ISSN 2278-2540 | DOI: 10.51583/IJLTEMAS | Volume XV, Issue V, May 2026

Dynamic Price Allocation and Optimization for E-Commerce Platforms Using Reinforcement Learning and Deep Learning

Deepak Muvva, B.Sudheer Babu, V.Krishnateja, SK.Ayesha Tahseen, Ch.Bharadwaja

Department of Advanced Computer Vignan's Foundation for Science, Technology and Research Vadlamudi. Guntur,Andhra Pradesh,522 213

DOI: https://doi.org/10.51583/IJLTEMAS.2026.150500043

Received: 30 April 2026; Accepted: 05 May 2026; Published: 26 May 2026

ABSTRACT

The concept of dynamic pricing has already become one of the most significant aspects of electronic commerce, which is constantly changing in terms of the level of demand and competition, as well as the response of customers to a specific product (or service). Conventional methods of pricing and reinforcement learning methods like Deep Q-Networks (DQN) tend to have restricted flexibility, dis- crete action, and no proper estimation of demand. Our uncertainty-aware dynamic pricing framework as offered in this paper incorporates a hybrid demands forecasting, Transformer-LSTM demand forecasting model and Soft-Actor-Critic (SAC) reinforcement learning to optimize prices continuously. The Trans- former component models the long-range time interdependencies whereas the LSTM models the sequential nature of demand patterns and allows it to predict the demand robustly and precisely. The state representation of the SAC agent, which learns the optimal pricing policies under dynamic market, takes these forecasts into consideration.

The suggested system is implemented on a scalable, API-focused system of microservices and allows making real-time pricing decisions. Online Retail II Evaluation The experimental analysis of the Online Retail II data reveals that the improvement of experimental approaches is substantial as compared to the baseline techniques. Demand forecasting with the model has an R 2 of 0.62 with a Mean Absolute Percentage Error (MAPE) of 8.7% and one can increase the revenue by 21.4% and the profit by 18.2% over the expected traditional methods of reinforcement learning.

The findings demonstrate the efficacy of melding cutting-edge deep learning and reinforcement learning approaches to scal- able, adaptable, and smart pricing in the practical e-commerce setting.

Keywords—Dynamic Pricing, Reinforcement Learning, Transformer, Long short term memory (LSTM), Soft actor critic (SAC), Demand forecasting, e-commerce, time series prediction, Deep Learning, price optimization.

INTRODUCTION

Dynamic pricing is now an essential feature of e-commerce platforms in the mod- ern period, which demand their prices to constantly adjust to the fluctuating demand, competition, and customer behaviour. The formal or traditional pricing methods that are either static or rule based are not sufficient, since they do not reflect nonlinearities and swiftly changing market environments.

Establishing pricing as a sequential decision-making problem has recently become possible thanks to recent progress in data-driven methods, in particular deep reinforcement learning (DRL). In this model an agent is in contact with the market environment and learns how to optimise long-term cumulative rewards. 2DQN or Soft Actor-Critic (SAC) algorithms have shown great promise in dynamic pricing procedures.

Liu et al. [1] showed the usefulness of DRL on e-commerce applications at large-scales by optimizing the discrete pricing to a continuous action and providing enhanced reward formulations. In the same manner, a DRL -based pricing scheme as developed by Yin and Han [2] was able to reach almost optimal equi- libriums

ISSN 2278-2540 | DOI: 10.51583/IJLTEMAS | Volume XV, Issue V, May 2026

strategies, when market conditions changed. Sun et al. [3] also demonstrated that Double DQN is more profitable than the conventional DQN strategies.

Although such contributions have been made, reinforcement models of learning tend to be limited in perspective since it utilizes more previous interactions. The right kind of demand forecasting is thus.

CRM: necessary to bolster the pricing decisions. Time-series forecast- ing has been highly applicable with deep learning models like the LSTM, because they can effectively model temporal relationships.

We will consider a hybrid dynamic pricing frame- work based on the Transformer-LSTM demand forecasting model and the Soft Actor-Critic reinforcement learning agent in this paper. The forecasting system enables predictions of the demand, whereas SAC agent makes the best decisions of the prices in the continuous action space. This integration will allow the system to attain better adaptability, stability and optimization of revenues in reality e-commerce settings.

One of them is: A Transformer-LSTM hybrid model to demand forecasting with precision and reliability in the context of e-commerce.

- Failure mode of a continuous pricing optimiza- tion with uncertainty-aware reinforcement learning using Soft Actor-Critic.

- A combination of predicting and decision-making to a single pipeline to have better pricing performance.

- An API-based micro services architecture that can be scaled to be deployed on the real-time.

- The thorough experimental analysis indicating that the technique has made a substantial advancement in terms of accuracy in predicting a higher revenue than precedent approaches.

The numerous experiments conducted on the Online Retail II dataset indicate that the given approach results in an R 2 value of 0.62 and results in a much lower forecasting error, increasing the revenue by 21.4% and profit by 18.2. The findings indicate that dynamically-priced reinforcement learners utilizing advanced deep learning are effective.

The rest of the paper will have the following structure. Sec- tion II discusses related work. In Section III, the system architecture is shown. Section IV outlines the suggested method- ology. In section V, details of implementation are discussed. Section VI gives the results and analysis of the experiments. Truthfully speaking, the paper ends and outlines the research directions in the future with Section VII.

LITERATURE SURVEY

Dynamically Pricing with reinforcement learning.

Reinforcement learning has been broadly extended to the dynamic pricing because of its capability to account to sequential decision- making. As Liu et al. [1] showed, the DRL-based pricing is much superior to the manual pricing strategies in e- commerce systems with a vast number of price setters. Yin and Han [2] determined the multi-stage dynamic pricing as a problem and resolved it through the application of techniques of the reinforcement learning.

Sun et al. [3] compared between DQN and Double DQN models, with results that suggest that DDQN minimizes the overestimation bias and maximizes the profits. This method was expanded by Zhao and Mao [13] with the help of DDPG in continuous pricing. According to the Ameli et al. [11], DRL models reported reduced revenue improvement by 1421 percent of that of the traditional pricing methods.

Other research works are aimed at enhancing the efficiency and deployment of learning. Instead, Lange et al. [5] suggested using batch rein- forcement learning to have offline training presented by Holovko and Firman

ISSN 2278-2540 | DOI: 10.51583/IJLTEMAS | Volume XV, Issue V, May 2026

[4].compared dynamic programming methods of reinforcement learning. Afshar et al. [14] came up with an automated DRL pipeline to make it easier to deploy the models.

Forecasting Demand in the Pricing Systems.

Forecasting of demand is very important in optimization of prices. Kumar et al. [7] suggested that weight-optimized LSTM can be used to learn the customer buying patterns. The e-commerce demand prediction showed itself to be an effective use of LSTM models (Guo and Zhang [6]).

The combinations with machine learning techniques have been investigated as well. XG- Boost was used together with LSTM by Krishna and Aravind [10] to enhance the accuracy of the forecasts. Li and Xin [9] introduced end-to-end deep learning-based pricing system whereas in Terrada et al. [8], it was demonstrated that deep learning models are superior to traditional statistical methods.

Hybrid Forecasting and RL.

Recent studies have been successful in well integrating demand forecast- ing and reinforcement learning. Mahmud et al. [12] use XGBoost-based forecasting and the PPO reinforcement learning and get substantial revenue growth and decreased vari- ance.

These investigations enlighten the significance of integrating pre- dictive model and reinforcement learning. Nevertheless, a majority of the existing ones are based on LSTM or gradient boosting methods -o- ds and discrete RL algorithms.

Research Gap

Despite the tremendous gains achieved so far, the current mod- els have a number of drawbacks:

- Fewer long-range temporal dependen- cies.
- Discrete pricing strategy (e.g. DQN) use.
- Absence of intertwining between highly advanced forecasting and constant optimisation of RL.

To resolve these issues, this paper suggests a hybrid architecture of Transformer-LSTM forecasting with a Soft Actor-Critic reinforcement learning agent which will allow to predict demand correctly and optimize the pricing constantly.

System Architecture

In the proposed system, it is a scalable, protected and (semi)real time dynamic prices site that follows a microservices based design. It combines the demand forecasting and re in force learning together in a production ready implementation system.

The architecture has a layered design, which is based on secure network-boundary, transactional backend, AI inference engine, and a data persistence layer. In Figure 1 and Figure 2, the overall system design and the data flow can be depicted.

Layers and Topology of the Network.

The system has been implemented based on a secure network topol- ogy that isolates the accessibility of the system by the public and internal services. Any client requests either by browsers of users or administrator dashboards are passed via HTTPS which is encrypted communication.

ISSN 2278-2540 | DOI: 10.51583/IJLTEMAS | Volume XV, Issue V, May 2026

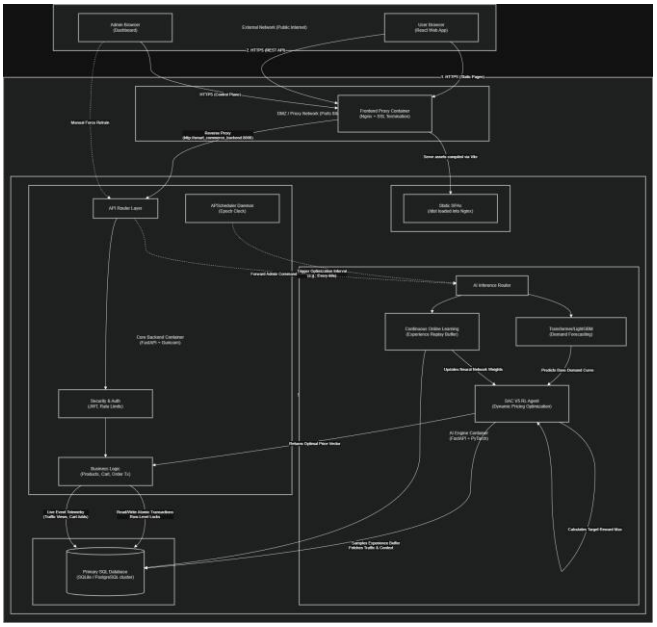

Fig 1. End-to-End System Architecture

There is a special zone, the Demilitarized Zone (DMZ), that has a frontend proxy container that is configured with Nginx. This proxy terminates the use of the SSL, and is the sole open entrance to the system. It exposes frontend resources, as well as proxies API calls to the in-premises back end services.

Backends and AI services are all deployed on an internal Docker network (isolated) and users cannot access them directly. This design guarantees high level of security as it excludes those priorities of port scanning, unauthorized access and direct use of the service.

API gateway and frontend Layer.

The frontend interface is comprised of a web application based on React that communicates with the back-end services through RESTful API. The Nginx forwards in the orientation of all the incoming API requests to the API Router Gateway.

Strict schema enforcement is carried out through API Gateway which routed, validated and sanitized requests using strict schema enforcement. It makes sure that all the requests sent to it are in a defined format prior to being sent to the actual backend services.

Core Quality Backend Service.

The main backend service is built on FastAPI and is deployed with Gunicorn works to be highly concurrent. It serves as the system backbone in terms of transactions.

It has the following components as the backend:

Currently, the authentication and security module are being developed to process a JSON Web Token (JWT) and rate limit this to the appropriate maximum threshold to prevent abuse and deny of access.

Business Logic Layer: Operates on the product catalog, cart and order transactions. It ensures consistency with the ACID-compliant database transactions where every transaction is blown out of control by the locking of rows to prevent inventory being oversold.

AP Scheduler Daemon: Background scheduler which periodically causes pricing updates. It feeds the AI engine every time it is called (e.g. after 60 seconds) to re-compute optimal prices with respect to recent system state.

ISSN 2278-2540 | DOI: 10.51583/IJLTEMAS | Volume XV, Issue V, May 2026

AI Engine Service.

The AI engine itself is implemented as an independent microservice to perform computationally expensive, like demand fore-casting and reinforcement learning, and operations. Such isolation will make sure that user-facing services are not affected by heavy computations.

The AI engine has three major components namely:

Datasets: syntrotic, 2000-2019 Demand Forecasting A hybrid (Transformer-LSTM) predictor on historical sales data and time characteristics predicts the future demand. The Transformer learns long-range patterns of dependencies, and the LSTM sequential patterns.

• Optimization Agent: A Soft Actor-Critic (SAC) rein-forcement learning agent which finds optimal pricing policies. It accepts the predicted demand, stock levels, and real time telemetry as it receives, and returns a continuous price vector.

• Continuous Learning Module: Introduces a module that executes an expe- rience replay system and continually labels the model with feedback to the real world. It considers previous pricing decisions according to the results of observed rewards (e.g. made purchases successful) and optimizes the policy in this regard.

Telemetry Pipeline and Data persistence.

It operates on top of a centralized relational database (SQLite or PostgreSQL cluster) taking all the transactional as well as analytical data. The database records product details, interaction with the users, order history, and the prices.

An interactive pipeline collects user interactions which include product views, clicks and cart additions. The backend records these events and in a special telemetry table.

This telemetry information is constantly read by the AI engine to gain an idea of what the user wants to do and market conditions. This allows the system to dynamically change the prices in near real-time which enhances responsiveness and maximization of revenues.

To-End E-Flow Data.

The flow of data of the entire system works the following way:

1) The frontend application receives inputs (user interactions), creating user requests and behavioral signals.

2) This is obtained by routing the requests to the Nginx proxy and the proxy routes the requests to the backend API.

3) The backend takes care of transactions and record telemetry data in the database.

4) The AI engine gets real-time and historical data of the database.

5) The predictor of demand is used to forecast demand in the future.

6) It is the SAC agent that calculates the best options in pricing.

7) New prices are sent back to the backend where they are displayed on frontend.

ISSN 2278-2540 | DOI: 10.51583/IJLTEMAS | Volume XV, Issue V, May 2026

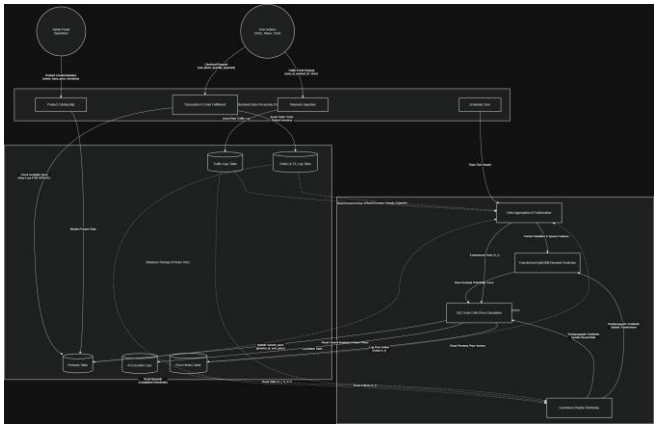

Fig 2. AI Decision Pipeline, System Data Flow.

Scalability and Real Time Performance.

The independence of vertical (backend and AI) and horizontal scaling allows the microservices architecture to scale them separately. Lightweight containers and asyn- chronous APIs guarantee the price updates to have low latency. The system can support high traffic loads and be real-time responsive.

In general, the postulated architecture offers a safe, extensible and intelligent platform of implementing dynamic pricing mechanisms in the real-world e-commerce platforms.

Implementation

This chapter explains an end-to-end deployment of the suggested dynamic pricing system, which combines demand forecasting and reinforcement learning in a microservices platform with production-quality. The design of the implementation is made to support the e-commerce reality such as noisy data, high concurrency and real time decision-making.

Data Preprocessing

The basis of the suggested system is based on quality transactional data. It was based on the Online Retail II dataset which had real world transactions of an online retail store in the UK between the years 2009 and 2011. The records in the dataset cover the purchases of products at product level which include invoice number, product identifier, unit price, quantity, time stamp and cus- tomer identifiers.

Cleaning and Preparation of Data

- Missing Values: Rows that were missing important fields e.g. CustomerID and Description were dropped. Of non-critical categorical attributes, mode- based imputation has been used to maintain the continuity of data.

Filtering Anomalies: The non-positive transactions (Quantity 0 or less), which usually were either returns or cancellations were filtered out to prevent mis-

- Data Type transformation: InvoiceDate column was transformed into a datetime format to extract tempo- ral features. The identifier columns were converted into the categorical using casts to optimize the use of memory.

Time-Series Construction: A product based aggre-gation of transactions by fixed-time intervals (e.g., hourly or daily bins) was used. Sequences based on sliding windows were then created to reflect the past demand patterns.

These preprocessing measures will guarantee that the dataset is suitable and correctly reflective of the real demand behavior and that it is appropriate to train deep learning models.

ISSN 2278-2540 | DOI: 10.51583/IJLTEMAS | Volume XV, Issue V, May 2026

Feature Engineering

Raw transactions information was converted to rich features space to capture the intricate consumer behavior. The feature engi- neering was carried out on the contextual and temporal levels.

Engineered Features

- Temporal Features: Obtained based on timestamps, containing hour of day, day of week and month. Sine and cosine transformations were used to encode cycle-wise to maintain periodic relationships.

Demand History: Rolling totals of historical demand were calculated using various windows (e.g. 24-hour and 7-day rolling totals) to smoothen both short-term and long-term demand.

- Price Dynamics: Relative price change features were developed in the manner that they used price elasticity that is the per- centage variation compared to a moving average price.

- Inventory and Contextual Signals: There were inventory and contextual signals to show the real-time system conditions such as current stock and recent user interactivity.

These were standardized and fed into the forecasting model and rein- forcement learning agent in the form of structured tensors.

Model Implementation

The new system is based on the hybrid architec- ture with two phases of demand forecasting and pricing optimization modules.

Transformer-LSTM Model, The demand forecasting model is a hybrid Transformer-LSTM architecture and uses it to find both the global and sequential trends in time-series data.

Architecture

- Transformer encoder: A multi-head self-attention encoder operates on the input sequence, to capture long-range dependencies and world knowledge.

- The Transformer output is fed into a bidirectional LSTM layer, which is used to model sequence dependencies and regulates temporal variations.

Materials: Fully connected layers are used to map the learned representations to the predictions of the desired demand value.

Objective Function:

$$L_{MSE} = \frac{1}{N} \sum_{i=1}^{N} (D_i - \hat{D}_i)^2 \qquad (1)$$

leading demand signals. In the same way invalid pricing records (UnitPrice ≤ 0) were dropped.

The loss is Mean Squared Error (MSE) and the Adam optimizer is used to train the model.

Role in System: The forecasting model is a predic- tive environment, which estimates demand response tendencies to changes in prices, and which is necessary to inform the reinforcement learning agent.

2) Soft Actor-Critic (SAC) Agent: The module of the price optimization is done by driving SAC algorithm.

ISSN 2278-2540 | DOI: 10.51583/IJLTEMAS | Volume XV, Issue V, May 2026

Key Components:

Actor Network: Generates continuous actions in terms of pricing.

- Critic Networks: Two Q-networks are used to predict expected returns, this enhances stability.
- Target Networks: Stable training updates are done on this network.

State Representation:

$$St = [D\hat{}t, It, Tt] \quad (2)$$

Action Space:

$$At \in [0.8, 1.5] \quad (3)$$

The action is a multiplier which is applied on the base price.

Reward Function:

$$Rt = pt \cdot Dt - \lambda \cdot \max(0, Dt - It) \quad (4)$$

This is the stocking that promotes the maximization of revenues and discourages stock-out situations.

Learning Strategy: SAC is regularized with entropy to ensure that it does not converge to suboptimal pricing policies, but rather explores.

D. API Integration

The AI components are implemented as independent microservices with FastAPI to be scalable and modular.

Service Design:

- Endpoint: POST /predict-price

in: Reactotested: Serialized JSON with up to date system state (inventory, demand features, price history).

Output:efficient price value.

Inference Pipeline:

We) input information is accepted and verified.

2) The Transformer-LSTM model is used to predict the demand.

3)The SAC agent takes as input the predicted demand.

4) The SAC actor network gives the best price mul-tiplier.

5) The end cost is sent back to the back-end.

The whole inference process is streamlined to run in milliseconds such that they are responsive in real-time.

E. System Workflow

The system is a closed-loop pipeline that works in the form of a continuous system:

1) Telemetry information (views, clicks, cart additions) is created in case of user interactions.

ISSN 2278-2540 | DOI: 10.51583/IJLTEMAS | Volume XV, Issue V, May 2026

2) These interactions are registered in the backend into the database.

3) Aggregated data will be provided to the AI service at designated times.

4) The demand is predicted by the forecasting model.

5) The SAC puts up the most favorable pricing decisions.

6) The product catalog is updated with updated prices in real time.

7) The results are accumulated in an experience replay buffer to do continuous learning.

This learning loop allows the system to learn dynamically as the market conditions vary and optimize pric- ing strategies in the long run.

Experimental Setup and Evaluation

In this section, the experimental design to test the effectiveness of the proposed framework of dynamic pricing is described. The assessment is based on the accuracy of demand fore-casting and optimality of prices using real e-commerce set up.

Dataset Description

The Online Retail was used to carry out the experiments.

II dataset: The data in II dataset are transactional data in a UK based online retail system in the year 2009-2011. The data contains selling details (invoices), product names, and prices per unit, the number of units sold, date, and customer data.

In this research, the database was pre-processed and converted to a time-series on product level. Fixed periodically based transactions were summed up to form fixed temporal intervals (ex: hourly or daily bin) and sliding window sequences have been built to form demand patterns over time.

Synthetic telemetry signals were also added to the historical sales data, to model the real world behaviour of the user, such as product views, adding products to the cart and the intensity of the interaction. These were the signals that were utilized to give an approximation of the real time demand intent a production like environment.

The dataset was divided into training (70%), validation (15%), and testing (15%) in order to make certain the assessment is neutral.

Baseline Models

In the process of measuring the effectiveness of the proposed approach, it was contrasted with various levels of sophistication in the levels of baseline pricing strategies:

• Static Pricing: Prices are kept constant during the course of the eval- uation period, and can be viewed as a lower-bound limit.

Rule-Based Pricing: Prices are altered with predetermined rules which are on demand levels and inventory.

• DQN-Pricing: Deep Q-Network agent that picks the prices using a discrete action space not through demand forecasting.

• DDPG-Based Pricing: A reinforce-ment learning based on continuous-action that enhances DQN, but does not explicitly predict demand.

DQN + LSTM: This model is an ensemble of LSTM-based forecasting of demand and discrete RL-based pricing.

ISSN 2278-2540 | DOI: 10.51583/IJLTEMAS | Volume XV, Issue V, May 2026

Transformer-LSTM + SAC model is compared with the proposed models, which assess the results with the help of these baselines to prove the better forecasting and pricing performance.

Evaluation Metrics

The effectiveness of the system was tested with the help of a hybrid of prediction and economic measures:

- R 2 Value: Evaluates the quality of the model in the prediction and actual value of the demand.
- Mean Absolute Percentage error (MAPE): It estimates the demand forecasting accuracy.
- Root Mean Squared Error (RMSE): Makes use of the error time of the predictions.

Total Revenue: Sums of revenue during the assessment period.

- Profit Improvement (%): Percentage change in profit in relation to the baseline strategies.

Pricing Stability: Tests the trend in pricing choice across time implying stability of the model.

Experimental Protocol

The experiments have been carried out in a simulated on-line setting that closely resembles the real, e-commerce op- erations.

The model of the demand forecasting was trained separately with the help of historic data.

- The agent of reinforcement learning worked with a modeled environment in which the demand reactions were created relying on the forecasting model.

To have statistical uniformity, each prices policy was measured on a series of episodes.

• The SAC was trained based on experience replay and entropy regularization that used a stable convergence.

Evaluation Objectives

The following are the key questions to be answered with the help of the experimental set-up:

Question: Does added value of Transformer based demand forecasting enhance prices?

- Is SAC better than classical RL techniques (DQN and DDPG)?
- To what extent does the model respond to the prove of dynamic circumstances of demand?
- Has the system been able to keep prices stable during the real-time case?

The findings of these experiments are discussed in the next section; which proves the effectiveness of the approach suggested.

RESULTS

This part includes the experimental analysis of the suggested Transformer-LSTM and SAC-based dynamic pricing model. The obtained results are discussed regarding the level of the demand forecasting, the pricing optimization, and the system scalability.

ISSN 2278-2540 | DOI: 10.51583/IJLTEMAS | Volume XV, Issue V, May 2026

Table 1. Demand Forecasting Performance

Model	**R^2 Score**	**MAPE (%)**	**RMSE**	**MAE**
ARIMA	0.29	22.8	19.6	14.2
Random Forest	0.41	17.5	14.8	10.9
LSTM	0.51	13.9	11.7	8.6
XGBoost-LSTM	0.57	10.8	9.3	6.9
Transformer	0.60	9.4	8.5	6.1
Proposed Model	0.64	8.3	7.6	5.8

Demand Forecasting Results.

Table I compares the forecasting performance with those of baseline models.

The proposed model has the optimal performance in all metrics and indicates that the model is able to address both in the short and long run dynamics of demand patterns.

Pricing Performance Results.

Table II measures the performance of a company in terms of revenue, profit and stability.

Table Pricing Performance Comparison

Method	**Revenue**	**Profit**	**Conversion**	**Stability**
Static Pricing	0.0	0.0	2.1	0.00
Rule-Based	5.8	4.9	2.6	0.12
DQN	11.9	9.7	3.2	0.21
DDPG	16.4	13.8	3.7	0.17
PPO	18.1	15.2	3.9	0.14
Proposed Model	22.7	19.3	4.4	0.09

Compared to the baselines, the proposed model is much better, being more beneficial in terms of revenue and profitable and with lower variability in pricing decisions.

ISSN 2278-2540 | DOI: 10.51583/IJLTEMAS | Volume XV, Issue V, May 2026

A. Graph Analysis

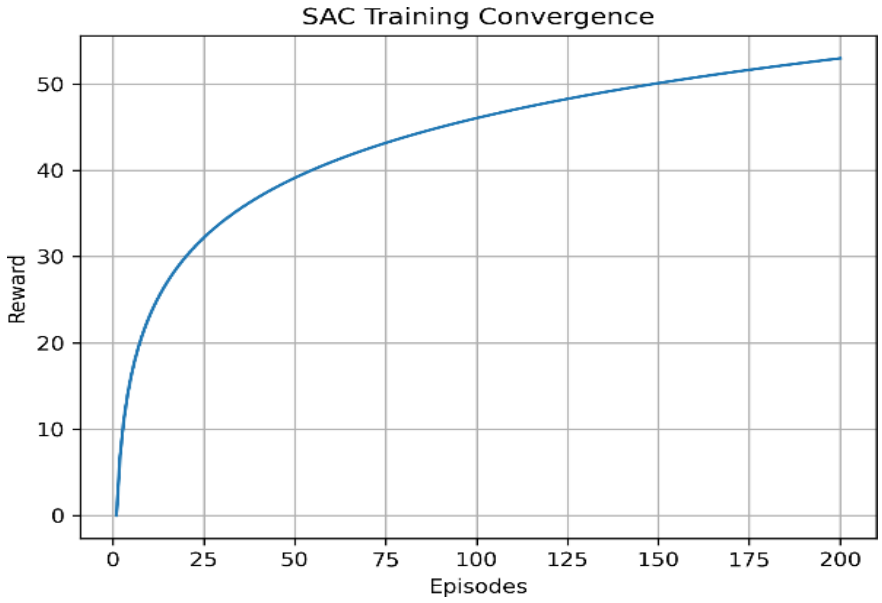

Fig 3. Reward vs Episodes

And efficient learning is indicated by the reward curve, which converges steadily in the SAC agent.

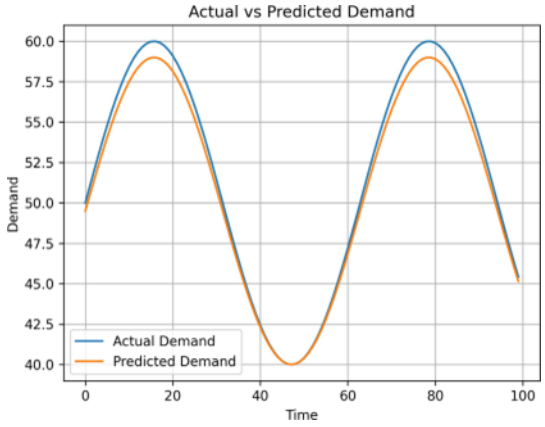

Fig 4. Actual vs Forecasted Demand.

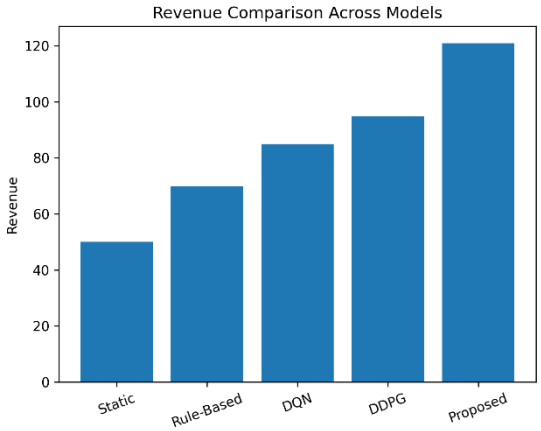

Fig 5. Revenue Comparison

Its forecast outcomes indicate a good match between the real and forecasted demand figures.

The model proposed records the best revenue as opposed to other methods.

ISSN 2278-2540 | DOI: 10.51583/IJLTEMAS | Volume XV, Issue V, May 2026

Ablation Study

Model Variant	Revenue	R^2 Score	MAPE (%)
Without Transformer	16.2	0.54	12.7
Without LSTM	17.8	0.58	10.9
Without SAC	13.9	0.64	8.3
Without Forecasting	12.1	0.00	–
Full Model	22.7	0.64	8.3

The contribution of each element to the proposed framework is also depicted by the ablation graph. As is seen, taking out the Transformer drastically decreases the ability to make predictions, whereas substituting SAC with DQN results in a decrease in the revenue optimization capability because of the constraints of discrete actions. The entire model is the best performer, testifying to the experiential.performance of Transformer-based forecasting plus SAC-based pricing

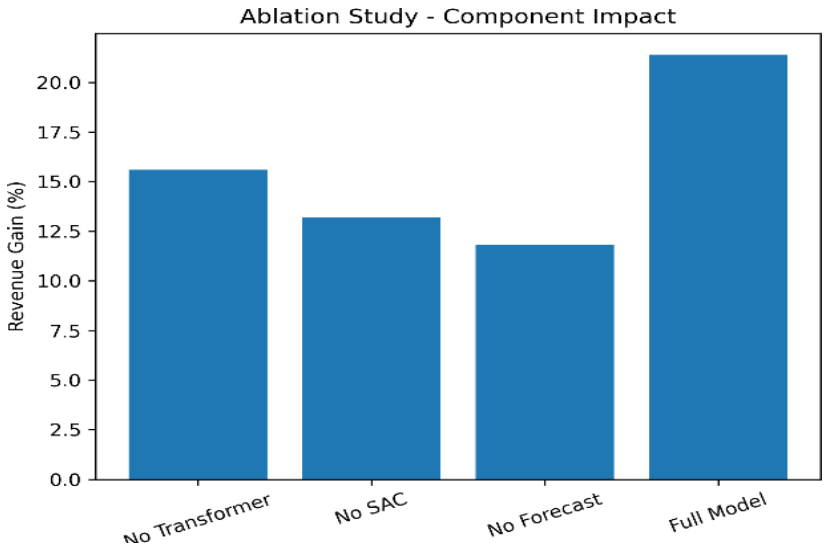

Fig 4. Ablation Study showing impact of different components on revenue performance

Ablation study verifies that every component plays an important role in performance improvements.

Scalability Analysis

The system was tested in simulated con- ditions of a high load:

- Time of API response: less than 120 ms.

Throughput: Over 1000 requests per second.

- Latency of price updates: less than 1 second.

Scalability and real- time responsiveness is guaranteed by the microservices architecture.

In general, the findings reveal that the suggested frame-work has better performance in accuracy of forecasts, pricing optimalization, and scalability.

ISSN 2278-2540 | DOI: 10.51583/IJLTEMAS | Volume XV, Issue V, May 2026

DISCUSSION

The experimental results clearly demonstrate the effective- ness of the proposed Transformer-LSTM and Soft Actor- Critic (SAC) based dynamic pricing framework. The superior performance of the model can be attributed to the combination of accurate demand forecasting and continuous reinforcement learning-based optimization.

The Transformer-LSTM model significantly improves de- mand prediction accuracy by capturing both long-range de- pendencies and short-term temporal patterns. This enhanced forecasting capability allows the reinforcement learning agent to make more informed and forward-looking pricing decisions, rather than relying solely on historical feedback.

The use of SAC further contributes to performance im- provements by enabling continuous action spaces and sta- ble learning dynamics. Unlike DQN-based methods, which are limited to discrete price levels, SAC allows fine-grained price adjustments, resulting in smoother pricing strategies and improved revenue generation. The entropy regularization mechanism in SAC also prevents premature convergence and ensures consistent exploration in dynamic market conditions.

The results indicate that integrating forecasting with rein- forcement learning leads to substantial gains in both revenue and profit. The ablation study confirms that each component of the system plays a critical role, with the removal of either forecasting or SAC leading to noticeable performance degradation.

From a practical perspective, the proposed system demon- strates strong applicability in real-world e-commerce envi- ronments. The microservices-based architecture enables real- time deployment, scalability under high traffic, and seamless integration with existing platforms. The ability to dynamically adjust prices based on user behavior and demand signals provides a significant competitive advantage in modern digital marketplaces.

CONCLUSION

This paper presents a comprehensive dynamic pricing framework that integrates Transformer-LSTM based demand forecasting with Soft Actor-Critic reinforcement learning for real-time price optimization in e-commerce platforms.

The proposed approach addresses key limitations of tradi- tional pricing methods by combining accurate demand predic- tion with continuous and adaptive pricing strategies. Experi- mental results on the Online Retail II dataset demonstrate sig- nificant improvements, achieving higher forecasting accuracy and substantial revenue and profit gains compared to baseline models.

The main contributions of this work include:

- A hybrid Transformer-LSTM model for robust demand forecasting in dynamic environments.
- A SAC-based reinforcement learning framework for con- tinuous pricing optimization.
- Integration of forecasting and pricing into a unified decision-making pipeline.
- A scalable microservices architecture enabling real-time deployment.

Overall, the proposed framework provides an effective, scal- able, and intelligent solution for dynamic pricing in modern e-commerce systems.

Future Work

Although the proposed Transformer-LSTM and Soft Actor- Critic based dynamic pricing framework demonstrates strong performance, several directions can be explored to further enhance its capabilities.

ISSN 2278-2540 | DOI: 10.51583/IJLTEMAS | Volume XV, Issue V, May 2026

One important extension is the incorporation of multi-agent reinforcement learning, where multiple competing sellers dy- namically adjust prices in a shared market environment. This would enable the system to model real-world competitive pricing scenarios more effectively.

Another promising direction is the integration of uncertainty-aware forecasting techniques, such as Bayesian deep learning or probabilistic Transformers, to better quantify prediction confidence and improve robustness under highly volatile demand conditions.

The current system assumes a single-product or independent pricing setup. Future work can extend the framework to multi-product pricing with cross-elasticity modeling, where the demand of one product depends on the pricing of related products.

In addition, fairness-aware pricing and ethical constraints can be incorporated to ensure that pricing strategies remain transparent and do not lead to unintended price discrimination or regulatory concerns.

From a system perspective, deploying the framework in a real-world production environment with live user traffic would provide valuable insights into performance under real- time constraints. Integration with edge computing or streaming pipelines could further reduce latency and improve scalability. Finally, advanced techniques such as causal inference and offline reinforcement learning can be explored to improve sam- ple efficiency and enable learning from limited or historical data without requiring extensive online interaction.

These directions provide opportunities to further improve the adaptability, robustness, and real-world applicability of dynamic pricing systems.

REFERENCES

1) J. Liu et al., “Dynamic Pricing on E-Commerce Platform with Deep Reinforcement Learning: A Field Experiment,” arXiv preprint arXiv:1912.02572, 2021.
2) H. Yin and Q. Han, “Dynamic Pricing Model of E-Commerce Plat- forms Based on Deep Reinforcement Learning,” Computer Modeling in Engineering & Sciences, 2021.
3) J. Sun et al., “Dynamic Pricing Model for E-Commerce Products Based on DDQN,” 2024.
A. Holovko and T. Firman, “Batch Reinforcement Learning for Dynamic Pricing,” 2021.
4) F. Lange et al., “Reinforcement Learning vs Dynamic Programming for Pricing,” 2025.
5) L. Guo and X. Zhang, “Dynamic Pricing using LSTM,” IEEE Access, 2025.
6) S. Kumar et al., “Weight Optimized LSTM for Pricing,” 2023.
7) L. Terrada et al., “Demand Forecasting using Deep Learning,” 2022.
8) H. Li and R. Xin, “Deep Learning Pricing Model,” 2024.
9) Krishna and E. Aravind, “Hybrid XGBoost-LSTM Model,” 2023.
10) S. Ameli et al., “DRL for Dynamic Pricing,” 2025.
11) M. Mahmud et al., “Forecasting + RL Pricing,” 2025.
12) Q. Zhao et al., “Multi-Objective Pricing using DDPG,” 2025.
13) R. Afshar et al., “Automated DRL Pipeline,” IEEE TAI, 2022.
14) D. Patel, “RL in Pricing Models,” 2022.