# APRS: A Unified Framework of Multi-Pricing Demand Simulation for Amazon Device Dependent Products

Yichao Li
Amazon Devices and Services
Seattle, WA, USA
yichaol@amazon.com

Priyanka Deshmukh
Amazon Devices and Services
New York, New York, USA
prideshm@amazon.com

## Abstract

Optimizing online pricing strategy for Amazon Device dependent products is a uniquely challenging topic in dynamic pricing of two-sided marketplaces when balancing the supply side economics and demand responsiveness. To address this challenge, we propose a scalable framework that integrates a hierarchical segmentation model with a sequential learning layer for high-velocity event (HVE) adjustment, to simulate counterfactual demand, business metrics, and price elasticity of demand (PED). Our approach utilizes a sequence deep learning model (LSTM) as a foundational learner for both structural simplicity and computational efficiency. To balance global robustness with segment pattern expertise, we implement a hierarchical segmentation of population-level and cluster-level models. Moreover, to address systematic under-forecasting during HVEs, we introduce a sequential adjustment layer that models demand residuals as a Gaussian process modulated by product lifecycle decay. By deploying this model pipeline within a rigorous simulation engine, we enable stakeholders to automatically architect pricing strategies tailored to inventory liquidation cycles, financial cost-benefit analysis, and optimized promotional scheduling. The framework is deployed on cloud services across multiple international marketplaces, covering core product lines, hundreds of product families, and thousands of products. By addressing both promotional events and business-as-usual (BAU) periods, the framework delivers high-granularity decision support for a global accessories portfolio.

## CCS Concepts

• **Computing methodologies** → **Machine learning**; **Modeling and simulation**; • **Applied computing** → **Economics**; *Marketing*.

## Keywords

Pricing Science, Demand Forecasting, Machine Learning, Deep Learning, Hierarchical Modeling, High Velocity Event

**ACM Reference Format:**

Yichao Li and Priyanka Deshmukh. 2026. APRS: A Unified Framework of Multi-Pricing Demand Simulation for Amazon Device Dependent Products. In *Proceedings of Workshop on Two-sided Marketplace Optimization:Search, Discovery, Matching, Pricing & Growth in conjunction with KDD Conference (KDD '26).* ACM, New York, NY, USA, 8 pages. https://doi.org/XXXXXXX.XXXXXXX



*KDD '26, Jeju, Korea*



ACM ISBN 978-1-4503-XXXX-X/2018/06
https://doi.org/XXXXXXX.XXXXXXX

## 1 Introduction

The Amazon Device dependent product ecosystem within a large-scale devices organization serves as a critical component in a two-sided e-commerce marketplace. It links the supply of a comprehensive suite of accessories and companion products with the demand from customers to complete the journey of purchasing compatible key devices (e-readers, smart speakers, streaming devices, and tablets). Since 2019 peak shopping event experiments first demonstrated the impact of strategic price optimization on profit, accessories pricing science has remained one of the business's largest optimization opportunities. Despite this potential, pricing device dependent products remains uniquely challenging because demand is determined by device-accessories interdependence, device compatibility constraints, and extreme volatility during high-velocity events (HVEs). To address these complexities, prior research has largely focused on three core areas: multi-horizon forecasting, price elasticity estimation, and peak event prediction.

**Multi-Horizon Demand Forecasting with Gaps.** Multi-horizon demand forecasting is a well established yet challenging domain with continuous methodological evolvement. Classical approaches leverage time series methods to capture temporal dependencies. The emergence of machine learning and deep learning [8] and [10] have expanded the methodological landscape, such as LSTM for demand forecasting and probabilistic demand forecasting with graph neural networks. More recently, large language models (LLMs) have been applied to multivariate time series forecasting tasks [16]. [6] introduces a forecasting approach ARCA that leverages inherent attach rate patterns to main devices, combined with residual prediction via Random Forest models. We also noticed the fusion method in deep learning and time series application are generally applied [12], addressing multi-modal data and model attributes. However, a unified workflow for the accessories business remains absent that captures cross device impact and enables multi-price demand simulation without upstream forecasting model dependencies.

**Pricing Elasticity of Demand Estimation.** Price elasticity of demand (PED) (see research in [11]) is a straightforward method to depict the incremental demand by price change. However, the estimation of PED exhibits significant challenges due to limited observability of counterfactual pricing scenarios, which is mentioned by [2]. Within Amazon, outside our business organization, there

are some ongoing working stream to address retail PED. Nevertheless, it is not customized for our business lacking critical business domain knowledge inputs to provide reliable recommendations.

**High Velocity Event Forecasting with Limited Data.** Amazon's signature and best deals promotional events, high velocity events (HVE, also known as Tier 1 events), bring up unique forecasting challenges. Although these events span short windows such as Prime Day (PD 2-4 days) and Prime Big Deal Day (PBDD 2 days), they drive peak sales representing a substantial share of annual sales and significantly impact traffic, conversion rates, and profitability. The approaches across Amazon, e.g. [17], have explored GNN-enhanced meta-learning frameworks that construct graph networks and exploit temporal sales trends to adapt to novel event dynamics. However, this solution is too complex to directly implement into our existing pipeline.

While existing literature offers solutions for forecasting and elasticity, these methods operate independently and are fundamentally unequipped to resolve the complex, interconnected nature of demand dynamics. In contrast, we introduce our unified framework, Accessories Pricing Recommendation System (APRS), to tackle the three primary objectives:

(1) **Demand forecasting for both business as usual (BAU) and full-tier promotion events:** Establishing BAU and high variance promotion event demand projections across multiple temporal horizons and lead-time gap chunks.
(2) **Multi-price (MP) inference simulation:** Developing a simulation engine capable of performing counterfactual analysis to predict demand variations under simultaneous pricing shifts in both the accessory and its compatible device.
(3) **Price elasticity of demand (PED) and metric estimation:** Utilizing the MP simulation framework to derive PED values and estimate incremental business metrics for various use cases.

Our contribution provides a scalable, computationally efficient solution (Figure 1) that integrates agnostically with existing promotion planning tools. The architecture combines sequence deep learning with a dedicated HVE adjustment layer, ensuring differentiable and transparent demand curves that support intuitive incremental analysis for financial and operational planning. The framework is validated across thousands of products spanning multiple international marketplaces during both promotional events and business-as-usual (BAU) periods, and the business team projects that the system will deliver a significant and measurable profit improvement improvement with high-quality pricing decisions.

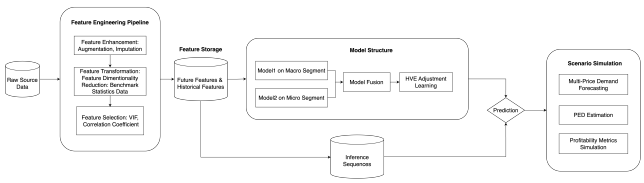


**Figure 1: Overview of APRS.**

## 2 Methodology

Our methodology is structured as a multi-stage training and inference pipeline that transitions from accessories demand estimation to multi-price counterfactual simulation. The framework begins with a hierarchical segmentation learning model (Section 2.1), utilizing RNN based sequence deep learning model, Long Short-Term Memory (LSTM) [9] to establish a robust foundation for demand prediction responding to pricing and other signals in respect to the sequence data. In addition, to address the systematic underforecasting of extreme demand spikes challenges in HVE demand forecasting due to *1) Data scarcity for extremes:* A standard two-year training window offers limited historical instances for each unique HVE; *2) Lack of quantified intensities:* Our feature set lacks robust numerical proxies for insufficiency of future marketing signals or future traffic intensity; *3) Optimization for the mean in foundational deep learning and machine learning models.* We integrate a sequential post-adjustment layer (Section 2.2), which applies Gaussian-based residuals and lifecycle decay factors to refine predictions during HVEs. Finally, this unified predictive engine is deployed within a multi-price simulation mechanism (Section 2.3), enabling the execution of granular "what-if" scenarios for incremental analysis and the robust estimation of PEDs through counterfactual analysis.

### 2.1 Hierarchical Segmentation Learning Approach

We propose a novel, scalable modeling framework that leverages hierarchical segmentation learning process (with late fusion method) to balance global robustness with local attributes sensitivity. This architecture addresses the trade-offs between data density at the marketplace product population level and patterns at the cluster level (Figure 2). We employ *LSTM* networks as a group of foundational estimators for demand forecasting system. It is motivated by the LSTM's capability to capture complex, non-linear temporal dependencies which are then mapped to a final differentiable linear output layer. Furthermore, the model architecture handles high-dimensional and multivariate feature spaces, enabling the simultaneous integration of diverse temporal signals. (see detailed feature engineering process in Appendix A.) In the Section 3.1, we have conducted analysis of horizontal model comparison. For a given product under configuration [1] attribute $c$, time $t$, the log-demand is denoted by $y_{c,t}$.

To optimize training at scale, a dual-level hierarchical structure is employed, see Figure 2. We utilize unsupervised clustering ($K$-means [1] or Agglomerative Clustering [15]) to segment unique attributes $c$. The clustering algorithm operates on temporal feature vectors extracted from both short-term rolling windows (7/14/28 days) and the entire demand time series, mapping each attribute to a distinct structural class. Then we have (1) Macro-segment model (All product population in single marketplace): Trained by all series to understand the cross-series interactions and shared patterns of demand distribution. This level benefits from high sample density,

[1] Accessories product configuration = compatible device+product category+program code name+normalized color. We choose configuration other than ASIN level because of noise information in attribution staging process. In 90% product configurations, it can 1:1 map to Accessories ASINs.

effectively managing cold-start because of new product launches and cross product lines shared patterns; (2) Micro-segment model (cluster level as pattern expert): Three clusters are identified as optimized clustering number to isolates localized demand signals and balance the model structure complexity (see Appendix D for patterns by clusters). For sparse attributes $c$ lacking sufficient time-series data in the training set for clustering analysis, the pipeline automatically falls back to the macro-level model during validation and inference.

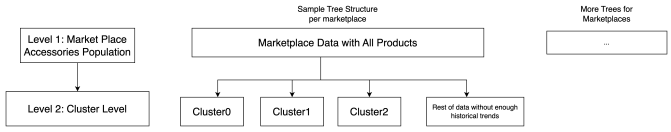


**Figure 2: Hierarchical Structure**

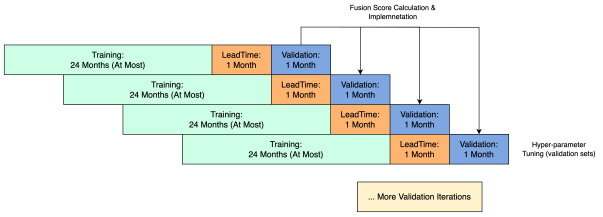


**Figure 3: Model Validation Framework**

We utilize a *time-series based cross-validation framework* in Figure 3, holding out a one-month lead-time gap chunk (excluding from the validation phase) to simulate real-world promotion and inventory planning cycles and prevent temporal data leakage. Model performance is quantified using two primary metrics: weighted mean absolute percentage error (wMAPE) and weighted bias (wBIAS). They are complementary, offering a robust assessment of forecast quality in observing forecasting errors for high-volume items and systematic drifting signs. For product configuration $c$ and time $d$, we have $\text{wMAPE} = \frac{\sum_{i=1}^{n_c}\sum_{d=1}^{n_d}\left|F_{i,d}-A_{i,d}\right|}{\sum_{i=1}^{n_c}\sum_{d=1}^{n_d}A_{i,d}}$ and $\text{wBIAS} = \frac{\sum_{i=1}^{n_c}\sum_{d=1}^{n_d}\left(F_{i,d}-A_{i,d}\right)}{\sum_{i=1}^{n_c}\sum_{d=1}^{n_d}A_{i,d}}$ where $F_{i,d}$ and $A_{i,d}$ represent forecasting and actual values respectively.

Next, in our experiment, we leverage a decision-level *late fusion* method [3] to combine predictions from the macro and micro models efficiently. Let $\mathcal{S} = \{\text{macro model, micro model}\}$ be the model set and let $m$ be the current validation month. We have the error $E_{i,m}$ for any model $i$ in model space $\mathcal{S}$ in trailing $k$ previous months (excluding current month) as $E_{i,m} = \frac{1}{k}\Sigma_{j=1}^{k}\left(NRMSE_{i,m-j}\right)$. Here, normalized root mean square error (NRMSE) is defined as $\frac{\sqrt{\frac{1}{n}\sum_{i=1}^{n}(F_{i,d}-A_{i,d})^2}}{\frac{1}{n}\sum_{i=1}^{n}A_{i,d}}$, which is a scale-independent, dimensionless error metric to ensure unbiased error comparison across disparate model spaces. Furthermore, it directly aligns with our training loss function (MSE). The total error of the model space as $E_m = \sum_{i\in\mathcal{S}} E_{i,m}$ and the fusion weight scheme follows an inverse error relationship principle. For each model, the weight is $W_{i,m} = 1 - \frac{E_{i,m}}{E_m}$. For $t \in m$ and any product, the final fused forecast is calculated as the weighted sum of the independent model outputs

$$F_{\text{fusion},m}(t) = \sum_{i\in\mathcal{S}} W_{i,m}F_{i,m}(t).$$

This approach dynamically considers the global robustness of the macro model with the unique distribution of the micro model, ensuring a reliable decision-making foundation.

## 2.2 Post-Adjustment Sequential Learning Layer with Decay Factor for High Velocity Events

We observed the persistent under-forecasting bias from Section 2.1 during HVE periods in evaluation by elevated forecast attainment value (metrics definition in Appendix B and visualization in Figure 4). To mitigate this bias, we introduce a simplified but powerful sequential ensemble learning layer that adjusts the hierarchical segmentation model output using historical residual patterns and a product lifecycle decay factor. We hold the following assumptions: (1) Temporal distribution: The residuals (between actual demand and fusion model prediction) follow a Gaussian or analogous distribution relative to the event peak day $t$. This aligns with the approximately symmetric, bell-shaped demand spikes observed during major HVEs (e.g., Prime Day); (2) Peak event accessory sales follow the device lifecycle with a decay pattern relative to the global maximum adjustment factor; (3) Stochastic dependency: Current adjustment factors to this year and specific event are modeled as a stochastic process with Markovian dependency on the prior year factors from same event, conditioned on the current lifecycle trajectory.

The adjustment factor $G(t)$ is defined in (1), which follows the Gaussian distribution assumption above:

$$\begin{aligned} G(t) &= (A_{event} - F_{fusion}(t))/F_{fusion}(t) \\ &= \alpha \cdot \exp\left(-\frac{1}{2}\left(\frac{t - t_{\text{event start}} - \mu}{\sigma}\right)^2\right) \end{aligned} \tag{1}$$

where $t_{\text{event start}}$ is the event start time. Additionally, to mitigate aggressive adjustments during the inference period, we introduce a three-stage piecewise decay function $D(t)$ that accounts for the product's position in its lifecycle, namely, Growth-Mature ($\frac{t-t_0}{t_1-t_0} < \tau_1$), Decaying ($\frac{t-t_0}{t_1-t_0} \in [\tau_1, \tau_2]$), End-of-Life (EOL) ($\frac{t-t_0}{t_1-t_0} > \tau_2$):

$$D(t) = \begin{cases} 1 & \text{if } \frac{t-t_0}{t_1-t_0} < \tau_1 \\ \exp\left(-k(\frac{t-t_0}{t_1-t_0} - \tau_1)\right) & \text{if } \frac{t-t_0}{t_1-t_0} \in [\tau_1, \tau_2] \\ \exp(-k(\tau_2 - \tau_1)) & \text{if } \frac{t-t_0}{t_1-t_0} > \tau_2 \end{cases}. \tag{2}$$

Here, (2) is parameterized using device launch time $t_0$, device EOL time $t_1$, current time $t$, and empirically derived constants $k, \tau_1, \tau_2$, see Appendix F for more details.

During training, we obtain fitted $\hat{G}(t)$ in (1) and fitted $\hat{D}(t)$ in (2). It follows that the global maximum peak adjustment factor $P_{\max}$ can be calculated by inverting the training decay: $P_{max} = \frac{\hat{G}(t)}{\hat{D}(t)}$. The final adjusted forecast $F_{\text{adjusted}}(t)$ is applied only during the lead-up, day-of, and lead-out periods of HVEs to make the adjustment effective in minimum days impacted. This sequential layer adds the estimated residual back into the fusion forecast $F_{adjusted}(t) = F_{fusion}(t) \cdot (1 + P_{max} \cdot \hat{D}(t))$. This approach ensures that while the base model provides the stable BAU and secondary promotion

events forecast, the sequential layer provides a targeted, lifecycle-aware correction for high-impact extreme events.

## 2.3 Multi-Price Simulation Mechanism & PED Estimation

Following model training and validation process in Section 2.1 and Section 2.2, we perform multi-price (MP) demand simulations in inference stage to address the counterfactual forecasting task. By holding exogenous covariates constant except pricing, we design a grid search across 17 accessory discount levels (0–80% in 5% steps) and 10 device levels (0–45% in 5% steps), generating approximately 5,100 scenarios per Country+Product Config+Day granularity in each simulation month. Given that true elasticity is unobservable (see details in [2]), we utilize the simulated MP demand to estimate incremental demand, PED and other key business metrics, including profit, and promotion ROI in Appendix B

Given certain device price $P_d$ and time $t$, to derive accessory $\text{PED}_{accy}$ from baseline demand (Accessories maintaining MSRP), we have both derivatives and extended logarithm definitions:

$$\text{PED}(P_a, Q_a; P_d, t, c) = \frac{d(Q_a)/Q_a}{d(P_a)/P_a} = \frac{dlog(Q_a)}{dlog(P_a)}.$$

By the logarithm definition, PED is the first order derivative of a continuously differentiable function $f$ where $log(Q) = f(log(P))$. For each MP scenario, we can derive $\text{PED}_{accy,mp}$ by the Lagrange's mean value theorem. The MP scenario is trying to get the incremental demand change from manufacturer's suggested retail price (MSRP) $P_{a,msrp}$. For each iterated accessory price $P_{a,mp}$, in particular, given an interval $[P_{a,msrp}, P_{a,mp}]$, there exists some $P_{a,\theta}$ in $(P_{a,msrp}, P_{a,mp})$ such that

$$\begin{aligned} f'(\log(P_{a,\theta})) &= \frac{f(\log(P_{a,mp})) - f(\log(P_{a,msrp}))}{\log(P_{a,mp}) - \log(P_{a,msrp})} \\ &\approx \frac{d\log(Q_a)}{d\log(P_a)} = \text{PED}(P_a, Q_a; P_d, t, c). \end{aligned} \tag{3}$$

At this stage, we have derived multiple PEDs. For certain financial analysis use cases, a constant PED is more convenient. Our objective is for the constant PED to serve as a reliable proxy for APRS when estimating demand lift relative to BAU (MSRP) demand. According to PED definition, we have PriceEffect = PED $\cdot \left[log\left(P_{a,mp}\right) - log\left(P_{a,msrp}\right)\right]$, such that $\text{Lift}_{\text{PED}} = e^{\text{PriceEffect}}$. At the same time, when we calculate Lift using APRS, given device price $P_d$, we have $\text{Lift}_{\text{APRS}} = \frac{f(P_{a,mp})}{f(P_{a,msrp})}$, where $f$ is the function of APRS model pipeline. To make the $\text{Lift}_{\text{PED}}$ close to $\text{Lift}_{\text{APRS}}$, we employ the searching algorithm to minimize the difference, defined as mean absolute percentage difference (MAPD) as

$$\arg\min_{\text{mp}}(\text{MAPD}) = \arg\min_{\text{mp}} \left( \frac{1}{n} \sum_{i=1}^{n} \frac{|\text{Lift}_{\text{PED}} - \text{Lift}_{\text{APRS}}|}{\text{Lift}_{\text{APRS}}} \right).$$

# 3 Experiments & Results

## 3.1 Hierarchical Modeling Results

Experiments were conducted on core US PL data (February 2023 – October 2025), with performance measured by wMAPE and wBIAS at both daily and weekly granularities (April – October 2025), following the feature space with logarithm transformation in Appendix A and evaluation mechanism mentioned in Section 2.1. We evaluated a suite of linear regressors, *Bayesian Ridge* [13], *Tweedie* [14], *XGBoost Linear* [4], *ElasticNetCV* [18] , and deep learning models for time series forecasting *LSTM*[9] [8] [7] and *GRU*[5]. Tree-based ensembles were intentionally excluded because the piecewise constant nature produces discontinuous and step-wise demand simulation curves. We compared multiple modeling strategies: (1) One-model-fits-all-data; (2) Hierarchical segmentation structure only; and (3) Hierachical segmentation model with late fusion method. Under strategy (1), *LSTM model* achieved optimal performance, balancing predictive accuracy and directional stability (Table 1 One model). The hierarchical segmentation approach in strategy (2) improved all models across both metrics (Table 1 Hierarchy with segment only. We selected best performing power for Tweedie related models). Among them, *LSTM* still maintained robust performance by reducing wMAPE by 80 bps [2] (daily) and 50 bps (weekly) and improving wBIAS by 140 bps. Notably, *XGBoost Linear and Bayesian Ridge*, which underperformed in strategy (1), showed substantial gains under strategy (2), demonstrating the framework's adaptability in ML regressors. After adding fusion method on the hierachical segmentation structure in strategy (3) (Table 1 Hierarchy segmentation with fusion), deep learning models continue increased performance in wMAPE and stayed neutral to wBIAS against strategy (2). The LSTM hyperparameters are optimized with structure outlined in Appendix C. Specially, in Table 2, we highlight representative product categories in benefit of fusion method by accuracy improvement (weekly level for LSTM). However, strategy (3) does not benefit ML regressors, as the weaker macro model dilutes the fusion effectiveness.

## 3.2 HVE Adjustment Impact & Final Performance

Prior to applying the HVE adjustment, the hierarchical segmentation model with fusion (strategy (3)) exhibited systematic underforecasting during high-volume events, particularly July 2025 (PD) and October 2025 (PBDD), on the same validation set used in Section 3.1. These months showed high forecast attainment (Appendix B for metric definitions) of 140% and 145%, respectively, alongside substantial wBIAS values and elevated wMAPE as shown in Figure 4, Figure 5, and Figure 6 as "before" scenarios . This pattern indicated the need for event-specific correction. Following the Gaussian adjustment with decay factor described in Section 2.2, we observed targeted improvements exclusively in the HVE months (July and October), as shown in Figure 4, Figure 5, and Figure 6 as "after" scenarios. The adjustment reduced attainment by 4300 bps in July and 700 bps in October. The impact on wBIAS was particularly pronounced. In July, wBIAS improved from -29% to 5%, nearly eliminating systematic bias. In October, wBIAS improved by 500 bps reduction as of -25%. Regarding wMAPE, July experienced a modest 200 bps decrease, which is acceptable given the substantial wBIAS improvement. October showed a 300 bps wMAPE reduction, demonstrating improvement across both metrics. Overall, the HVE adjustment method substantially improved attainment and wBIAS

[2]Basis points (bps) are used throughout this paper to express percentage point differences. One basis point equals 0.01 percentage points. This notation eliminates ambiguity between relative percentage changes and absolute percentage point differences.

**Table 1: Model Performance by Different Strategies**

| Str. | Model | Daily | | Weekly | |
|---|---|---|---|---|---|
| | | wMAPE | wBIAS | wMAPE | wBIAS |
| OneModel | **LSTM** | **50.0%** | **-20.6%** | **39.2%** | **-20.6%** |
| | GRU | 50.2% | -24.5% | 42.3% | -24.5% |
| | XgbLinear | 50.1% | -34.7% | 44.9% | -34.7% |
| | BayesianRidge | 50.3% | -35.2% | 45.1% | -35.2% |
| | TweedieRegressor_power=1 | 58.5% | -44.4% | 54.4% | -44.4% |
| | ElasticNetCV | 62.2% | -49.7% | 58.4% | -49.7% |
| | TweedieRegressor_power=1.5 | 61.4% | -47.5% | 57.5% | -47.5% |
| | TweedieRegressor_power=2 | 64.0% | -50.6% | 60.3% | -50.6% |
| | XgbLinearTweedie_power=1 | 65.1% | -50.1% | 61.3% | -50.1% |
| | XgbLinearTweedie_power=1.5 | 68.0% | -54.0% | 64.5% | -54.0% |
| | TweedieRegressor_power=0 | 67.8% | -52.2% | 64.0% | -52.2% |
| Hierarchy(SegOnly) | **LSTM** | **49.2%** | **-19.2%** | **38.7%** | **-19.2%** |
| | XGBLinear | 47.0% | -21.1% | 39.1% | -21.1% |
| | BayesianRidge | 47.9% | -21.8% | 39.0% | -21.8% |
| | GRU | 50.4% | -22.2% | 41.4% | -22.2% |
| | Tweedie | 57.3% | -37.1% | 52.7% | -37.1% |
| | ElasticNetCV | 60.2% | -41.2% | 55.5% | -41.2% |
| | XGBLinearTweedie | 61.0% | -40.4% | 59.0% | -40.4% |
| Hierarchy(Seg+Fusion) | **LSTM** | **48.6%** | **-19.9%** | **38.1%** | **-19.9%** |
| | GRU | 48.7% | -23.4% | 40.9% | -23.4% |
| | XGBLinear | 48.0% | -27.8% | 40.1% | -27.8% |
| | BayesianRidge | 48.8% | -28.4% | 40.1% | -28.4% |
| | Tweedie | 57.0% | -40.9% | 52.5% | -40.9% |
| | XGBLinearTweedie | 63.2% | -45.9% | 59.1% | -45.9% |
| | ElasticNetCV | 60.0% | -45.6% | 55.7% | -45.6% |

**Table 2: Fusion Model Impact Highlights**

| PL | Category | Micro | | Macro | | Fusion | |
|---|---|---|---|---|---|---|---|
| | | wMAPE | wBIAS | wMAPE | wBIAS | wMAPE | wBIAS |
| D | Cat-D1 | 40.9% | 3.3% | 42.6% | -15.3% | 33.8% | -6.5% |
| D | Cat-D2 | 45.0% | -21.2% | 38.5% | -14.6% | 33.3% | -17.7% |
| A | Cat-A1 | 46.9% | -12.4% | 40.8% | -7.5% | 37.9% | -9.5% |

and maintaining neutral impact on wMAPE. In above figures mentioned, all metrics reported at daily aggregation to capture minor performance changes for HVE adjustment impact. Zooming into the impact on the finest data granularity of product configuration, we presented a representative sample product from PBDD. On October 7th–8th, actual sales surged to peak values. While the original fusion model predicted elevated demand, a substantial gap remained. The HVE-adjusted predictions successfully captured these event-driven outliers, aligning more closely with observed actuals, see Figure 7.

In summary, final model performance shows in Table 3. It demonstrates substantial improvement over the hierarchical segmentation with fusion in *LSTM* (results in Section 3.1): daily wMAPE of 47.3% (130 bps improvement vs 48.6% ) with wBIAS improved to -7.4% (1250 bps improvement vs -19.9% ); weekly wMAPE of 37.5% (neutral vs -38.1% ), with wBIAS consistent at -7.4% across both aggregations due to the calculation method properties. PL level breakdown is also exhibited.

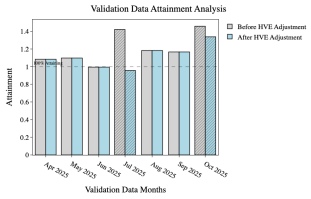


**Figure 4: Attain.**

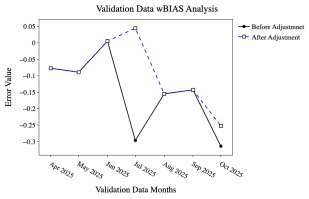


**Figure 5: wBIAS**

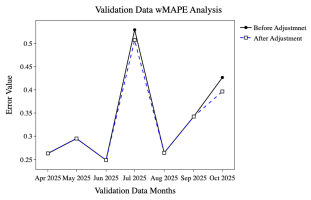


**Figure 6: wMAPE**

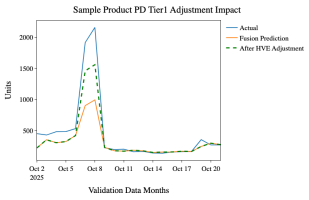


**Figure 7: SampleProduct**

**Table 3: Final Model Performance**

| Product Line | Daily Aggregation | | Weekly Aggregation | |
|---|---|---|---|---|
| | wMAPE | wBIAS | wMAPE | wBIAS |
| A | 48.7% | -5.1% | 37.4% | -5.1% |
| B | 45.0% | -2.1% | 35.0% | -2.1% |
| C | 39.4% | -4.8% | 25.7% | -4.8% |
| D | 49.4% | -14.1% | 41.6% | -14.1% |
| **Summary** | **47.3%** | **-7.4%** | **37.5%** | **-7.4%** |

## 3.3 Simulation Analysis

In the simulation analysis, our interests locate in the PED properties and counterfactual scenarios in business metrics for use cases in Section 2.3. We continue focusing on the representative product in Section 3.2 during inference period spanned October and November 2025 (see more products simulation analyses in Appendix E). We first examined whether accessory PEDs exhibits significant variance across different device discount levels. Both the PED distribution plot (Figure 8) and variance plot (Figure 9) demonstrated that PED values rapidly converge to a constant after the 25th percentile of iterated accessory discount levels, suggesting this percentile serves as a robust baseline candidate, remaining stable across time periods and device discount variations. Additionally, absolute PED values decrease monotonically as discount percentiles increase. Next, we conducted lift analysis to identify which PED percentile best approximates the APRS original lift predictions by the method in Section 2.3. As shown in Figure 10, the lift MAPD by discount level percentile exhibits a declining trend, reaching a local minimum at the 75th percentile as a good proxy to derive constant PED. In Appendix E, we compare more scenarios on constant PED value selection for Lift trend. Finally, Figure 11 the trade-off plot displays the demand units-profit visualization. Our modeling system successfully captures the inverse relationship between demand units and profit across multiple accessory pricing points. Using APRS original predictions to estimate profit, we identify an optimal profit sweet spot at approximately 5% discount for the sample product.

While PED-estimated profit and units directionally align with APRS model insights, they exhibit some deviation in identifying the local optimal profit range, with slight shifts relative to APRS original predictions. However, PED estimation offers a significant advantage: it provides a rapid and flexible approach to extrapolate demand and profitability for any discount level within the observed accessory discount bounds (0–80%).

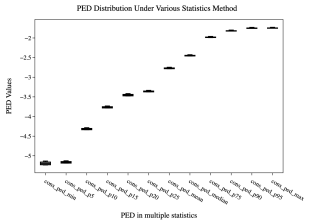


**Figure 8: PED Dist.**

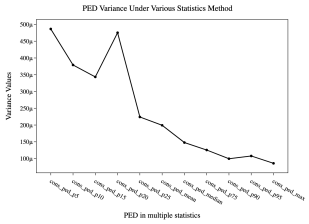


**Figure 9: PED Var.**

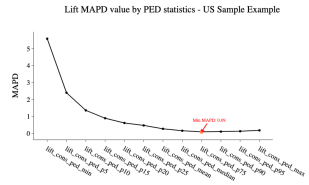


**Figure 10: Lift MAPD**

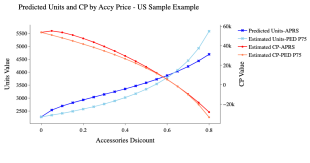


**Figure 11: Trade-off**

## 4 Conclusion & Future Work

In this paper, we introduced APRS, a scenario simulation application powered by hierarchical segmentation models using LSTM and Gaussian sequential learning modeling layer. This application achieves function of dealing with complex signals for demand forecasting and pricing elasticity estimation and generating smooth simulation curves and reasonable PEDs. In various two-sided marketplace business use-cases, it provides stable and actionable decision support across supply management, promotion planning, and finance cost-benefit analysis. APRS is productionized through an integrated data and model pipeline in AWS services. Inference outputs are persisted to Amazon S3 and production reporting table in Redshift, flowing into QuickSuite dashboard and AI agent environment for insightful visualization and narratives generation. Overall, the model system achieved weekly wMAPE as 37.5% and -7.4% wBIAS across largest marketplace US and including all types of promotion deal events. Besides, we observed the limitations subject to historical feature bias, where the model could over-index on prior promotional regimes rather than emerging market shifts or marketing campaign intensity. The framework suggests incorporating more market signals by continuous monitoring for model drift and periodic retraining to ensure that the system adapt to evolving consumer behavior and hardware lifecycles in the evolving retail landscape. Future development of the APRS framework will prioritize more business entities expansion and bundles promotion strategies. We further aim to benchmark our current baseline against transformer based foundation models for time-series forecasting use cases to enhance zero-shot predictive capabilities for enlarged portfolio.

## Acknowledgments

We thank our former colleague Jing Sun for early experimentation and proof-of-concept work that helped shape the foundation of this research.

## References

[1] David Arthur and Sergei Vassilvitskii. 2006. *k-means++: The Advantages of Careful Seeding*. Technical Report 2006-13. Stanford InfoLab. http://ilpubs.stanford.edu:8090/778/

[2] Richard W Blundell, Dennis Kristensen, and Rosa Liliana Matzkin. 2017. *Individual counterfactuals with multidimensional unobserved heterogeneity*. Technical Report. cemmap working paper.

[3] Wided Bouchelligua. 2025. PatternFusion: a hybrid model for pattern recognition in time-series data using ensemble learning. *Scientific Reports* 15, 1 (2025), 43371.

[4] Tianqi Chen. 2016. XGBoost: A Scalable Tree Boosting System. *Cornell University* (2016).

[5] Kyunghyun Cho, Bart Van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. 2014. Learning phrase representations using RNN encoder-decoder for statistical machine translation. *arXiv preprint arXiv:1406.1078* (2014).

[6] Kiki Dimitriadou, Vicky yu, Michael Behrman, Lihong Xie, Kushagra Bhushan, and Lynn Teresi. 2025. ARCA: Forecasting demand for device accessories at Amazon. In *KDD Conference Workshop AI for Supply Chain*. https://www.amazon.science/publications/arca-forecasting-demand-for-device-accessories-at-amazon

[7] Taha Falatouri, Farzaneh Darbanian, Patrick Brandtner, and Chibuzor Udokwu. 2022. Predictive analytics for demand forecasting–a comparison of SARIMA and LSTM in retail SCM. *Procedia Computer Science* 200 (2022), 993–1003.

[8] Marta Gołąbek, Robin Senge, and Rainer Neumann. 2020. Demand Forecasting using Long Short-Term Memory Neural Networks. *arXiv preprint arXiv:2008.08522* (2020).

[9] Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural computation* 9, 8 (1997), 1735–1780.

[10] Nikita Kozodoi, Elizaveta Zinovyeva, Simon Valentin, João Pereira, and Rodrigo Agundez. 2024. Probabilistic demand forecasting with graph neural networks. *arXiv preprint arXiv:2401.13096* (2024).

[11] Chengcheng Liu and Mátyás A Sustik. 2021. Elasticity based demand forecasting and price optimization for online retail. *arXiv preprint arXiv:2106.08274* (2021).

[12] Lei Ni, Zhonglin Huang, and Ning Fu. 2025. A Stacking-Based Fusion Framework for Dynamic Demand Forecasting in E-Commerce. *Mathematics* 13, 21 (2025), 3436. doi:10.3390/math13213436

[13] Michael E. Tipping. 2001. Sparse Bayesian Learning and the Relevance Vector Machine. *Journal of Machine Learning Research* 1 (2001), 211–244. https://www.jmlr.org/papers/volume1/tipping01a/tipping01a.pdf

[14] Maurice C. K. Tweedie. 1984. An Index which Distinguishes between some Important Exponential Families. *Statistics: Applications and New Directions. Proc. Indian Statistical Institute Golden Jubilee International Conference* (1984), 579–604.

[15] Joe H. Ward. 1963. Hierarchical Grouping to Optimize an Objective Function. *J. Amer. Statist. Assoc.* 58, 301 (1963), 236–244. doi:10.1080/01621459.1963.10500845

[16] Malcolm L Wolff, Shenghao Yang, Kari Torkkola, and Michael W Mahoney. 2025. Using pre-trained llms for multivariate time series forecasting. *arXiv preprint arXiv:2501.06386* (2025).

[17] Zexing Xu, Linjun Zhang, Sitan Yang, and Nan Jiang. 2024. Peak period demand forecasting with proxy data: GNN-enhanced meta-learning. *WACV 2024 Workshop on Physical Retail AI* (2024). https://www.amazon.science/publications/peak-period-demand-forecasting-with-proxy-data-gnn-enhanced-meta-learning

[18] Hui Zou and Trevor Hastie. 2005. Regularization and Variable Selection via the Elastic Net. *Journal of the Royal Statistical Society Series B: Statistical Methodology* 67, 2 (2005), 301–320. doi:10.1111/j.1467-9868.2005.00503.x

## A Features Engineering

We extracted features from various distinct data sources to capture the key drivers of demand across domains: product metadata, promotional deal events, historical sale series, price information, traffic intensity, supplier constraints, customer reviews, and inventory health. Our engineering framework contains two sequential phases.

*Phase I: Data Enhancement and Temporal Synchronization.* To capture the dynamics of demand, we applied moment-based statistical functions and quantile mappings to expand univariate temporal

data into comprehensive location and scale statistics. This was performed at both daily and monthly granularities to reveal short-term signals and long-term benchmarks by extracting the statistics moments as mean, median, sum, and min operations. Furthermore, we engineered ratio-based features to quantify structural business distributions. Given the high data integrity (missing values <5%), we utilized brief forward-filling imputation without introducing looking ahead bias or leakage.

*Phase II: Feature Space Construction and Dimensionality Reduction.* Numerical variables are logarithmic-transformed to normalize right skewness, and categorical features are mapped via one-hot encoding. To resolve multicollinearity and high-dimensional redundancy for stable causal estimation, we implemented a multi-stage filtering pipeline. We applied variance thresholding to eliminate low-information features, followed by linear Principal Component Analysis (PCA) on the benchmark monthly statistics. Then, we utilized a Variance Inflation Factor (VIF) filter with a threshold of 5, complemented by correlation analysis relative to the target variable.

This systematic approach yielded a comprehensive and balanced set of future and historical features

**Future features**

- Product metadata: We collected Accessories designed attributes including product line, category, color, compatible devices, internal program code name to capture the attributes as much as we can. For these key attributes, we applied one-hot encoding. To better understand product life cycle, we collect device launch date, End-Of-Life timeline as product lifetime, and derive the device age, device life stage.
- Promotion deal events & Seasonality: Amazon devices have deal event calendar aligning the e-commerce retail events (Mother's day, Father's day, Back to school, Black Friday) and unique Amazon sales events (Prime Day, Prime Big Deal Day). Derive the next, previous events, and the length between them.
- Pricing information: MSRP, discounted price, discounted amount for both Accessories and compatible devices.

**Historical features** We collect the historical features from the month before lead time $l$ (aka the gap chunk) to both capture the historical trends as benchmark performance and avoid possible leakage of future features.

- Sales: Device and dependent products historical sales volume by multiple merchant types.
- Device and dependent products price info in historical windows.
- Traffic: Glance views on the online detail pages.
- Customer review: review star count, low star review count.
- Inventory level: on-hand-inventory, weeks of cover, supplier constraints.

## B More Metrics Definition & Calculation

Here we use $Q$ to denote device dependent products Demand, and $P$ to denote device dependent products Price

**Table 4: Hyperparameter Search Space**

| Hyperparameter | Search Space |
|---|---|
| Window size ($w$) | 3, 5, 7, 14, 28 |
| Hidden size ($h$) | 32, 64, 128 |
| Batch size | 32, 64, 128, 256 |
| Learning rate ($\eta$) | $10^{-3}$, $5 \times 10^{-4}$, $10^{-4}$ |
| Dropout rate ($p$) | 0.1, 0.2 |

**Incremental Demand**

$$\Delta Q = Q_{mp} - Q_{msrp}$$

**Profit: Contribution Profit**

We have the CPPU as marginal historical contribution profit per unit

$$\text{CPPU} = P - \text{historical monthly average cost}$$

such that

$$\text{CP} = \text{CPPU} \cdot Q$$

$$\text{Incremental CP} = \text{CP}_{mp} - \text{CP}_{msrp}$$

**Promotion ROI**

$$\text{ROI} = \text{Incremental CP/Discount Promotion Cost}$$

**Model Forecast Attainment**

$$\text{Overall Forecast Attainment} = \frac{\sum_{i=1}^{n_c} \sum_{d=1}^{n_d} A_{i,d}}{\sum_{i=1}^{n_c} \sum_{d=1}^{n_d} F_{i,d}}.$$

In the finance forecast interpretation task, forecast Attainment serves as a macro-level performance indicator, quantifying the total volume of realized demand relative to the projected plan in evaluation period. Attainment identifies systemic volume achievement gaps.

## C LSTM Architecture and Parameters

We employ a single-layer LSTM with tanh activation, taking a sliding window of $w \in \{3, 7\}$ timesteps as input, where each timestep is a feature vector $\mathbf{x}_t \in \mathbb{R}^{n_f}$ ($n_f \approx 384$–$391$ after preprocessing). The short window is sufficient because upstream feature engineering already encodes historical aggregates, rolling statistics, and seasonal indicators into each observation. We choose tanh over sigmoid for its zero-centered output, which better suits the standardized log-transformed target space. The architecture consists of a single LSTM layer ($h = 64$, tanh), followed by dropout, a fully connected layer (32, tanh), dropout, and a linear output layer, totaling approximately 130,000 trainable parameters. Training uses AdamW with weight decay $\lambda = 10^{-2}$, 20 epochs, MSE loss, and FP16 mixed-precision acceleration. For the segmentation-based variant, hyperparameters are tuned per level and cluster via random search (Table 4).

## D Clusters Analysis

We applied K-Means clustering ($K = 3$) to log-transformed and standardized daily sales time series. The PCA projection (Figure 12) confirms clear separation between clusters along the first two principal components, validating the segmentation as a meaningful basis for hierarchical model specialization. The resulting cluster demand patterns (Figure 13) reveal three distinct segments:

- **Cluster 0**: Products with moderate and stable demand patterns showing signals of deal event-driven peaks.
- **Cluster 1**: Products exhibiting higher volatility and most intense response aligning Amazon HVEs promotion events.
- **Cluster 2**: Products with consistently low baseline demand and low variance across time.

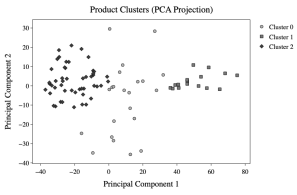


**Figure 12: Cluster Segments**

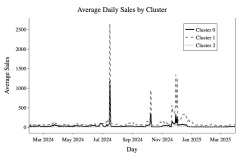


**Figure 13: Sales Trend**

## E Extended Analysis for PED and Impact

In Figure 14, we compared the 25th, median, and 75th percentile PED lifts against APRS original lift. Consistent with PED theory, larger absolute PED values (corresponding to lower discount percentiles) produce greater lift impacts relative to baseline demand at MSRP. These three representative PED percentiles serve distinct strategic purposes. Business teams with specific discount intensity expectations can select the corresponding PED as an appropriate lever. For applications requiring a single PED across the entire discount range (0–80%), we recommend using the 75th percentile value.

In Figure 15 and Figure 16, we exhibited Product Line D sample program PED distribution properties. It aligns the trend and observations we have in Section 3.3. In Figure 17, we showed the Median PED distribution across all PLs in US.

## F Accessories Sales Life Cycle by Device Life Index & Decay Pattern

**Decay factor threshold selection:** As we mentioned in Section 2.2, we selected the decay threshold from empirical distribution per PL (PL-A as example here), where only keep MSRP period for both Device and Accessories to track the organic sales as we can to understand the decay pattern. In both bar charts, the x axis are life time mid point value in each bin. From the threshold corresponding to approximately two-thirds of peak normalized sales (over the entire accessory product life), we observed the long tail pattern appears. This is where $\tau_1$ happens. And we know the information when Devices become EOL from product configurations, where $\tau_2$ happens. The exponential decay rate $k$ is determined by $\tau_1$ and $\tau_2$. To ensure the residual demand reaches a predefined negligible floor $\epsilon$ at the onset of the EOL stage, we derive $k = -\frac{\ln(\epsilon)}{\tau_2 - \tau_1}$. where

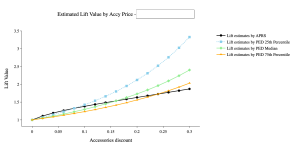


**Figure 14: Lift Impact**

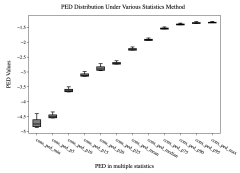


**Figure 15: Program Sample**

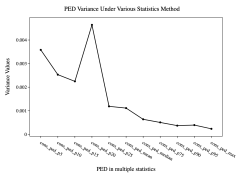


**Figure 16: PED Var Sample**

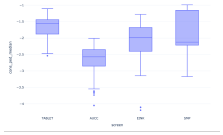

**Figure 17: PED Dist. by PL**

$\epsilon$ represents the target convergence threshold (here we use $10^{-2}$), ensuring that the decay function $D(t)$ smoothly transitions to a stable, low-variance state upon reaching $\tau_2$. These threshold might be conservative cause for individual product, the decay might happen earlier. But we keep this to give ambitious HVE impact due to nothing else signal we can get.

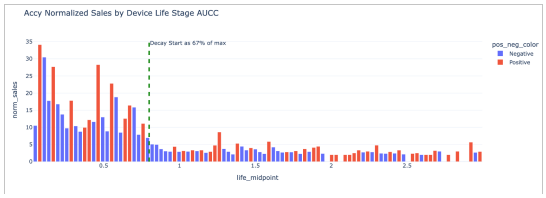


**Figure 18: Lift Impact**

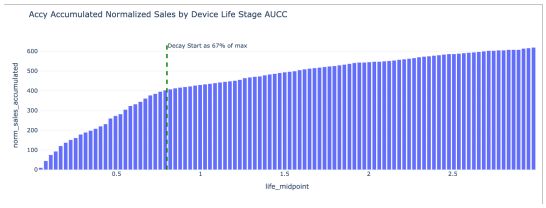


**Figure 19: PED Distribution-Tablet**