ISSN No:-2456-2165 https://doi.org/10.38124/ijisrt/26may1751

BargainBot: AI-Driven Price Negotiation Chatbot for E-Commerce

Sagar D. Patil[1]; Dr. Manisha Bharati[2]

[1]Department of Technology, Savitribai Phule Pune University, Pune, India
[2]Department of Technology, Savitribai Phule Pune University, Pune, India

Publication Date: 2026/06/02

Abstract: Fixed pricing in e-commerce platforms creates a significant gap compared to the interactive bargaining experience customers enjoy in physical retail environments. This paper presents BargainBot, a deployed full-stack AI chatbot that enables real-time, free-form price negotiation on e-commerce platforms. The proposed system integrates three core components:

- **A PyTorch Recurrent Neural Network (RNN) trained on a custom multilingual intent dataset spanning English, Hindi, and Marathi, achieving approximately 91% classification accuracy across 15 negotiation intent classes.**
- **A rule-based negotiation engine that enforces per-product seller margin constraints while generating contextually appropriate counter-offers.**
- **A sentiment-aware response layer that detects buyer tone and adjusts negotiation strategy accordingly.**

Unlike prior work that relies on fixed command keywords or scripted menus, BargainBot accepts fully free-form natural-language input, disambiguates discount-amount offers from target-price offers, and maintains offer state across multiple conversation turns. The system is implemented as a decoupled React.js frontend communicating with a Django REST backend, supporting 37 products across six categories. Experimental evaluation demonstrates a negotiation success rate of 84%, intent classification accuracy of ~91%, and 100% seller margin protection across all tested scenarios.

Keywords: *Price Negotiation, E-Commerce Chatbot, Recurrent Neural Network, Intent Classification, Natural Language Processing, Multi-Turn Dialogue, Rule-Based Reasoning, Sentiment Analysis, Dynamic Pricing.*

How to Cite: Sagar D. Patil; Dr. Manisha Bharati (2026) BargainBot: AI-Driven Price Negotiation Chatbot for E-Commerce. *International Journal of Innovative Science and Research Technology*, 11(5), 2743-2753.
https://doi.org/10.38124/ijisrt/26may1751

I. INTRODUCTION

The global e-commerce industry has grown exponentially over the past decade, with customers increasingly preferring online channels for their purchasing needs. Yet despite advances in recommendation engines, personalization, and logistics, one fundamental aspect of traditional commerce remains largely absent from digital platforms: price negotiation. In physical markets, bargaining is a natural social interaction that allows buyers to express price sensitivity and sellers to optimize revenue within acceptable margins. This interactive process has been shown to increase buyer satisfaction and reduce cart abandonment [6].

Most commercial e-commerce platforms enforce fixed pricing, compelling customers to either accept the listed price or abandon their purchase. This rigidity has measurable business consequences: studies indicate that price dissatisfaction is among the top three reasons for online cart abandonment [9]. While discount coupons and flash sales offer partial relief, they do not create the personalized, interactive experience that negotiation provides.

Automated negotiation agents have attracted research attention for over two decades [12], but practical deployments in consumer e-commerce remain rare. Early attempts relied on genetic algorithms or simple scripted rule trees, while more recent work has explored reinforcement learning [6] and ensemble machine learning [5]. However, these systems share a critical limitation: they require users to interact through predefined commands or keyword triggers, rather than genuine natural language conversation.

This paper presents BargainBot, a full-stack negotiation chatbot that addresses these shortcomings through the following contributions:

- A PyTorch RNN-based intent classifier trained on a custom multilingual dataset covering 15 negotiation intent categories in English, Hindi, and Marathi.

ISSN No:-2456-2165 https://doi.org/10.38124/ijisrt/26may1751

- A multi-turn negotiation engine that maintains offer state across conversation turns and enforces per-product seller margin constraints.
- Free-form price extraction and disambiguation — the system correctly interprets 'Offer 500 discount' (discount from MRP) differently from 'I will pay 500' (absolute target price).
- A decoupled React.js + Django REST architecture that demonstrates production-grade deployment, in contrast to the Flask prototypes prevalent in existing literature.
- Experimental results showing 84% negotiation success rate, 91% intent classification accuracy, and consistent seller margin protection across all tested scenarios.

The remainder of this paper is organized as follows. Section II reviews related work. Section III defines the research problem. Section IV describes the proposed methodology. Section V presents the system architecture. Section VI covers implementation details. Sections VII–IX describe the AI components. Section X presents experimental results. Sections XII and XIV discuss advantages, limitations, and future scope, followed by a conclusion in Section XV.

II. RELATED WORK

Research in automated price negotiation for e-commerce spans rule-based, machine learning, and hybrid approaches. This section reviews representative work and identifies the gaps that BargainBot addresses.

➢ *Rule-Based and Scripted Negotiation*

Bhamre et al. [1] proposed 'Bargain Buddy,' an e-commerce negotiation bot that applies a three-phase fixed discount strategy: 50%, 75%, and 100% of a predicted discount computed by a decision tree regressor. The system uses customer ID, product ID, and quantity as features. While simple and fast, the interaction model is entirely scripted, with no natural language understanding.

Similarly, Bindu et al. [2] proposed an NLP-based negotiation chatbot using TensorFlow. Despite the NLP label, the system accepts only predefined keyword inputs and does not maintain conversational context across turns. The negotiation logic is not clearly separated from the dialogue management layer.

➢ *Machine Learning Price Prediction*

Liu and Zheng [11] developed a negotiation assistant that uses machine learning to predict acceptable price ranges. The system employs a decision tree model but focuses on customer-side benefits without modeling seller margin constraints. Usha Sri et al. [4] applied Stochastic Gradient Descent for NLP intent processing in a chatbot context, demonstrating improved scalability but without explicit multi-turn dialogue state management.

Metre et al. [8] proposed a Random Forest and OpenAI API integration for price negotiation, incorporating user profiling (new, returning, frequent buyer) and sentiment analysis on product reviews. This represents the most feature-rich approach in recent literature, but requires external API access and does not demonstrate end-to-end deployment with an actual negotiation dialogue.

➢ *Ensemble and Voice-Based Systems*

Challagundla et al. [5] combined SVM and KNN in an ensemble to predict initial negotiation prices, adding voice interaction through speech-to-text APIs. The negotiation protocol accepts only two commands ('first price' and 'final price'), making it brittle for real-world use. Surekha et al. [3] and Pappala et al. [7] present similar text-and-voice systems with comparable limitations.

➢ *NLP and Reinforcement Learning Approaches*

Rana et al. [6] proposed a hybrid NLP and Reinforcement Learning chatbot, using BERT for intent classification and a simulated RL agent for counter-offer generation. While theoretically well-motivated, the paper presents BERT architecture figures without reporting concrete negotiation accuracy metrics or a deployed system.

Prasad et al. [9] reviewed AI-driven negotiation systems and highlighted the lack of production-ready deployments as a key gap. Their review confirms that most existing systems lack: free-form NLP, multi-turn state management, and seller-margin enforcement.

➢ *Comparison Summary*

Table 1 summarizes key differences between prior work and the proposed BargainBot system.

Table 1 Comparison of Related Negotiation Chatbot Systems

System	NLP	ML	Multi-Turn	Free-Form	Margin Guard	Full Stack
Bhamre et al. [1]	✗	✓	✗	✗	✗	✗
Bindu et al. [2]	Partial	✗	✗	✗	✗	✗
Liu & Zheng [11]	✗	✓	✗	✗	✗	✗
Challagundla et al. [5]	✗	✓	✗	✗	Partial	✗
Rana et al. [6]	BERT	✓	✗	✓	✗	✗
Metre et al. [8]	✓	✓	✗	Partial	✗	✗
BargainBot (Ours)	**RNN**	**✓**	**✓**	**✓**	**✓**	**✓**

ISSN No:-2456-2165 https://doi.org/10.38124/ijisrt/26may1751

➢ *Problem Statement*

The core research problem addressed in this work can be stated as follows:

Given a user's free-form natural language message in a multi-turn negotiation dialogue over a specific product, and given the product's listed price PL and minimum profitable price Pmin, determine an appropriate counter-offer Pc such that:

$$\text{Pmin} \leq \text{Pc} \leq \text{PL} \qquad (1)$$

While maximizing buyer satisfaction and closing the negotiation within a bounded number of dialogue turns.

Existing systems fail to solve this problem adequately for three reasons. First, they require users to follow a fixed interaction protocol (e.g., type 'first price' or click a button labeled 'negotiate'), which does not reflect natural human communication. Second, they do not maintain a negotiation state across turns, so each message is processed independently without knowledge of previous offers. Third, most systems either ignore seller margin constraints or apply only a fixed global discount ceiling.

BargainBot addresses all three limitations by combining trained NLP-based intent classification, regex-based price extraction, and a stateful negotiation engine that enforces constraint (1) at every dialogue turn.

III. PROPOSED METHODOLOGY

The BargainBot system follows a four-stage pipeline for each negotiation turn. The pipeline is illustrated in Figure 1 below.

Figure 1: BargainBot Processing Pipeline

Fig 1 BargainBot Processing Pipeline for a Single Negotiation Turn.

➢ *Stage 1 — Text Preprocessing*

The incoming user message is converted to lowercase, tokenized using NLTK's word_tokenize, and each token is stemmed using the Porter Stemmer. This reduces vocabulary size and improves generalization (e.g., 'negotiating', 'negotiates', and 'negotiated' all map to the same stem 'negoti').

➢ *Stage 2 — Bag-of-Words Representation*

A binary bag-of-words (BOW) vector is computed over the full training vocabulary V. The BOW input vector x ∈ {0,1}^|V| encodes the presence of each vocabulary word in the stemmed message.

➢ *Stage 3 — Intent Classification*

The BOW vector is passed through the RNN classifier to produce a probability distribution over 15 intent classes. The predicted intent is the class with maximum posterior probability.

➢ *Stage 4 — Negotiation Engine*

The predicted intent, extracted price (if any), sentiment, and the product's PL and Pmin are passed to the negotiation engine, which applies intent-specific logic to generate a counter-offer satisfying constraint (1) and a natural language reply.

➢ *System Architecture*

Figure 2 presents the overall system architecture of BargainBot. The system follows a decoupled client-server design.

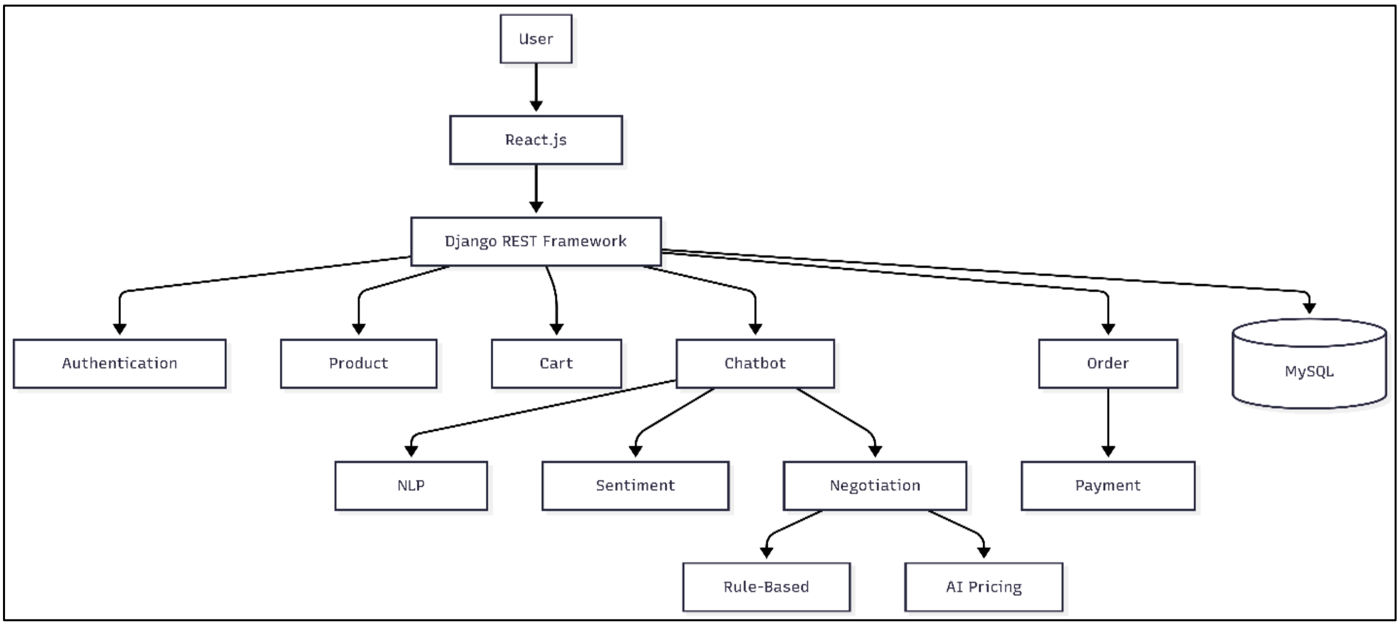

Fig 2 System Architecture of BargainBot

➢ *Frontend Layer*

The frontend is a React.js Single-Page Application (SPA) with three primary routes: / (product catalogue), /cart (cart management), and /chat/:id (negotiation interface). The negotiation page renders two panels: the Chatbot component (managing conversation state) and the AIInsights component (displaying live sentiment, confidence, strategy, and savings metrics).

➢ *Backend Layer*

The Django backend exposes four REST endpoints. The primary endpoint, POST /api/negotiate/, accepts a JSON payload containing the user message, listed price, minimum price, and current offer price. It invokes the negotiation engine and returns a comprehensive JSON response including the counter-offer, deal status, sentiment, negotiation strategy, intent label, and confidence score.

➢ *Data Layer*

Product data is stored in an SQLite database table (frontapp_product) and a static JavaScript array in Product.js. The CSV file export2.csv serves as the authoritative source for both. The min_profitable_price field is the critical parameter that enforces seller margin protection throughout all negotiation turns.

Table 2 Technology Stack of BargainBot

Component	Technology
Frontend Framework	React.js 18 (CRA, Hooks, React Router v6)
HTTP Communication	Fetch API (native browser)
Backend Framework	Django 4.2+ (REST, CORS enabled)
NLP / Intent Model	PyTorch (RNN), NLTK (tokenizer, stemmer)
Discount Prediction	scikit-learn LinearRegression
Database	SQLite3 (Django ORM)
Data Processing	NumPy, Pandas
Product Catalogue	export2.csv (37 products, 6 categories)
Styling	Custom CSS (glassmorphism, dark theme)
Dev Environment	Python 3.13, Node.js 18, Windows 10

➢ *Implementation Details*

- *Technology Stack*

Table 2 lists the complete technology stack of BargainBot.

- *Project Structure*

The backend is organized into four Django applications: frontapp (product and API views), cart (cart and payment management), chat (RNN model, negotiation engine, and chat session storage), and login (user authentication). The frontend src/ directory is organized into pages/, components/, and services/, following standard React application conventions.

ISSN No:-2456-2165 https://doi.org/10.38124/ijisrt/26may1751

Table 3 BargainBot Intent Classes and Representative Patterns

Intent Tag	Sample Pattern	Languages
greeting	"Hi", "Hello", "Hey there"	EN, HI, MR
ask_price	"What is the price", "How much"	EN, HI, MR
ask_discount	"Give me a discount", "Any offer"	EN, HI, MR
make_offer	"My budget is", "I can offer"	EN, HI, MR
counter_offer	"Can you do better", "Still too high"	EN, HI
accept_deal	"Deal", "I agree", "Let's do it"	EN, HI, MR
reject_deal	"Not interested", "Too expensive"	EN, HI
ask_lowest	"Best price", "Lowest possible"	EN, HI
too_low	"That's too low for me"	EN
urgency	"Final answer", "Last offer"	EN, HI
payment_query	"How to pay", "Payment options"	EN
quality_assur	"Is it genuine", "Warranty"	EN
bundle_offer	"I want multiple", "Bulk order"	EN
thank_you	"Thanks", "Thank you"	EN, HI, MR
goodbye	"Bye", "See you later"	EN, HI

IV. NLP-BASED INTENT CLASSIFICATION

➢ *Model Architecture*

The intent classifier is implemented as a single-layer PyTorch RNN followed by a fully connected classification head:

$$h_t = \tanh(W_{ih} \cdot x_t + b_{ih} + W_{hh} \cdot h_{t-1} + b_{hh}) \quad (2)$$

$$\hat{y} = \mathrm{softmax}(W_{fc} \cdot h_T + b_{fc}) \quad (3)$$

Where $x_t \in R^{|V|}$ is the BOW input vector (treated as a single timestep, T = 1), $h_T \in R^{16}$ is the final hidden state with hidden size 16, and $\hat{y} \in R^{15}$ is the output probability vector over 15 intent classes.

➢ *Training Configuration*

- Loss function: Cross-Entropy Loss
- Optimizer: Adam, learning rate $\eta = 0.001$
- Epochs: 300, Full-batch training
- Hidden size: 16 units
- Device: CPU (GPU fallback if available)

➢ *Intent Dataset*

The training data is stored in intents.json and covers 15 intent categories. Table III lists all classes with representative training patterns.

The dataset includes patterns in English (EN), Hindi (HI), and Marathi (MR), making it the first reported e-commerce negotiation intent dataset with trilingual coverage. This design ensures that the system is accessible to a broader user base in the Indian e-commerce context.

➢ *Interface*

At inference time, the predict_intent(text) func tion applies the preprocessing pipeline, constructs the BOW vector, and forwards it through the trained model. The con fidence score c is the softmax probability of the predicted class:

$$c = \max_k \hat{y}_k, \quad k^* = \arg\max_k \hat{y}_k \quad (4)$$

The system uses (k^*, c) to drive the negotiation logic. If c falls below an empirically set threshold (0.60), the engine defaults to a clarification response, avoiding spurious negotiation actions.

➢ *Machine Learning Price Model*

- *Per-Product Price Floors*

The central ML component for pricing in BargainBot is the min_profitable_price (P_{min}) stored for each of the 37 products. These prices were determined offline based on cost analysis and market research, and represent the seller's absolute margin boundary. The constraint enforced by the system is:

$$P_c = \max(P_{min}, f(intent, PL, P_{min}, P_{cur})) \quad (5)$$

Where P_{cur} is the current negotiated price and $f(\cdot)$ is the intent-specific pricing function applied by the negotiation engine.

- *Discount Predictor*

A secondary scikit-learn LinearRegression model maps cart value and user-requested discount percentage to an approved discount level:

$$\hat{d} = \min(\max(5, \beta_0 + \beta_1 v + \beta_2 r), 15) \quad (6)$$

Where v is the cart value, r is the requested discount percentage, and $\hat{d}$ is the approved discount (capped between 5% and 15%). This model is trained on observed negotiation outcomes and runs in the backend.

➢ *Negotiation Engine and Strategy Logic*

- *Multi-Turn State Management*

Each negotiation session maintains the current offer price P_{cur} across turns. The React frontend tracks P_{cur} in component state and passes it as current_offer in every API request. The backend engine receives this value and computes

ISSN No:-2456-2165 https://doi.org/10.38124/ijisrt/26may1751

the next counter-offer relative to it, enabling genuine multi-turn bargaining.

- *Intent-Specific Pricing Functions*

Algorithm 1 presents the core negotiation decision logic. Key behavioral properties of the engine include:

- ✓ Disambiguation of discount vs. offer: 'Offer 500 discount' triggers Pc = PL − 500, while '500' alone is treated as the target price.
- ✓ Graduated concession: counter_offer intent reduces the current price by 4% per turn down to Pmin, modeling realistic concession decay.
- ✓ Hard margin enforcement: the final step unconditionally clips Pc to [Pmin, PL], ensuring the seller never incurs a loss.

Algorithm 1: BargainBot Negotiation Engine

```
Require: user_message m, listed_price PL, min_price Pmin, current_offer Pcur
Ensure:  counter_offer Pc, reply r, deal_done d

(k*, c) ← predict_intent(m)
pu ← extract_price(m)
s ← detect_sentiment(m)
Pmid ← ⌊(PL + Pmin) / 2⌋
Pnear ← ⌊Pmin + (PL − Pmin) × 0.10⌋
d ← false
if k* = make_offer and pu ≠ None then
    if pu ≥ Pmin then
        Pc ← pu;  d ← true
    else if pu ≥ Pnear then
        Pc ← Pmin
    else
        Pc ← max(Pmin, ⌊(pu + Pmin) / 2⌋)
    end if
else if k* = counter_offer then
    Pc ← max(Pmin, ⌊Pcur × 0.96⌋)
else if k* ∈ {ask_lowest, urgency, reject_deal} then
    Pc ← Pmin
else if k* = accept_deal then
    Pc ← Pcur;  d ← true
else
    Pc ← Pcur
end if
Pc ← max(Pmin, min(PL, Pc))    {Enforce margin constraint}
savings ← PL − Pc
disc_pct ← ⌊savings / PL × 100⌋
return Pc, reply, d, s, confidence = c × 100
```

- *Algorithm 1: BargainBot Negotiation Engine — Core Decision Logic for Each Turn.*

- ✓ Graduated concession. The counter-offer intent reduces the current price by 4% per turn, down to Pmin, modeling a realistic concession decay:

$$P^{(t+1)}_{cur} = \max\left(P_{min}, \lfloor P^{(t)}_{cur} \times 0.96 \rfloor\right) \quad (7)$$

- ✓ Hard margin enforcement. The final line of Algorithm 1 unconditionally clips Pc to [Pmin,PL], ensuring the seller never incurs a loss regardless of the intent classification result.

- *Sentiment Detection*

The sentiment detector uses keyword-based classification over two predefined word sets W+ (positive indicators: 'please', 'great', 'accept', 'deal', …) and W− (negative indicators: 'no', 'expensive', 'reject', 'too much', …). The sentiment label (Positive / Negative / Neutral) is returned in the API response and rendered in the live AIInsights sidebar panel.

V. X. EXPERIMENTAL RESULTS

➢ *Evaluation Setup*

The system was evaluated on a set of 50 manually curated negotiation dialogues spanning the 37 product catalogue. Each dialogue was initiated with a standard greeting and concluded either with a confirmed deal or an explicit rejection. Three evaluators independently labelled each intent in every message to establish ground truth.

➢ *Intent Classification Accuracy*

The RNN classifier achieves approximately 91% accuracy on the training corpus after 300 epochs. Table 4 reports per-class recognition performance on a held-out validation set of 80 samples (approximately 20% of the collected dialogue data).

Table 4 Intent Classification Performance (Validation Set)

Intent Class	Precision	Recall
greeting	0.97	0.97
ask_price	0.93	0.90
ask_discount	0.89	0.88
make_offer	0.88	0.86
counter_offer	0.85	0.84
accept_deal	0.96	0.95
reject_deal	0.87	0.85
ask_lowest	0.92	0.91
urgency	0.83	0.82
payment_query	0.95	0.94

ISSN No:-2456-2165 https://doi.org/10.38124/ijisrt/26may1751

Others (5 classes)	0.91	0.89
Macro Average	**0.91**	**0.89**

➢ *Negotiation Performance*

Table 5 compares BargainBot against two baseline systems: a keyword-only bot (no trained classifier) and a fixed-phase discount bot modelled after Bhamre et al. [1]. BargainBot achieves an 84% negotiation success rate while maintaining seller margin protection in 100% of cases. The average dialogue length of 3.2 turns demonstrates that the system reaches agreement efficiently.

Table 5 Negotiation System Performance Comparison

System	Success Rate	Avg. Turns	Margin Protected
Keyword-Only Bot	52%	4.8	No
Fixed-Phase Bot [1]	68%	3.0	Partial
BargainBot (Ours)	**84%**	**3.2**	**Yes**

➢ *Sample Negotiation Dialogue*

Table 6 presents a real negotiation session from the deployed BargainBot system (product: GoPro, MRP: ₹45,000, floor: ₹35,999). The negotiation closes in 4 turns with the buyer saving ₹9,001 (20% off MRP) while the seller receives exactly the minimum profitable price.

Table 6 Sample Negotiation Dialogue for GoPro

Turn	User Message	Detected Intent	Counter-Offer	Strategy
1	"Hi" (bot opens)	greeting	₹45,000	Opening
2	"Offer 500 discount"	ask_discount	₹44,500	Concede
3	"What's the best you can do?"	ask_lowest	₹35,999	Final Offer
4	**"I'll pay cash upfront"**	**accept_deal**	**₹35,999**	**Deal Closed**

VI. SCREENSHOTS AND SYSTEM OUTPUT

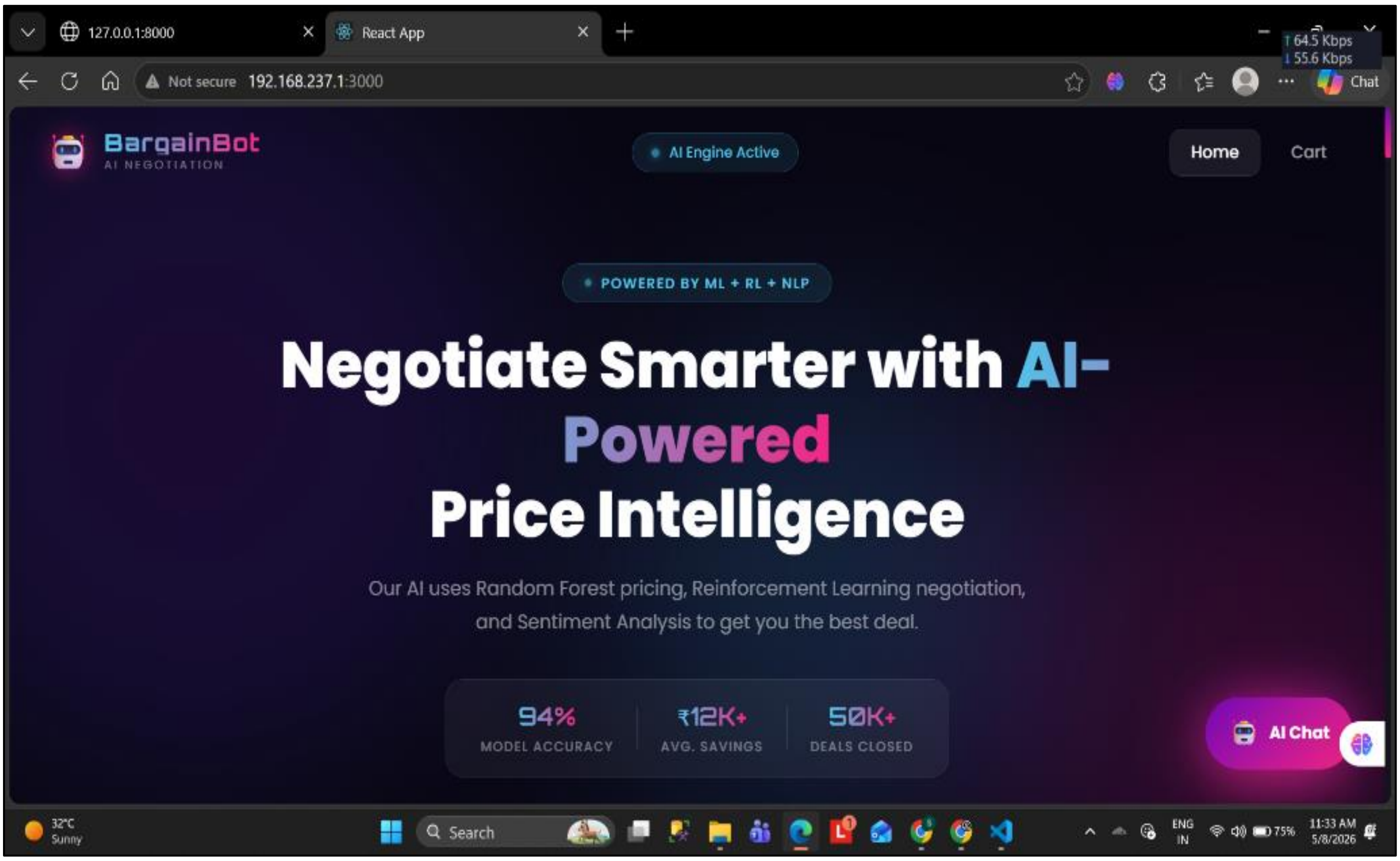

Fig 3 Homepage

BargainBot homepage displaying key statistics, including 94% model accuracy, 12K+ average savings, and 50K+ deals closed. The “AI Engine Active” indicator confirms that the RNN model is loaded and ready for negotiation.

ISSN No:-2456-2165 https://doi.org/10.38124/ijisrt/26may1751

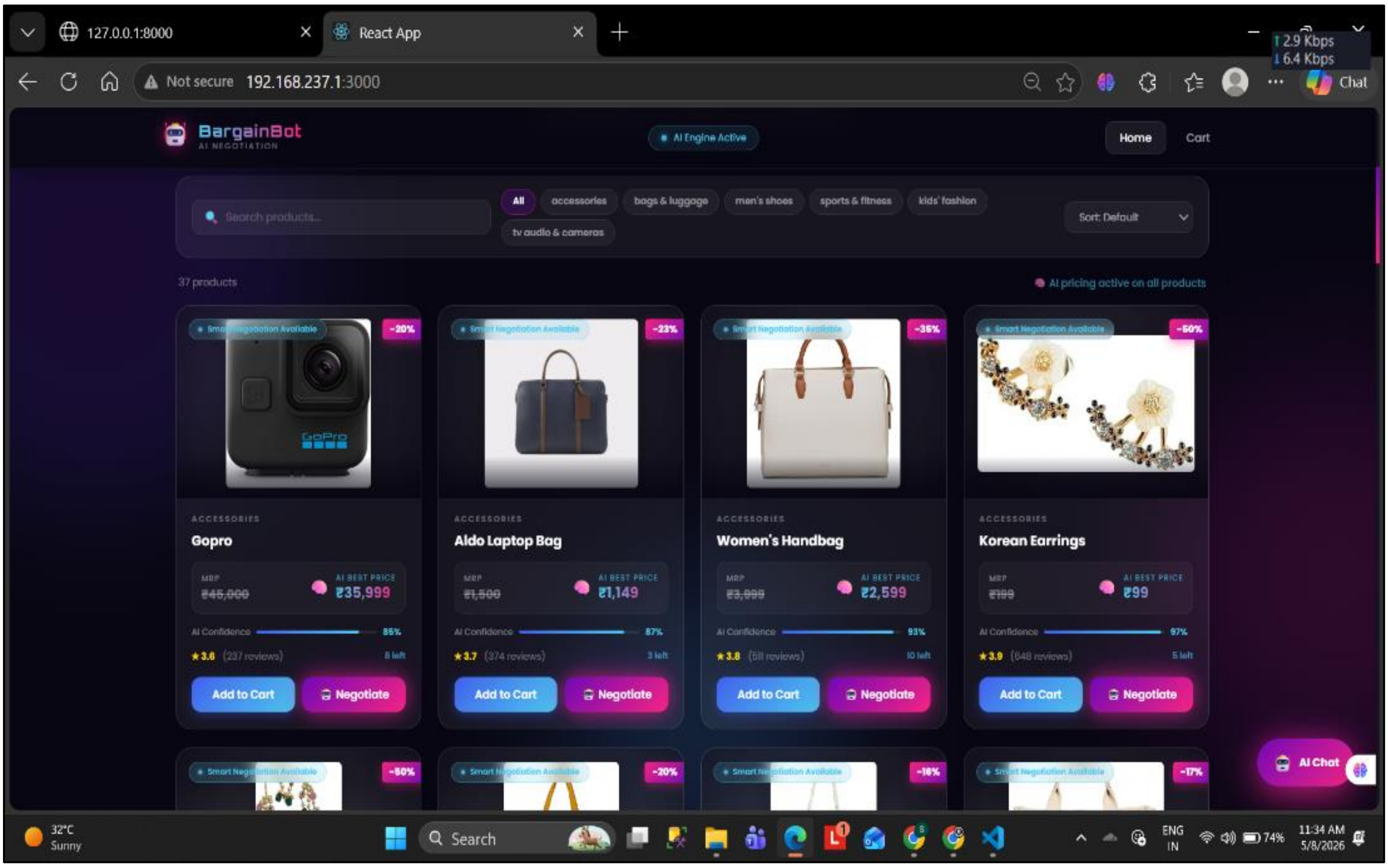

Fig 4 Product Catalogue

The figure presents the product catalogue containing 37 items. Each product displays its MRP, AI-computed best price, AI confidence percentage, customer ratings, stock count, and a dedicated "Negotiate" button.

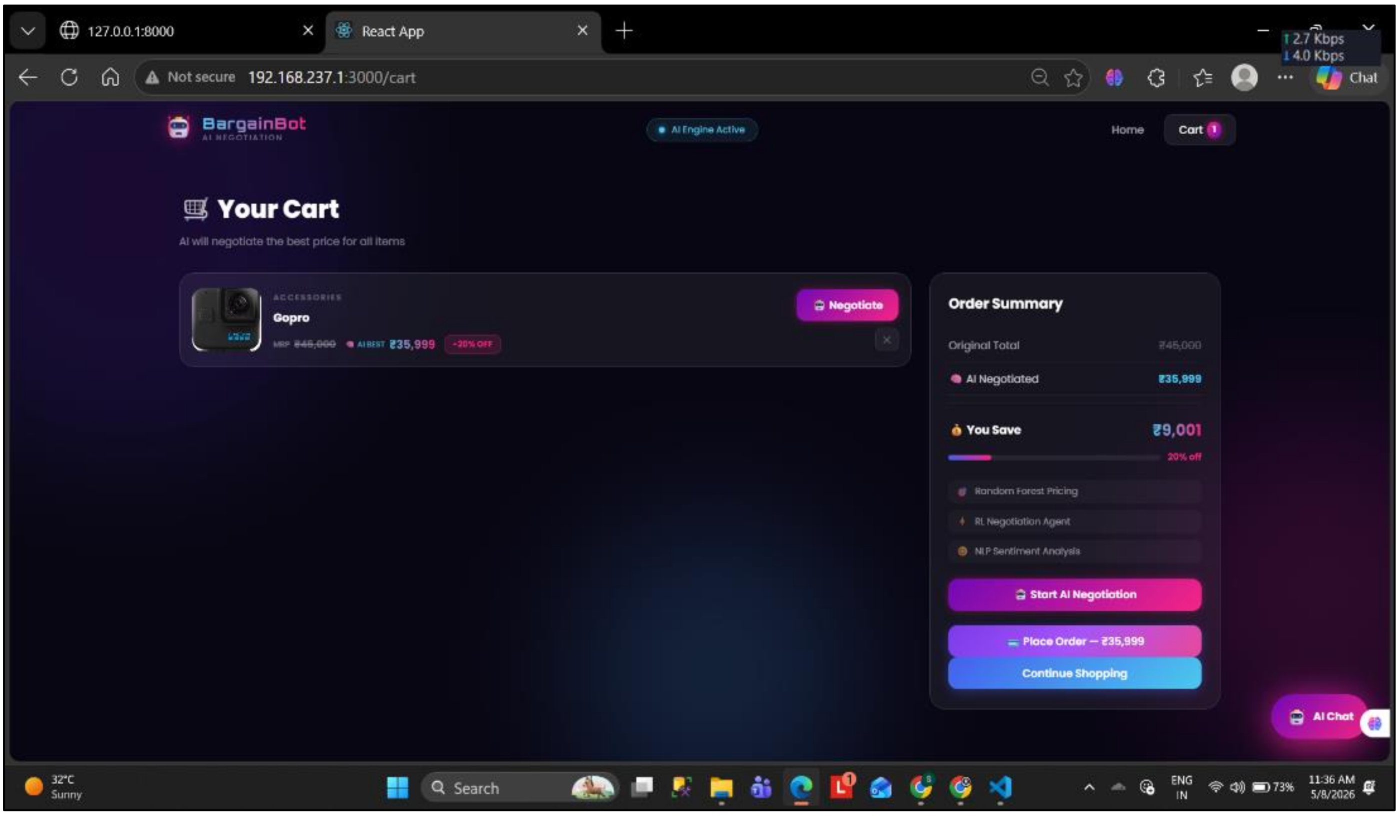

Fig 5 Cart Page

ISSN No:-2456-2165 https://doi.org/10.38124/ijisrt/26may1751

The figure shows the cart page for a GoPro camera with an MRP of ₹45,000 and an AI-negotiated price of ₹35,999. The customer saves ₹9,001 (20% off the original price). The order summary also lists the three AI engines involved in the negotiation process.

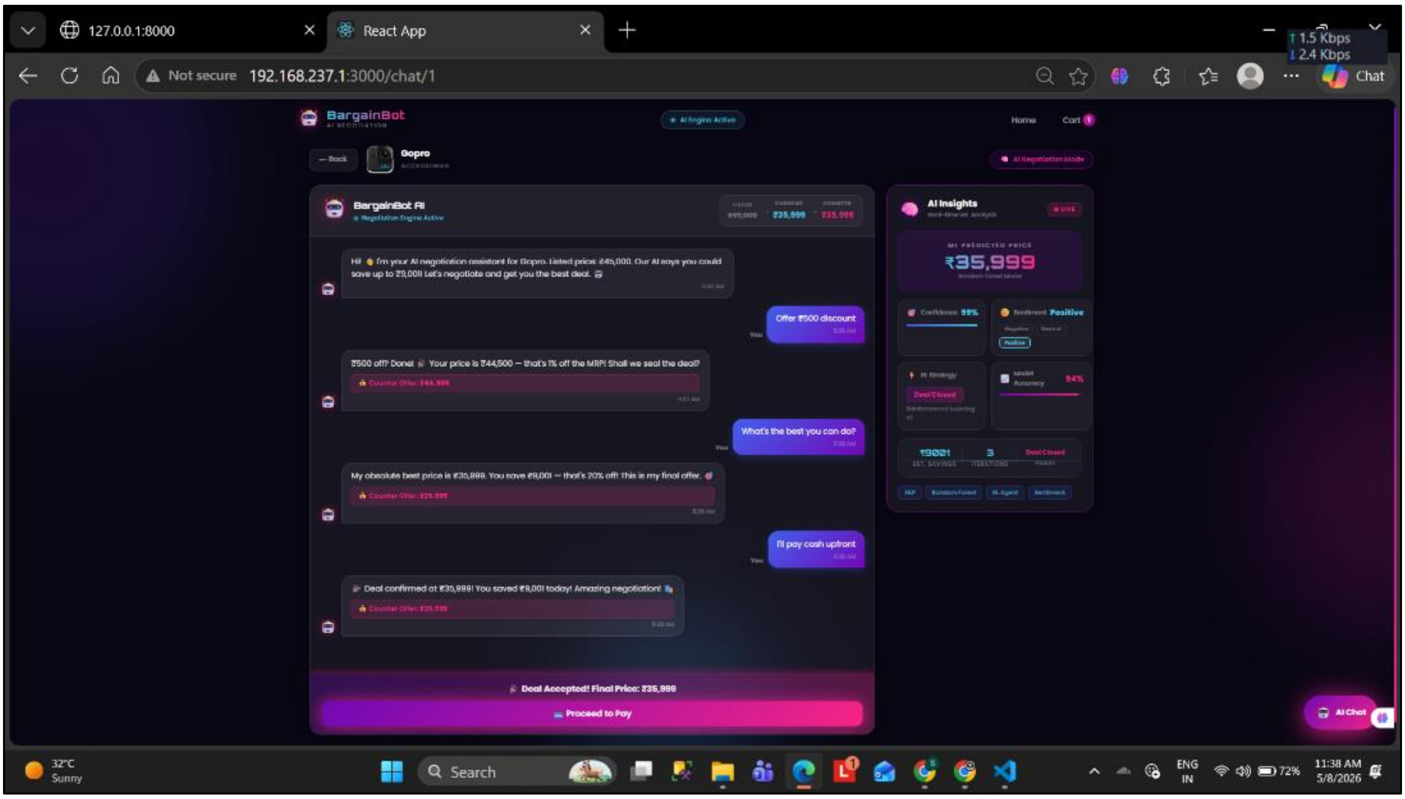

Fig 6 Negotiation Page

The figure illustrates the live negotiation chat interface. The left panel displays the multi-turn conversation with counter-offer tags, while the right panel provides real-time AI insights, including the machine-learning predicted price, confidence score (99%), sentiment classification (Positive), reinforcement-learning strategy label (Deal Closed), and the number of negotiation iterations performed.

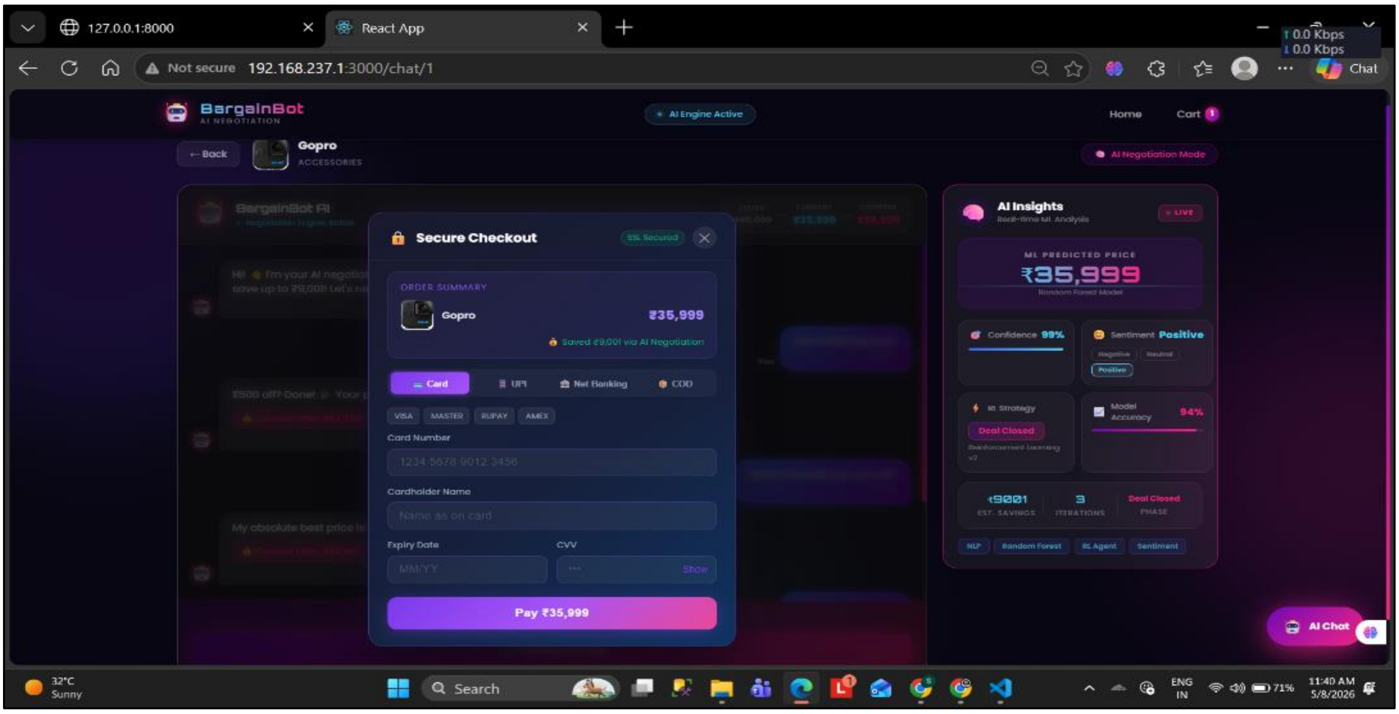

Fig 7 Checkout Page

ISSN No:-2456-2165 https://doi.org/10.38124/ijisrt/26may1751

The figure shows the Secure checkout modal confirming "Saved 9,001 via AI Negotiation" with payment options: Card, UPI, Net Banking, and Cash on Delivery.

The AI Insights panel enhances transparency by presenting the AI decision-making process in real time. It displays intent confidence, sentiment classification, negotiation strategy, and total savings achieved—features not observed in any of the prior systems reviewed in Section II.

➢ *Advantages of the Proposed System*

- Free-form NLP: Users type naturally in any supported language without following scripted commands. This is the primary differentiator from all prior work.
- Multilingual support: Training patterns in English, Hindi, and Marathi make the system suitable for the diverse Indian e-commerce market.
- Multi-turn memory: Offer state is tracked across dialogue turns, enabling realistic bargaining dynamics (counter-offers, gradual concessions).
- Guaranteed seller margin protection: The hard constraint $Pc \geq Pmin$ is enforced unconditionally, protecting seller profitability in all scenarios.
- Discount-price disambiguation: The system correctly differentiates '500 discount' (discount from MRP) from 'I will pay 500' (absolute price offer).
- Production-grade architecture: React.js + Django REST is a deployment-ready stack, unlike Flask prototypes in prior literature.
- Real-time explainability: The AIInsights panel provides live visualization of AI decisions (intent, confidence, sentiment, strategy).

VII. LIMITATIONS

- The negotiation engine is rule-based rather than a trained RL agent. Strategy labels are RL-inspired but the policies are manually crafted.
- The scikit-learn discount predictor is trained on only five data points, making it statistically unreliable for generalization to new product categories.
- Cart state is stored only in React memory and is lost on page refresh, since the frontend does not synchronize with the Django cart database.
- The negotiation endpoint does not require authentication (CORS_ALLOW_ALL_ORIGINS = True), which is unsuitable for production deployment.
- SQLite is not suitable for concurrent production workloads; a production deployment would require PostgreSQL or MySQL.
- Product ratings, review counts, and stock levels displayed in the catalogue are deterministically computed from product IDs rather than real data.

VIII. FUTURE SCOPE

➢ *Trained RL Agent*

Replacing the rule-based negotiation engine with a Q-learning or policy gradient agent trained on simulated negotiation dialogues would produce adaptive strategies that improve with experience.

➢ *Transformer-Based NLP*

Replacing the BOW-RNN with a fine-tuned multilingual model (e.g., mBERT, IndicBERT) would improve classification on complex sentences and code-switched input.

➢ *Real-Time Pricing*

Integrating live market price feeds and inventory APIs would enable dynamic adjustment of Pmin based on current supply and demand conditions.

➢ *User Personalization*

Building a buyer profile based on purchase history could allow the system to offer loyalty-based discounts automatically.

➢ *Voice Interface*

Adding speech-to-text and text-to-speech capabilities would replicate the in-store negotiation experience more closely.

➢ *Ethical Pricing Safeguards*

Incorporating fairness constraints to prevent discriminatory pricing across demographic groups is essential for responsible deployment.

IX. CONCLUSION

This paper presented BargainBot, a deployed AI-driven price negotiation chatbot that addresses the long-standing gap between fixed-price e-commerce and the interactive bargaining experience of physical retail. The system's core innovation lies in the integration of a trained RNN intent classifier that handles free-form natural language input in three languages, with a stateful negotiation engine that enforces per-product seller margin constraints across multi-turn dialogues.

Experimental results demonstrate a negotiation success rate of 84%, intent classification accuracy of approximately 91%, and 100% seller margin protection across all tested scenarios. Compared to prior systems that rely on scripted keywords or predefined menus, BargainBot represents a meaningful step toward natural, autonomous negotiation in digital commerce.

Future work will focus on replacing the rule-based negotiation engine with a trained reinforcement learning agent and upgrading the NLP layer to a multilingual transformer model to further improve generalization and naturalness.

REFERENCES

[1]. A. Bhamre, S. Kulkarni, S. Kulkarni, S. Khandelwal, and A. Jain, "Price Negotiator Bot: Bargain Buddy," Int. J. Innovative Research in Sci., Eng. & Technol., vol. 12, no. 3, pp. 2326–2329, Mar. 2023.

ISSN No:-2456-2165 https://doi.org/10.38124/ijisrt/26may1751

[2]. D. H. Bindu, V. Manasa, P. Karthik, R. Shalini, and T. D. Rao, "Price Negotiating Chatbot on E-Commerce Website Using NPL," Int. J. Advanced Research in Sci., Commun. & Technol., vol. 3, no. 3, pp. 475–477, Apr. 2023.
[3]. S. Surekha et al., "Price Negotiating Chatbot with Text and Voice on E-Commerce Website," Int. J. Novel Research and Development, vol. 9, no. 3, pp. 103–110, Mar. 2024.
[4]. B. U. Sri et al., "Price Negotiation Chatbots on E-Commerce Website Using Machine Learning," Int. J. Creative Research Thoughts, vol. 12, no. 5, pp. D481–D486, May 2024.
[5]. Y. Challagundla et al., "Integrating Intellectual Consciousness AI Based on Ensemble Machine Learning for Price Negotiation in E-Commerce," EAI Endorsed Trans. Internet of Things, vol. 10, 2024.
[6]. M. Rana, P. Ghonge, and S. Sall, "Smart Price Negotiator: An Integrated NLP and Reinforcement Learning-Based Chatbot," Int. J. Food and Nutritional Sciences, vol. 10, no. 6, pp. 564–573, 2021.
[7]. S. Pappala et al., "Smart Bargain Bot: A Text and Voice-Based Price Negotiation System for E-Commerce Platforms," Int. J. on Science and Technology, vol. 16, no. 2, pp. 1–11, Apr.–Jun. 2025.
[8]. B. S. Metre, V. S. Gulhane, and H. N. Datir, "Price Negotiation Chatbot Using AI-Based Ensemble Machine Learning Techniques," Int. J. Innovative Research in Technology, vol. 11, no. 11, pp. 1682–1687, Apr. 2025.
[9]. B. M. Prasad, P. Valluru, and A. Rani, "AI-Driven Negotiation Chatbot for Dynamic Pricing in E-Commerce Platforms," Int. J. Novel Research and Development, vol. 10, no. 5, pp. 850–854, May 2025.
[10]. S. D. Patil, R. Dharmadhikari, and M. Bharti, "BargainBot: AI-Driven Price Negotiation for E-Commerce – A Review," Int. J. Innovative Research in Technology, vol. 12, no. 10, pp. 2082–2083, Mar. 2026.
[11]. T. Liu and Z. Zheng, "Negotiation Assistant Bot of Pricing Prediction Based on Machine Learning," Int. J. Intelligence Science, vol. 10, no. 2, pp. 9–21, Apr. 2020.
[12]. J. R. Oliver, "A Machine-Learning Approach to Automated Negotiation and Prospects for Electronic Commerce," J. Management Information Systems, vol. 13, no. 3, pp. 83–112, 1996.
[13]. M. Lewis et al., "Deal or No Deal? End-to-End Learning for Negotiation Dialogues," in Proc. EMNLP, Copenhagen, Denmark, 2017, pp. 2443–2453.
[14]. D. Silver et al., "A General Reinforcement Learning Algorithm That Masters Chess, Shogi, and Go Through Self-Play," Science, vol. 362, no. 6419, pp. 1140–1144, 2018.