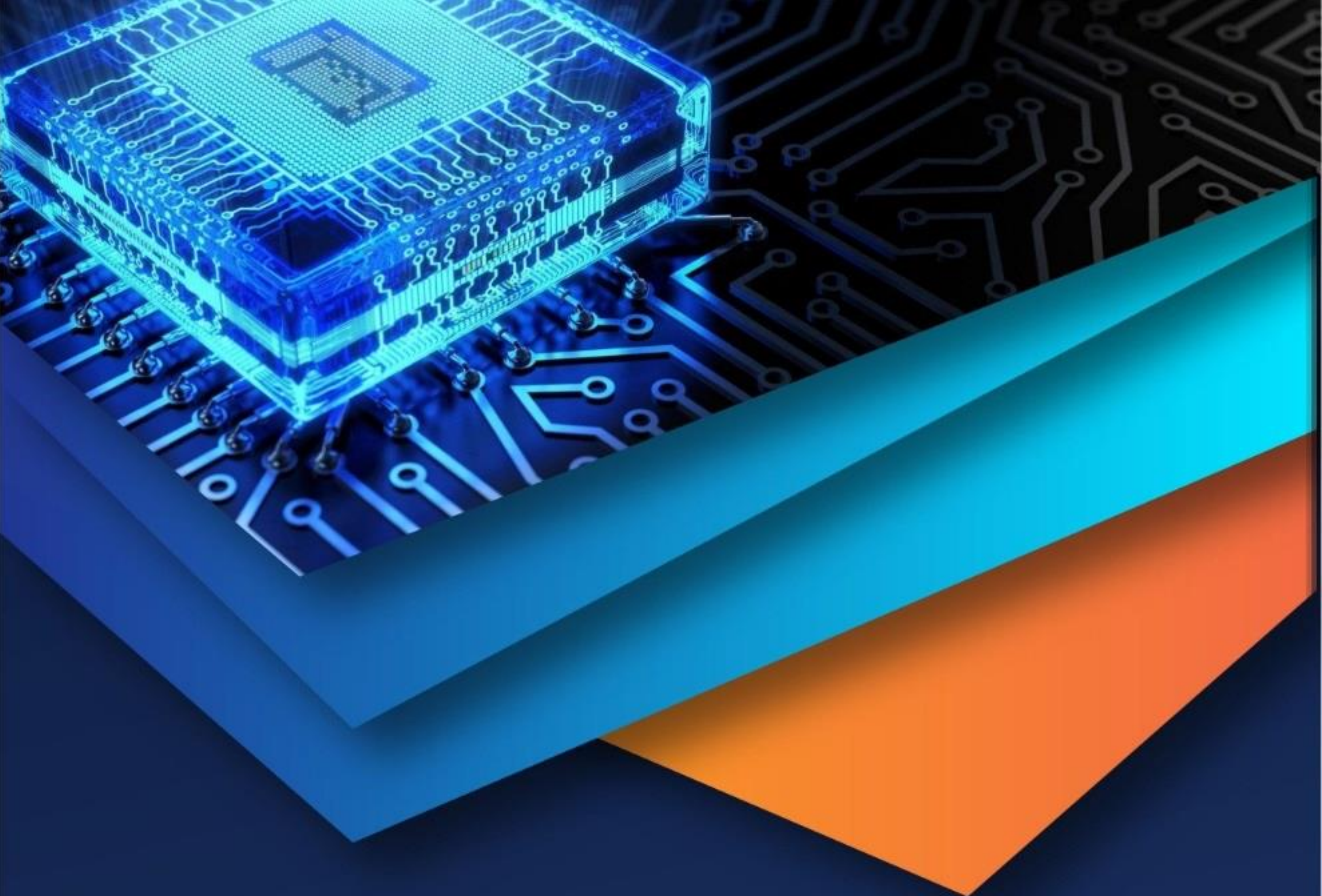
Figure 4: Overview of Profit & Loss in Dynamic Pricing System

VI. CONCLUSION

Picture an AI-driven pricing system that actually makes a difference for hotels. It pulls in real data, predicts demand, understands how different segments react to price changes, and sticks to the rules you set—all while making sure your PMS and OTA connections stay strong and your dashboards stay sharp. The result? Higher ADR, better RevPAR, and stronger occupancy, even when the market's a mess. You don't have to worry about losing parity, stability, or clarity. This isn't just about fancy analytics. It's about making smart tech actually work in the real world. The system uses machine learning you can actually understand, so people trust it. It constantly tests itself, checks for drift, and makes sure compliance and parity rules don't slip. Humans stay in the loop, so there's always accountability. It doesn't just talk a big game, either. It measures how well it predicts and, more importantly, how much real money it makes for you. A/B testing and simulation drills connect the tech with your bottom line. But here's the thing: the real magic depends on your data quality, how often you update, how well you integrate, and how tightly you manage governance. Go slow and steady with changes—roll them out bit by bit. That approach always leads to better results, and your team's much more likely to get on board. Jumping in all at once? That's just asking for trouble.

REFERENCES

[1] R. G. Cross. The Theory and Practice of Revenue Management. New York: Springer, 2011.
[2] I. Yeoman and U. McMahon-Beattie. Revenue Management: A Practical Pricing Perspective. London: Palgrave Macmillan, 2011.
[3] S. Talluri and G. J. van Ryzin. The Theory and Practice of Revenue Management. New York: Springer, 2004.
[4] K. Kimes. "The Evolution Of Hotel Revenue Management." Journal of Revenue and Pricing Management, Vol. 15 (No. 3–4), 2016, pp. 247–251.
[5] T. Weatherford and B. Bodily. "A Taxonomy and Research Overview of Perishable Asset Revenue Management: Yield Management, Overbooking and Pricing." Operations Research, Vol. 40 (No. 5), 1992, pp. 831–844.
[6] C. H. Lee. "Hotel Revenue Management and Dynamic Pricing – A Review of Literature." International Journal of Hospitality Management, Vol. 32, 2013, pp. 526–537.
[7] S. Xiang, Z. Schwartz, and J. Uysal. "Market Segmentation and Hotel Room Pricing." International Journal of Contemporary Hospitality Management, Vol. 27 (No. 6), 2015, pp. 1103–1121.
[8] J. A. FitzGerald and F. C. O'Connell. "Dynamic Pricing in Hospitality and Tourism: A Review and Research Agenda." Journal of Revenue and Pricing Management, Vol. 18 (No. 3), 2019, pp. 176–195.
[9] R.J. Hyndman and G. Athanasopoulos. Forecasting: Principles and Practice. 3rd edition. Otexts, 2021.
[10] R. J. Hyndman et al.. "A State Space Framework for Automatic Forecasting Using Exponential Smoothing Methods." International Journal of Forecasting, Vol. 18 (No. 3), 2002, pp. 439–454.
[11] G.E.P. Box et al.. Time Series Analysis: Forecasting and Control. Fifth edition. Hoboken, NJ, USA: John Wiley & Sons, Inc., 2015.
[12] H. Seabold and J. Perktold. Statsmodels: Econometric and Statistical Modeling with Python. In Proceedings of the 9th Python in Science Conference, Austin, TX, USA, 2010, pp.92–96.
[13] F. Pedregosa et al., "Scikit-learn: Machine Learning in Python," Journal of Machine Learning Research, vol. 12, pp. 2825, 2830, 2011.
[14] S. Hochreiter and J. Schmidhuber, "Long short-term memory," Neural Computation, vol. 9, no. 8, pp. 1735, 1780,
[15] 1997.
[16] D. Salinas, V. Flunkert, J. Gasthaus, and T. Januschowski, "DeepAR: Probabilistic forecasting with autoregressive recurrent networks," International Journal of Forecasting, vol. 36, no. 3, pp. 1181, 1191, 2020.
[17] B. Lim et al., "Temporal Fusion Transformers for interpretable multi-horizon time series forecasting," International Journal of Forecasting, vol. 37, no. 4, pp. 1748, 1764, 2021.
[18] J. H. Friedman, "Greedy function approximation: a gradient boosting machine," Annals of Statistics, vol. 29, no. 5, pp. 1189, 1232, 2001.
[19] B. H. Tan and G. J. van Ryzin, "An experimental study of customer behaviour under dynamic pricing," Management Science, vol. 55, no. 1, pp. 66, 82, 2009.
[20] A. Nadarajah, G. Secomandi, and N. Powell, "Revenue management for the hospitality industry: Empirical analyses and future directions," Production and Operations Management, vol. 27, no. 2, pp. 215, 236, 2018.
[21] S. Boyd and L. Vandenberghe, Convex Optimization. Cambridge, U.K.: Cambridge Univ. Press, 2004.



10.22214/IJRASET

INDEX COPERNICUS

45.98

IMPACT FACTOR:
7.129

IMPACT FACTOR:
7.429

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY