Indian Journal of Computer Science and Technology
https://www.doi.org/10.59256/indjcst.20260503125
Volume 5, Issue3 (September-December 2026), PP: 1046-1056.
www.indjcst.com
ISSN No: 2583-5300

Dynamic Pricing and Revenue Optimization Using Machine Learning: A Survey

Girish Nayak[1], Harini V[2]
[1, 2]*Department of Master of Computer Applications, Acharya Institute of Technology, Bengaluru, Karnataka, India.*

To Cite this Article: *Girish Nayak[1], Harini V[2], "Dynamic Pricing and Revenue Optimization Using Machine Learning: A Survey", Indian Journal of Computer Science and Technology, Volume 05, Issue 03 (September-December 2026), PP: 1046-1056.*

Copyright: ©2026 This is an open access journal, and articles are distributed under the terms of the Creative Commons Attribution License; Which Permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited.

Abstract: *Retailers, airlines and hotels sell limited or perishable inventory to customers whose willingness to pay changes with time, season and competition. Online sales and live data now make frequent price changes possible, so pricing has become a major lever for revenue. Poorly set prices either lose sales or leave money on the table.* ***Gap.*** *Existing reviews usually cover a single industry or a single technique, such as reinforcement learning, and rarely compare demand forecasting, price optimization and real-world business constraints together, and they pay little attention to how pricing models are validated before launch.* ***Novelty.*** *This survey links demand estimation, price optimization and business guardrails in one framework across retail, airline and hotel pricing. It contributes a literature review, an industry comparison, a technique taxonomy, a reference architecture and a forecast-guided pricing algorithm.* ***Method.*** *Peer-reviewed studies from 2015 to 2026 were collected from IEEE Xplore, ACM Digital Library, ScienceDirect and Google Scholar using a PRISMA-style protocol. Studies were compared on datasets, preprocessing, demand model, optimization method, validation design and business metrics.* ***Results.*** *Gradient-boosted trees and neural networks produce the most accurate demand forecasts, while bandits and reinforcement learning adapt prices best when demand is uncertain or changing. Reported revenue gains depend heavily on simulation assumptions, and few studies test on real market experiments. Fairness, competitor reactions and limits on price changes are seldom evaluated.* ***Conclusion.*** *Effective dynamic pricing needs accurate demand models, exploration that is safe for the business, and clear pricing rules. Future work should focus on realistic evaluation, fairness and explainable pricing decisions for customers.*

Keywords: *dynamic pricing, revenue management, demand forecasting, price elasticity, multi-armed bandits, reinforcement learning, retail, airlines, hotels*

I. INTRODUCTION

Setting the right price is one of the most direct ways a business can change its revenue. In dynamic pricing, a seller adjusts prices over time in response to demand, remaining inventory, the time left before sale and competitor behaviour. The idea began in airline revenue management, where seats are perishable and demand differs sharply between early and late buyers [1]. It later spread to hotels, where an unsold room night cannot be recovered, and to retail and e-commerce, where prices can now be changed many times a day at almost no cost [1], [2]. Machine learning (ML) strengthens this practice because it can learn how price, season, location and customer type affect demand from large volumes of data.

Pricing with ML is nevertheless difficult. The true demand curve is unknown and noisy, and it shifts with seasons, promotions and competitor moves. Observed sales are censored, because a sold-out flight or hotel shows only the demand that was served and not the demand that was turned away. Learning what customers will pay requires trying different prices, but each experiment can cost revenue, which creates the exploration-exploitation dilemma. Companies must also respect price floors and ceilings, avoid sudden changes that upset customers, and stay clear of unfair or collusive behaviour. Finally, the effect of a price policy is hard to verify without a live experiment. The problem addressed in this paper is how to combine demand estimation and price optimization so that revenue improves while business and fairness constraints are respected.

The classical literature formulated pricing as an optimization problem. Gallego and van Ryzin [3] analysed the pricing of a finite stock over a finite horizon, and Besbes and Zeevi [4] studied how to price when the demand function is unknown. den Boer [5] surveyed the resulting field of pricing and learning. Hotel studies compared forecasting methods for revenue management [6]. As richer data became available, researchers introduced feature-based pricing models that use product, customer and market attributes [7], together with machine learning forecasters such as gradient boosting [8] and long short-term memory (LSTM) networks [9].

Another line of work treats pricing as sequential decision making under uncertainty. Thompson-sampling bandits were applied to online network revenue management [10], and reinforcement learning was used for real-time pricing in changing markets and under competition [11], [12]. At the same time, evidence of algorithmic pricing on online marketplaces [2] and of learning algorithms that can settle on collusive prices [13] has raised concerns about fairness and regulation.

Despite this progress, several gaps remain. Most reviews concentrate on a single industry or a single technique and do not compare retail, airline and hotel pricing on common ground. Demand forecasting, price optimization and operational

rules are usually discussed separately even though they form one decision pipeline. Evaluation practice is uneven: many studies rely on simulated demand, random data splits or revenue gains without a baseline, and few report real experiments. Fairness, competitor reactions and limits on price changes receive little attention.

This paper has four objectives: to build a taxonomy of ML techniques for dynamic pricing and revenue optimization; to compare how retail, airline and hotel studies differ in data, constraints and methods; to present a reference architecture and a forecast-guided pricing algorithm that can be used to compare studies; and to identify open challenges and directions for future work.

The study is useful to several groups. Researchers obtain a consistent basis for comparing pricing methods across industries. Revenue managers and data scientists in retail, airline and hotel businesses can see which techniques fit their data and constraints and what evaluation evidence to demand before changing prices. Regulators and consumer groups gain a clearer view of where pricing algorithms may raise fairness concerns.

The rest of the paper is organized as follows. Section 2 reviews the related literature. Section 3 presents the methodology: the survey protocol, datasets, data preprocessing and normalization, the reference architecture, a comparison of industries, technique families and evaluation metrics, the pricing algorithm, and a mathematical derivation of the pricing rule, and three example ML models. Section 4 discusses the findings and open challenges, and Section 5 concludes.

II. LITERATURE REVIEW

This section reviews the work on which dynamic pricing with machine learning is built. It follows the order in which the field developed: classical revenue management, demand estimation and forecasting, pricing under demand uncertainty, reinforcement learning, and finally the concerns of competition and fairness. Table 1 lists ten representative studies from 2021 to 2025.

2.1 Classical Revenue Management and Pricing Theory

Revenue management started in the airline industry, where a fixed number of seats must be sold before departure. Early work proposed a rule for deciding how many low-fare seats to protect for later, higher-paying passengers, and this was extended to several fare classes with the expected marginal seat revenue heuristic. Textbook treatments of the field [1] bring together capacity control, overbooking and pricing under one framework and show how the same logic applies to hotels, car rental and other perishable goods.

The pricing side of the problem was formalized by Gallego and van Ryzin [3], who derived optimal price paths for a finite stock sold over a finite horizon when demand is stochastic. Their result shows that the optimal price rises when stock is scarce and falls as the deadline approaches with stock left over, which is the behaviour observed in real airline and hotel prices. These models, however, assume that the demand function is known. In practice it is not, and this assumption is the main reason that learning-based approaches were developed.

2.2 Demand Estimation and Forecasting

Every pricing decision depends on an estimate of how demand responds to price. Econometric models such as log-linear and logit demand functions remain the common baseline because their elasticity parameters are easy to interpret [1]. In hotel revenue management, Weatherford and Kimes [6] compared exponential smoothing, moving averages, regression and pickup methods and found that the choice of forecasting method and the level of data aggregation clearly affect forecast error. Their work is still a standard reference for benchmarking forecasts in revenue management.

Feature-based models extend this approach by letting demand depend on product, customer and market attributes. Ban and Keskin [7] studied personalized pricing with high-dimensional features and heterogeneous price sensitivity. Regularized regressions such as stacked lasso and elastic-net models can handle many correlated inputs without overfitting [14]. Such studies move pricing from one price per product toward prices that depend on context.

Machine learning forecasters bring more flexibility. Gradient-boosted trees [8] handle tabular features such as lead time, day of week and promotions without scaling, and they have been strong performers in large forecasting comparisons, including the M5 competition on Walmart sales data [15]. Random forest and gradient boosting have likewise outperformed linear regression when forecasting demand from tabular data in other domains, such as building energy use [16]. Recurrent and attention-based networks, such as LSTM [9] and the Temporal Fusion Transformer [17], are better suited to many related series with long seasonal patterns, but they need more data and computing effort. A common limit of all these forecasters is that they predict demand under observed prices and do not choose prices. Because sales are censored by stock-outs and capacity, forecasts built on raw sales also tend to understate true demand unless it is unconstrained first [1].

2.3 Pricing under Demand Uncertainty: Learning and Bandits

When the demand curve is unknown, the seller must learn it while selling. Besbes and Zeevi [4] showed that pricing policies that first explore and then exploit can achieve near-optimal revenue. den Boer and Zwart [18] showed that prices which are too stable can stop the seller from learning, which means that some deliberate variation in price is necessary. den Boer [5] surveyed this field and made clear that the main tension is between earning revenue now and gathering information for the future.

Multi-armed bandits give a practical way to manage this tension. Bandit methods such as upper confidence bound and Thompson sampling treat each candidate price as an arm and balance trying uncertain prices against using the best known price. Ferreira et al. [10] applied Thompson sampling to online network revenue management, where many products share limited resources, and reported strong performance in experiments, with a field deployment at an online retailer. The main

risk of bandit pricing is the cost of exploration, since a poor exploratory price loses real sales, so safe exploration within price limits is needed.

2.4 Reinforcement Learning for Pricing

Reinforcement learning (RL) extends bandits by letting the current price change the future state, such as remaining inventory or customer reference prices. Deep reinforcement learning methods made it possible to handle larger state spaces, and Kastius and Schlosser [11] studied RL for dynamic pricing under competition and showed that agents can learn to respond to rival prices in a simulated market. Deep Q-networks have also been applied to adaptive driver-passenger matching on on-demand mobility platforms [19], where dispatch and surge pricing respond to the same imbalance between supply and demand.

RL is attractive for sequential problems with inventory, but it has practical difficulties. It usually needs a simulator for training because exploring on real customers is expensive, and results can be sensitive to the simulator's assumptions. Training can also be unstable. Policy-gradient methods, which update the policy directly, are one response to this and have been used for real-time driver-passenger matching [20]. These points explain why many RL pricing studies report gains in simulation that have not been confirmed in live experiments, a pattern that this survey examines in later sections.

2.5 Industry-Specific Studies

Airlines. The airline literature builds on seat-inventory control and fare-class optimization [1]. Recent studies add machine learning to forecast booking demand and, in some cases, to predict fares, using data such as days left before departure and route (for example the DB1B ticket sample and online fare datasets listed in Table 2). Fare classes and regulation limit how freely prices can move.

Hotels. Hotel studies emphasize forecasting and booking behaviour. Weatherford and Kimes [6] compared forecasting methods, and a public hotel booking dataset (Table 2) is used to model lead time, cancellations and average daily rate. Rate parity agreements and cancellation rules are the main business constraints.

Retail and e-commerce. Retail pricing has the largest data volume and the fastest price changes. Chen et al. [2] documented algorithmic pricing by sellers on Amazon Marketplace. Association-rule mining of e-commerce transactions shows which products are bought together [21], which matters when the price of one item shifts demand for its complements. The M5 competition [15] provided a large public retail forecasting benchmark. For perishable produce, image-based ripeness detection [22] can signal when markdown pricing should begin. Because stock can be replenished, retail pricing is often framed around margin, promotion effects and competitor prices instead of a fixed capacity.

2.6 Competition, Collusion and Fairness

As algorithmic pricing spread, researchers began to ask what happens when several sellers use learning algorithms at the same time. Chen et al. [2] found evidence of algorithmic sellers and price changes on a major online marketplace. Calvano et al. [13] showed in simulations that Q-learning agents can learn to sustain prices above the competitive level without any explicit communication, which raises concerns for competition regulators. Studies of pricing under competition also show that competitor prices must be part of the demand model [11]. Personalized pricing [7] adds a further concern, because charging different customers different prices can be seen as unfair. Despite this, most pricing studies do not report fairness metrics, competitor reactions or limits on price changes, which is one of the gaps this survey addresses.

2.7 Summary and Research Gaps

Ref.	Study	Setting	Approach	Main contribution
[7]	Ban and Keskin (2021)	General / e-commerce	High-dimensional personalized pricing	Handles heterogeneous price sensitivity
[17]	Lim et al. (2021)	Forecasting (multi-series)	Temporal Fusion Transformer	Interpretable multi-horizon demand forecasts
[11]	Kastius and Schlosser (2022)	Competitive market	Deep reinforcement learning	Pricing agents that react to competitor prices
[15]	Makridakis et al. (2022)	Retail	Forecasting competition (M5)	Benchmark showing the strength of boosted trees and deep models
[23]	Gao, Le and Fang (2022)	Airlines	Q(lambda)-learning	Ticket pricing with strategic and myopic passengers, tested in simulation
[12]	Kastius, Kiele and Schlosser (2023)	Competitive online markets	Multi-agent RL with asymmetric information	Addresses unstable learning when several sellers use learning agents
[24]	Liu et al. (2023)	E-commerce	RL with product clustering and pre-training	Pricing for sparse, low-traffic products

Ref.	Study	Setting	Approach	Main contribution
[25]	Jo, Lee and Moon (2024)	Airlines	Deep exploration-based RL	Pricing when customers may delay their purchase
[26]	Saitta, D'Amico and Farinella (2024)	Hotels	ML models for room-price prediction	90-day price forecasts, tested on real data from three hotels
[27]	Bhargavi et al. (2025)	E-commerce	LSTM and ARIMA forecasts with DQN	Reported 18% revenue gain over static pricing in an experiment

Table 1. Representative studies from 2021 to 2025

The ten recent studies in Table 1 show the same pattern. Most of them use reinforcement learning, and several are evaluated in simulated or numerical settings instead of live markets [23], [12], [24]. Only one reports tests on real hotel data [26], and one reports a revenue gain from an experiment [27].

Three conclusions follow from this review. First, accurate demand estimation has improved strongly through boosted trees and deep forecasters, but accurate forecasts alone do not give good prices. Second, bandits and reinforcement learning provide principled ways to learn prices online, but their reported benefits depend on simulations and often lack real market tests. Third, issues of competition, fairness and limits on price changes are discussed mainly in the economics and regulation literature and are rarely built into the pricing pipelines themselves.

Earlier surveys usually address only one of these threads, for example pricing and learning in general [5], forecasting for hotels [6] or a single technique. None compares retail, airline and hotel pricing on common ground while linking demand estimation, price optimization and business guardrails. This survey fills that gap through the reference framework and algorithm presented in Section 3.

III. METHODOLOGY

3.1 Survey Protocol

The review follows a PRISMA-style process. Searches were run in IEEE Xplore, ACM Digital Library, ScienceDirect and Google Scholar using keyword groups such as ("dynamic pricing" OR "revenue management" OR "price optimization") AND ("machine learning" OR "reinforcement learning" OR "bandit") AND ("retail" OR "airline" OR "hotel"). Records were identified, duplicates removed, titles and abstracts screened, full texts assessed, and the remaining studies coded for industry, dataset, demand model, optimization method, validation design and metrics. Studies were included if they were peer-reviewed journal or conference papers in English, published from 2015 to 2026 (foundational earlier work excepted), on ML-based pricing or revenue optimization in retail, airlines or hotels, with a named dataset or experiment and reported metrics. Pure economic theory without learning or data, blogs, theses and non-reviewed reports were excluded.

3.2 Datasets

Pricing data is mostly proprietary, so the surveyed studies draw on a small number of public datasets or on simulated demand. Table 2 summarizes the datasets that appear most often across the three industries.

Dataset	Industry	Approx. size	Key characteristic
M5 Forecasting (Walmart)	Retail	42,840 product-store series	Daily sales with weekly sell prices, events and SNAP days
UCI Online Retail II	Retail / e-commerce	About 1.07 million invoice lines	UK online store, unit prices and quantities, 2009 to 2011
Hotel Booking Demand	Hotels	119,390 bookings	Two Portuguese hotels, lead time, average daily rate, cancellations
Inside Airbnb listings	Hotels / lodging	Varies by city	Listing prices, calendar availability and reviews
Flight Price Prediction (EaseMyTrip)	Airlines	About 300,000 fares	Indian domestic routes, days left before departure, class
US DOT DB1B Origin and Destination Survey	Airlines	10% sample of US tickets	Itinerary fares by route and quarter

Table 2. Commonly used datasets

3.3 Data Preprocessing and Normalization

Pricing data has features that ordinary sales data does not, in particular censored demand and strong dependence on time. The surveyed pipelines were therefore compared stage by stage.

1. **Cleaning:** remove duplicates, cancellations and returns where they distort demand, correct negative or zero prices, and fill missing prices by forward fill within each product, store, route or hotel.
2. **Demand unconstraining:** sales are capped by stock-outs, seat limits and room capacity, so true demand is estimated by excluding sold-out periods or by statistical unconstraining methods.

3. **Encoding:** one-hot or target encoding for product, route, hotel and customer segment, and cyclic encoding of day of week and month.
4. **Normalization:** numeric features are rescaled as shown in Table 3. Prices are also expressed relative to a reference price so that products and routes can be compared.
5. **Feature construction:** lead time (days to arrival or departure), remaining inventory and booking pace, lagged sales, rolling averages, holiday and event flags, the gap to competitor prices, and cluster labels that group sparse, low-traffic products, which nonlinear clustering optimization can scale to large catalogues [28].
6. **Validation split:** rolling-origin validation in which the model is trained on the past and tested on the next block of time; random splits are avoided because they leak future information. Figure 1 shows the overall machine-learning workflow.
7.

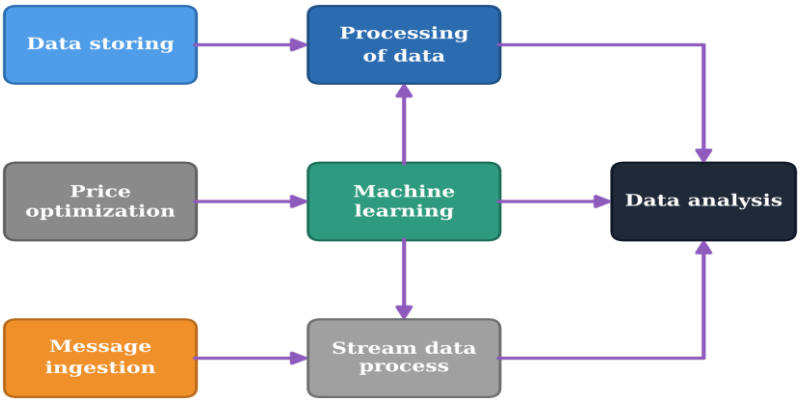

Figure 1. Machine-learning workflow for data processing, feature extraction, price optimization and data analysis.

Technique	Formula	When used
Min-max scaling	x' = (x - x_min) / (x_max - x_min)	Neural networks; bounded inputs
Z-score standardization	x' = (x - mean) / std	Regression, SVM and gradient-based models
Log transform	x' = ln(1 + x)	Skewed sales and revenue values
Relative price index	p_rel = p / p_ref	Comparing prices across items, routes or hotels
Log-log demand	ln q = a + e ln p + noise	Estimating price elasticity e

Table 3. Normalization and transformation techniques

All scaler parameters and reference prices are learned on the training period only and then applied to later data. Tree-based models are insensitive to scaling, whereas neural networks and linear models depend on it.

3.4 Architecture of the Reference Framework

Figure 2 shows the architecture used as the reference for comparing the surveyed work. It connects demand estimation to price optimization and places business guardrails and monitoring around the pricing decision.

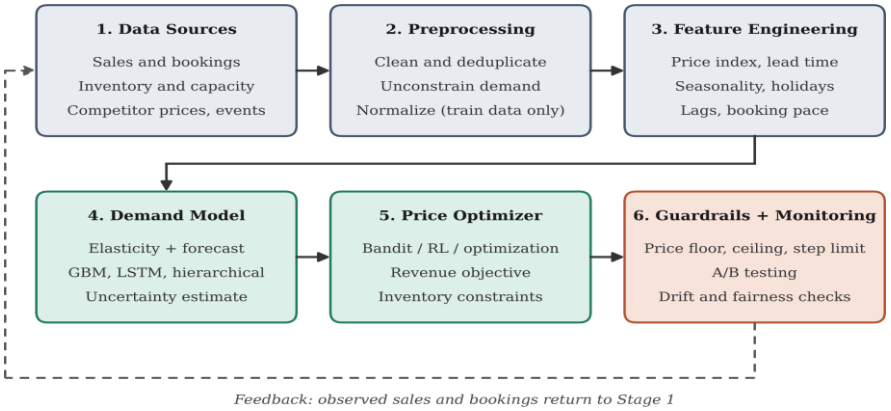

Figure 2. Reference architecture for dynamic pricing and revenue optimization.

Stage 1, Data sources: historical sales or bookings, current inventory and capacity, competitor prices and calendar events such as holidays and local festivals, and, in physical stores, IoT sensor and point-of-sale streams mined for demand patterns [29].

Stage 2, Preprocessing: data is cleaned and deduplicated, censored demand is unconstrained and numeric fields are normalized using training data only.

Stage 3, Feature engineering: price index, lead time, seasonality and holiday flags, lagged sales and booking pace are

created.

Stage 4, Demand model: gradient boosting, LSTM or hierarchical models estimate demand and price elasticity, and an ensemble or Bayesian model supplies uncertainty.

Stage 5, Price optimizer: a bandit, reinforcement learning agent or constrained optimization chooses the price that maximizes expected revenue given remaining inventory.

Stage 6, Guardrails and monitoring: price floors, ceiling

s and step limits are enforced, A/B tests measure real uplift, and drift and fairness checks run continuously. Observed sales and bookings flow back to the data sources. Where sales data cannot be pooled across stores or partners, federated continual learning with differential privacy can update the demand model without sharing raw data and without forgetting earlier seasons [30].

3.5 Comparison of Industries, Techniques and Metrics

Aspect	Retail	Airlines	Hotels
Inventory	Large stock, replenishable	Fixed seats, perishable	Fixed rooms, perishable
Price change frequency	Daily or hourly (online)	Many times before departure	Daily, by stay date
Key demand drivers	Promotions, competitor price, seasonality	Days to departure, route, booking pace	Lead time, events, day of week
Main constraints	Margin, brand image, price matching	Fare classes, regulation	Rate parity, cancellation rules
Typical ML methods	GBM, deep forecasters, bandits	Demand forecasting, RL, optimization	Forecasting, bandits, optimization

Table 4. Pricing characteristics by industry

Family	Representative methods	Best suited for	Main risk
Econometric / regression	Log-linear, logit demand models	Elasticity estimates, baselines	Rigid assumptions
Tree ensembles	Random Forest, XGBoost, LightGBM	Accurate tabular demand forecasts	Predict demand but do not choose prices
Deep forecasters	LSTM, Temporal Fusion Transformer	Many related series, long seasonality	Data and compute cost
Bandits	UCB, Thompson sampling	Learning prices online with limited data	Costly exploration
Reinforcement learning	Q-learning, DQN, PPO	Sequential pricing with inventory	Needs simulators; unstable training
Optimization-based	Dynamic programming, linear programming	Constrained revenue maximization	Depends on accurate demand input

Table 5. Technique families for dynamic pricing

Metric	Formula / meaning	Use
WAPE	sum |actual - forecast| / sum actual	Forecast accuracy across many items
RMSE	sqrt(mean((actual - forecast)^2))	Forecast error in demand units
Revenue uplift	(R_policy - R_baseline) / R_baseline	Business gain over current pricing
RevPAR	Room revenue / available rooms	Hotel performance
Load factor	Seats sold / seats available	Airline performance
Regret	Revenue of best fixed price - revenue achieved	Quality of online learning

Table 6. Evaluation metrics used for comparison

Forecast accuracy alone is not sufficient, since a very accurate demand model can still lead to poor prices. Studies are therefore compared on both forecast metrics and business metrics against a clearly stated baseline.

3.6 Algorithm

Algorithm 1 describes the forecast-guided pricing procedure used as the reference for comparing studies. A bootstrap ensemble of demand models provides uncertainty, one model is sampled at each step to balance exploration and exploitation, and business guardrails limit the chosen price. Hyperparameters may be chosen by grid search or by metaheuristic optimizers such as Coyote and Badger optimization [31]. The policy is judged against a baseline by A/B testing or off-policy evaluation

instead of simulation alone.

Algorithm 1: Forecast-Guided Pricing with Thompson-Style Exploration

```
Input : History H, price grid P, floor p_min, ceiling p_max, step limit
delta,
        horizon T, initial inventory I_1, salvage value v, retrain
interval k
Output: Pricing policy and revenue report R

 1. Clean H; unconstrain demand; build features
 2. Split H by time into Train, Valid, Test (rolling origin)
 3. Fit scaler and reference prices on Train only; transform Valid,
Test
 4. Train bootstrap ensemble {D_1, ..., D_B} of demand models on Train
 5. Select hyperparameters by WAPE on Valid
 6. for t = 1 to T do
 7.   Observe context x_t, inventory I_t, competitor price c_t
 8.   Draw one model D_b at random from the ensemble
 9.   for each price p in P with p_min <= p <= p_max, |p - p_prev| <=
delta do
10.     q <- min(D_b(x_t, p, c_t), I_t)
11.      V(p) <- p * q + v * (I_t - q)        // leftover stock at
salvage
12.   end for
13.   p_t <- argmax over p of V(p); apply business rules
14.   Offer p_t; observe sales q_t; I_(t+1) <- I_t - q_t
15.   Append (x_t, p_t, q_t) to the data
16.   if t mod k = 0 then retrain the ensemble on recent data
17.   if drift or error alert then retrain and widen exploration
18. end for
19.  Estimate uplift against baseline by A/B test or off-policy
evaluation
20. return policy, R
```

Figures 3 and 4 give the wider operational context and the idea behind dynamic pricing.

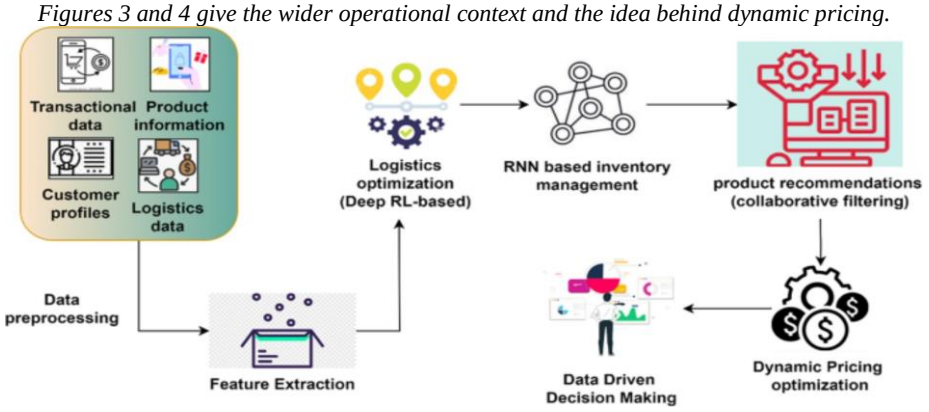

Figure 3. Data-driven operational architecture showing logistics optimization, inventory management, recommendations and dynamic pricing.

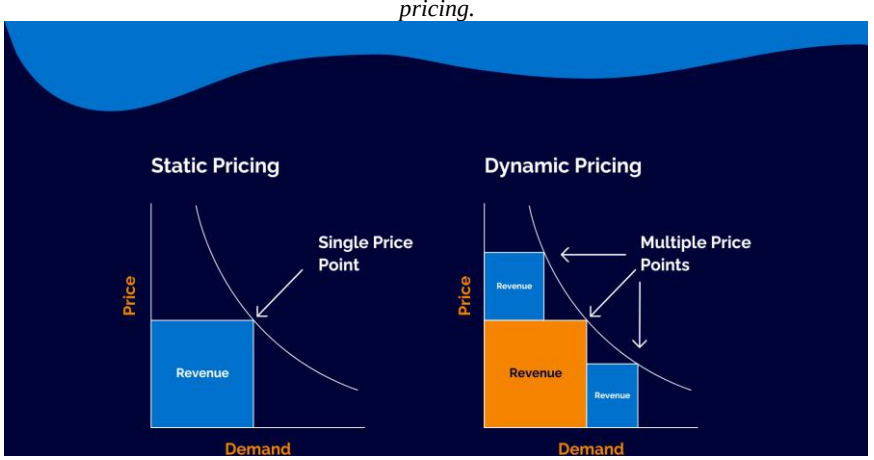

Figure 4. Comparison of static pricing and dynamic pricing in relation to demand and revenue.

3.7 Mathematical Formulation of the Pricing Rule

This subsection shows how the price in Algorithm 1 follows from the log-log demand model in Table 3 and the value function in step 11. The symbols are: p the price, q the demand, I the inventory, v the salvage value, p_min and p_max the price floor and ceiling, and A and e the demand parameters. The numbers in the example are illustrative and are not taken from any dataset.

Step 1: Demand and price elasticity. Table 3 writes demand as a log-log function of price, where u is random noise. Ignoring the noise and writing A = exp(a), demand becomes a power function of price:

$$\ln q = a + e \ln p + u \tag{1}$$

$$q(p) = A\, p^{e}, \qquad A = \exp(a) \tag{2}$$

The exponent e is the constant price elasticity: a 1% rise in price changes demand by e%. Demand falls when price rises, so e is negative; for e = -2, a 1% price rise lowers demand by 2%. The value of e is estimated as the slope of ln q on ln p by least squares. If prices are held almost constant, e cannot be learned. This is why den Boer and Zwart [18] state that some deliberate price variation is necessary, and why Algorithm 1 keeps exploring through the sampled ensemble.

Step 2: Best price with limited stock. In step 11 of Algorithm 1 the quantity sold is capped by inventory, and leftover stock earns the salvage value:

$$V(p) = p\, q_s + v\,(I - q_s), \qquad q_s = \min\{q(p),\, I\} \tag{3}$$

Let p_I be the price at which demand equals the inventory:

$$p_I = \left(\frac{I}{A}\right)^{1/e} \tag{4}$$

Below p_I all stock sells out, so the value rises with price. Above p_I all demand is served, and setting the derivative of the value to zero gives the best price in this region:

$$p_u = \frac{e\, v}{1 + e} \tag{5}$$

For e = -2, p_u is exactly twice the salvage value v. The best price is the larger of p_u and p_I, kept inside the price floor and ceiling:

$$p^{*} = \min\{p_{\max},\ \max\{p_{\min},\ p_u,\ p_I\}\} \tag{6}$$

Because e is negative, lower inventory raises p_I, so the price rises when stock is scarce. This matches the result of Gallego and van Ryzin [3], discussed in Section 2.1.

Step 3: Numerical example. Take e = -2, A = 8000 and v = 10. Eq. (5) gives p_u = 20 for any inventory level.

Inventory I	p_u	p_I	p*	Units sold	V(p*)
30	20.00	16.33	20.00	20	500.0
10	20.00	28.28	28.28	10	282.8

With I = 30, stock is plentiful, so the best price is p_u: 20 units sell and 10 are salvaged, giving V = 20 x 20 + 10 x 10 = 500. With I = 10, demand at p_u (20 units) exceeds the stock, so the price rises to the clearing price 28.28 and all 10 units sell. The revenue uplift defined in Table 6 is

$$U = \frac{R_{\text{policy}} - R_{\text{baseline}}}{R_{\text{baseline}}} \tag{7}$$

For I = 10, a baseline that keeps the price at 20 sells only 10 units and earns 200, while p* = 28.28 earns 282.8, so U = (282.8 - 200) / 200 = 41.4%.

3.8 Example ML Models

Three models illustrate how the reference framework works in practice. They were run on simulated demand, because pricing data is mostly proprietary, so their numbers show how each method behaves and are not evidence from real markets. Models 1 to 3 correspond to Stage 4 (demand model), Stage 5 with unknown demand and Stage 5 with limited stock of Figure 2.

Model 1: Gradient-boosted demand forecast. This model predicts how much will sell at a given price. It builds many small decision trees one after another, and each new tree corrects the errors of the earlier ones. The inputs are day of week, lead time, holiday flag and the log of the price. A log-log line (Eq. 1) is also fitted to estimate price elasticity. The model is tested by rolling-origin validation, as in Section 3.3. In the simulation its WAPE was about 0.07, against 0.12 to 0.14 for a log-log baseline, and the estimated elasticity of -1.82 was close to the true value of -1.8. The model forecasts demand but does not choose prices, which is why a pricing stage is still needed.

Model 2: Thompson-sampling bandit. This model learns the best price while selling, when the demand curve is unknown. Each candidate price is an arm, and the model keeps a belief about the revenue each price earns. For each customer it draws one sample from each belief and offers the price with the highest draw. Uncertain prices are therefore tried from time to time, and prices that earn well are used more often. In the simulation it settled on the best price (30) for about 80% of customers. Because a poor trial price loses real sales, the price must stay inside the floor and ceiling of Stage 6.

Model 3: Q-learning with limited stock. This model chooses a price from the stock left and the time left. It plays many simulated selling seasons and records, for each situation, how much total revenue each price leads to. Over time it prefers the higher-earning prices, so price can follow scarcity, as in Eq. (4). In the simulation its average revenue (139.5) was close to the best fixed price (139.1) and slightly below the exact dynamic-programming optimum (141.8), and its choices at very low stock were not always optimal. This agrees with the finding in Section 4 that reinforcement learning needs a simulator and careful validation. Hybrid designs that pair a random-forest classifier with a deep Q-network have been used for context-aware decisions under changing load [32], and a similar split could separate demand-state detection from pricing.

IV. RESULTS AND DISCUSSION

This section synthesises the findings of the reviewed studies. It does not report new experiments.

4.1 Main Findings

Accurate demand estimation is necessary but not sufficient. Boosted trees and deep forecasters have improved demand estimation strongly [8], [15], [17], and gradient-boosted trees and neural networks produce the most accurate demand forecasts, but accurate forecasts alone do not give good prices.

Method choice follows the pricing problem. Bandits and reinforcement learning adapt prices best when demand is uncertain or changing [10], [11], while optimization-based methods depend on accurate demand input (Table 5). When demand is imprecise, fuzzy-logic nonlinear optimization [33] and neural-network-assisted nonlinear optimization of fuzzy models [34] offer ways to allocate limited capacity under uncertainty.

Exploration must be safe for the business. Some deliberate variation in price is needed to learn demand [18], but a poor exploratory price loses real sales, so safe exploration within price limits is needed.

Reported gains depend on simulation. Revenue gains depend heavily on simulation assumptions, and few studies test on real market experiments. Many RL pricing studies report gains in simulation that have not been confirmed in live experiments.

Evaluation needs forecast and business metrics together. Forecast accuracy alone is not sufficient, so policies are compared against a clearly stated baseline by A/B testing or off-policy evaluation instead of simulation alone.

Industry constraints shape the method. Retail pricing is framed around margin, promotions and competitor prices, while airline and hotel pricing are limited by fare classes and regulation, or by rate parity and cancellation rules (Table 4).

Competition, fairness and price-change limits are seldom built into pricing pipelines. Learning algorithms can sustain prices above the competitive level [13], personalized pricing can be seen as unfair [7], and competitor prices must be part of the demand model [11].

4.2 Open Problems

- **Realistic evaluation:** real market experiments, A/B tests and off-policy evaluation are needed in place of simulation alone, because RL usually needs a simulator and its results are sensitive to the simulator's assumptions.
- **Safe exploration:** exploring prices on real customers is costly, so exploration must stay within price floors, ceilings and step limits.
- **Competition and collusion:** competitor reactions and the risk of learned collusive prices need to be evaluated and built into pricing pipelines [11], [13].
- **Fairness and price-change limits:** fairness metrics and limits on price changes are seldom reported, and personalized pricing raises further fairness concerns [7].
- **Explainability:** pricing decisions need to be explainable to customers.

V. CONCLUSION AND FUTURE WORK

This survey linked demand estimation, price optimization and business guardrails in one framework across retail, airline and hotel pricing. It grouped the methods into a taxonomy, compared industries, and proposed a reference architecture and a forecast-guided pricing algorithm. The reviewed work shows that effective dynamic pricing needs accurate demand models, exploration that is safe for the business, and clear pricing rules. Future work should focus on realistic evaluation, fairness and explainable pricing decisions for customers.

REFERENCES

1. K. T. Talluri and G. J. van Ryzin, The Theory and Practice of Revenue Management. Springer, 2004.
2. L. Chen, A. Mislove and C. Wilson, "An empirical analysis of algorithmic pricing on Amazon Marketplace," in Proc. 25th Int. onf.

World Wide Web, 2016, pp. 1339-1349.
3. G. Gallego and G. van Ryzin, "Optimal dynamic pricing of inventories with stochastic demand over finite horizons," Management Science, vol. 40, no. 8, pp. 999-1020, 1994.
4. O. Besbes and A. Zeevi, "Dynamic pricing without knowing the demand function: Risk bounds and near-optimal algorithms," Operations Research, vol. 57, no. 6, pp. 1407-1420, 2009.
5. A. V. den Boer, "Dynamic pricing and learning: Historical origins, current research, and new directions," Surveys in Operations Research and Management Science, vol. 20, no. 1, pp. 1-18, 2015.
6. L. R. Weatherford and S. E. Kimes, "A comparison of forecasting methods for hotel revenue management," International Journal of Forecasting, vol. 19, no. 3, pp. 401-415, 2003.
7. G.-Y. Ban and N. B. Keskin, "Personalized dynamic pricing with machine learning: High-dimensional features and heterogeneous elasticity," Management Science, vol. 67, no. 9, pp. 5549-5568, 2021.
8. T. Chen and C. Guestrin, "XGBoost: A scalable tree boosting system," in Proc. 22nd ACM SIGKDD Int. Conf. Knowledge Discovery and Data Mining, 2016, pp. 785-794.
9. S. Pandikumar, S. B. Sethupandian, M. S. Saravanan, S. N. Prasad and M. Arun, "Deep learning based long short-term memory recurrent neural network for stock price movement prediction," Indian Journal of Science and Technology, vol. 15, no. 11, pp. 474-480, 2022, doi: 10.17485/IJST/v15i11.27.
10. K. J. Ferreira, D. Simchi-Levi and H. Wang, "Online network revenue management using Thompson sampling," Operations Research, vol. 66, no. 6, pp. 1586-1602, 2018.
11. A. Kastius and R. Schlosser, "Dynamic pricing under competition using reinforcement learning," Journal of Revenue and Pricing Management, vol. 21, pp. 50-63, 2022.
12. A. Kastius, N. Kiele and R. Schlosser, "Multi-agent dynamic pricing using reinforcement learning and asymmetric information," in Operations Research Proceedings 2022. Springer, 2023, pp. 557-563.
13. E. Calvano, G. Calzolari, V. Denicolo and S. Pastorello, "Artificial intelligence, algorithmic pricing, and collusion," American Economic Review, vol. 110, no. 10, pp. 3267-3297, 2020.
14. M. Sailaja, M. Prema Kumar, B. Swarna Jyothi, G. L. Narasamba Vanguri, S. Manjula and D. Divya Priya, "Remote sensing-based UAV imaging in heat pattern analysis impact on climate change detection using fuzzy stacked lasso elastic-net model," Remote Sensing in Earth Systems Sciences, vol. 7, no. 4, pp. 699-708, 2024, doi: 10.1007/s41976-024-00158-4.
15. S. Makridakis, E. Spiliotis and V. Assimakopoulos, "M5 accuracy competition: Results, findings, and conclusions," International Journal of Forecasting, vol. 38, no. 4, pp. 1346-1364, 2022.
16. C. Jamunadevi, N. Rajasekaran, S. Pandikumar, R. Raja and K. P. Narendra Prasanth, "Energy performance forecasting in buildings using machine learning," in Proc. Int. Conf. Communication, Computing and Internet of Things (IC3IoT), Chennai, India, 2024, pp. 1-6, doi: 10.1109/IC3IoT60841.2024.10550233.
17. B. Lim, S. O. Arik, N. Loeff and T. Pfister, "Temporal Fusion Transformers for interpretable multi-horizon time series forecasting," International Journal of Forecasting, vol. 37, no. 4, pp. 1748-1764, 2021.
18. A. V. den Boer and B. Zwart, "Simultaneously learning and optimizing using controlled variance pricing," Management Science, vol. 60, no. 3, pp. 770-783, 2014.
19. S. Pandikumar, S. A. Pallavi, V. M, M. Santhalakshmi and M. M. T, "Adaptive driver-passenger matching in on-demand mobility platforms using deep Q-networks," in Proc. Int. Conf. Sustainable Communication Networks and Application (ICSCN), Theni, India, 2025, pp. 261-266, doi: 10.1109/ICSCN67106.2025.11308611.
20. S. Pandikumar, B. Maram, C. Menaka, M. M. T and M. K. Rajan, "Policy gradient-based reinforcement learning for driver-passenger matching in ride-hailing platforms," in Proc. Int. Conf. Visual Analytics and Data Visualization (ICVADV), Tirunelveli, India, 2026, pp. 878-884, doi: 10.1109/ICVADV67766.2026.11470467.
21. B. Arivazhagan, S. Pandikumar, S. B. Sethupandian and R. S. Subramanian, "Pattern discovery and analysis of customer buying behavior using association rules mining algorithm in e-commerce," in Proc. Int. Conf. Electrical, Electronics, Information and Communication Technologies (ICEEICT), Trichy, India, 2022, pp. 1-5, doi: 10.1109/ICEEICT53079.2022.9768473.
22. K. Bhavana, R. Sumathi, K. Vanaja, S. Manjula, T. Srinivasulu and S. Divya, "Fruit ripeness detection using RipeNet," in Proc. Int. Conf. Electronics and Renewable Systems (ICEARS), 2025, pp. 1598-1604.
23. J. Gao, M. Le and Y. Fang, "Dynamic air ticket pricing using reinforcement learning method," RAIRO-Operations Research, vol. 56, pp. 2475-2493, 2022.
24. Y. Liu, K. L. Man, G. Li, T. Payne and Y. Yue, "Enhancing sparse data performance in e-commerce dynamic pricing with reinforcement learning and pre-trained learning," in Proc. Int. Conf. Platform Technology and Service (PlatCon), 2023, pp. 39-42.
25. S. Jo, G. M. Lee and I. Moon, "Airline dynamic pricing with patient customers using deep exploration-based reinforcement learning," Engineering Applications of Artificial Intelligence, 2024.
26. S. Saitta, V. D'Amico and G. Farinella, "Dynamic price prediction for revenue management system in hospitality sector," in Proc. 13th Int. Conf. Data Science, Technology and Applications (DATA), 2024, pp. 218-228.
27. G. U. Bhargavi, M. Balakrishna, H. Shivani, B. Bramhani and D. Pavithra, "Dynamic price optimization for ecommerce platform," in Proc. 1st Int. Conf. Research and Development in Information, Communication, and Computing Technologies (ICRDICCT), vol. 4, 2025, pp. 18-23.
28. S. Manjula, A. Bostani, A. Srinivasan, R. Sindoori, R. Madinakhon, B. Bozorov and U. Amirbek, "Nonlinear clustering optimization for scalable data mining in cloud and quantum computing environments," Results in Nonlinear Analysis, vol. 8, no. 3, 2025.
29. S. Pandikumar, A. Bostani, D. Sundaranarayana, I. Fakhriddin, D. Alimjon, U. Ilkhom and A. Juraboyev, "AI-driven nonlinear optimization for knowledge extraction and pattern analysis in IoT-enabled data mining systems," Results in Nonlinear Analysis, vol. 8, no. 3, pp. 36-46, 2025, doi: 10.31838/rna/2025.08.03.005.
30. S. Pandikumar, V. Pandimurugan, A. Rama Prasath, S. Karattil, S. C. B. Jaganathan and P. Vigneshwaran, "A novel framework for differential privacy based federated continual learning for dynamic medical data analysis," SN Computer Science, vol. 7, no. 6, p. 664, 2026, doi: 10.1007/s42979-026-05245-5.
31. S. Manjula, M. Sharmila Begum and P. Ilangovan, "HARCE: Human activity recognition in complex environments: An improved YOLOv8s model with Coyote and Badger Optimization for enhanced detection accuracy," International Journal of System Assurance

Engineering and Management, 2026.

32. P. Savarimalai, M. Chellappan, M. S. Nidhya, M. M. Thomas, S. Nallathambi and M. Selvaraj, "A context-aware intelligent scheduling framework for modern operating systems using hybrid machine learning models," Journal of Computational and Cognitive Engineering, vol. 5, no. 2, pp. 321-332, 2026, doi: 10.47852/bonviewJCCE62026664.
33. S. Pandikumar, A. Bostani, A. Srinivasan, G. Menaka, J. Chandhini, R. Sabitha and K. Sathishkumar, "Fuzzy logic-driven nonlinear optimization for resource allocation in 6G networks," Results in Nonlinear Analysis, vol. 8, no. 2, pp. 239-252, 2025, doi: 10.31838/rna/2025.08.02.019.
34. S. Manjula, S. Chitra, S. Ranjitha Kumari, R. Ramesh, K. Radhika, R. Naveenkumar and N. Radha, "Artificial intelligence and neural network-driven quantum calculus framework for nonlinear optimization of fuzzy partial differential equations in fluid dynamics," Results in Nonlinear Analysis, vol. 9, no. 1, pp. 7-16, 2026.