Journal of Theoretical and Applied Information Technology
31st October 2026. Vol.104. No.20
© Little Lion Scientific
ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

AN INTELLIGENT MULTI-AGENT AGENTIC AI FRAMEWORK FOR RETAIL SUPPLY CHAIN OPTIMIZATION AND LSTM-BASED DEMAND PREDICTION

MALLANNAGARI SUNITHA[1], THONDEPUADILAKSHMI[2], DR.B. JYOSHNA[3], DR.R. SATEESH KUMAR[4], DR.B. KIRANMAI[5]

[1]associateprofessor, Department of Computer Science Engineering, Vasavi College of Engineering, Telangana, India
[2]Director&Professor, Department of Computer Science Engineering, Vasavi College of Engineering, Telangana, India
[3, 5]Associate Professor, CSE(DS) , Sreyas Institute Of Engineering And Technology , Bandlaguda, Telangana, India
[4] Associate Professor, Department Of Computer Science And Engineering Koneru Lakshmaiah Educational Foundation Bowrampet, Hyderabad-500043, Telangana, India

E-mail: [1]m.sunithareddy@staff.vce.ac.in,[2]t adilakshmi@staff.vce.ac.in,[3]drjyoshnabejjam@gmail.com, [4]sateeshramatenki@ Klh.edu.in, [5]kiranmaimtech@gmail.com

ABSTRACT

Traditional retail inventory evaluation and control systems rely heavily on static threshold criteria and predetermined lot-sizing rules that lack adaptability during compound supply chain disruptions and fail to incorporate multi-modal contextual information. This paper proposes a comprehensive five-layer Agentic artificial intelligence framework that synergistically integrates four independently developed technologies into a unified, deployable retail supply chain simulation environment: Large Language Model (LLM) ReAct reasoning, Long Short-Term Memory (LSTM) primary demand forecasting, ChromaDB episodic memory retrieval, and LangGraph-orchestrated state machine workflow control. The multi-agent architecture comprises four collaborating autonomous actors Demand, Warehouse, Supplier, and Logistics agents executing on top of a discrete-event Mesa simulation scaffold. Each agent generates structured, auditable decision records (Decision Record) directly at inference time. In a rigorous 30-day stress simulation bench marking against classical rule-based control policies under compound multi-stage disruptions, the proposed agentic system achieves an approximate 9 percentage point improvement in order fill rate (86%to95%), a 71% reduction in stock out days (21days down to 6 days), and a35% decline into supply chain operational cost ($57k down to $37k), driven primarily by eliminating catastrophic stock out penalties. Component ablation demonstrates that LangGraph conditional state routing and Message Bus peer-to-peer communication contribute most significantly to system's ability. Comprehensive mathematical modeling, workflow state machines, and operational counter measures are detailed to support enterprise deployment.

Keywords: *Multi-Agent Systems, Retail Supply Chain, LLM Agents, LSTM Forecasting, LangGraph*

1. INTRODUCTION

Inventory management in contemporary retail supply networks operates under the persistent tension between conflicting operational objectives: maintaining high product availability to prevent stock outs while constraining working capital locked in holding and warehousing costs. Conventional inventory management models, such as the Economic Order Quantity (EOQ)[1] and periodic (*s,Q*) reorder point policies [2], govern inventory dynamics through static mathematical thresholds derived under stationary demand assumptions. However, modern supply chains routinely encounter non-stationary disruptions, such as unexpected supply routes, transportation bottle-necks, severe weather phenomena, and synchronized consumer demand surges [3–5]. Under such compound stresses, static reorder rules exhibit

acute structural fragility, frequently precipitating the bullwhip effect [6].
In modern retail environments, customer demand is heavily influenced by exogenous factors including hyper local weather shifts, promotional events, competitive price adjustments, and regional macroeconomic developments. Traditional Enterprise Resource Planning (ERP) systems and classical Material Requirements Planning (MRP) heuristics are structurally in capable of ingesting unstructured, multi-modal real-world context. Consequently, inventory controllers frequently operate reactively: replenishment orders are triggered only after physical stock has breached Critical reorder thresholds, resulting in severe back orders and costly expedited logistics.

Agent-Based Modeling (ABM) [7–9] partially mitigates these limitations by simulating decentralized supply chain participants as autonomous inter-acting entities. Nevertheless, traditional AB Magnets remain bounded by rigid heuristic rule sets, lacking the contextual adaptability and semantic reasoning necessary to interpret unstructured external market signals. Recent breakthroughs in Large Language Model (LLM) agent architectures [10–13] present an opportunity to bridge this gap. By interleaving natural language reasoning with discrete action execution, the ReAct paradigm [12] LLM agents can synthesize heterogeneous information, deliberate on trade-offs, and dynamically adjust operational policies. Despite the promise of generative artificial intelligence, deploying LLMs directly into mission counter factual alternative sat every decision node, ensuring full enterprise auditability.

The specific contributions of this work are summarized as follows:

- **Integrated Five-Layer Architecture:** We design and implement an additive five-layer sys-tem architecture combining discrete-event simulation (Mesa), deep time-series forecasting (LSTM), semantic episodic retrieval (ChromaDB), state-machine routing (LangGraph), and external multimodal intelligence synthesis into a single deployable framework.
- **Constrained Agentic Reasoning Protocol:** We formulate mathematical state-space transitions and deterministic capacity enforcement bounds ($a^{exec} = \min(a^{LLM}, C_v)$) that clamp LLM generator since of hard physical constraint enforcement, and lack of temporal memory across extended operational horizons [14, 15]. Furthermore, commercial LLMs operate as probabilistic black boxes, whereas sup-ply chain executives require deterministic workflow bounds, fail-safe fallback guarantees, and granular auditability for compliance and fiduciary accountability. To overcome these fundamental limitations, this paper proposes an end-to-end, five-layer Agentic AI Framework that orchestrates specialized multi-agent reasoning for retail supply chain decision support. The framework establishes a unified operational architecture where:

i. LLM inference (LLaMA 3.1 8B Instant via Groq) serves as the primary cognitive decision layer, executing structured ReAct deliberation.

ii. An LSTM neural network functions as the predictive anticipation layer for time-series demand forecasting, backed by Prophet and historical moving average fallbacks.

iii. ChromaDB vector database acts as an episodic memory bank, enabling retrieval-augmented decision grounding from historical operational episodes.

iv. LangGraph directed-graph state machine pro-vides deterministic workflow orchestration, conditional emergency escalation, and Human-in-the-Loop (HITL) risk gating.

v. An inline structured logging mechanism (DecisionRecord) captures complete operational rationales, contributing variables, and actions to physical operational limits, eliminating hallucination risks.

- **Inline Decision Traceability Mechanism:** We introduce the DecisionRecord paradigm, providing 100% audit log completeness with template-driven slot-filling fallback during API outages.
- **Empirical Stress Evaluation and Ablation:** We execute a comprehensive 30-day multi-

phase compound disruption simulation, isolating the quantitative contributions of each layer and demonstrating superior resilience over classical heuristics.

The remainder of this paper is structured as follows: Section 2.0 surveys literature across classical inventory theory, ABM, and LLM reasoning. Section 3.0 details the five-layer architecture and mathematical state-space formulation. Section 4.0 describes the methodology, forecasting pipeline, and cost evaluation models. Section 5.0 presents empirical results from 30-day stress simulations and ablation studies. Section 6.0 provides systematic comparative analysis. Section 7.0 addresses research challenges and countermeasures. Section 8.0 concludes with future directions.

2.0 LITERATURE REVIEW

2.1 Classical Inventory Optimization

The foundation of modern inventory theory rests upon Harris's Economic Order Quantity (EOQ) formulation [1], which establishes the optimal order quantity Q^* that minimizes total annual holding and ordering costs: (DecisionRecord) captures complete operational rationales, contributing variables, and

$$Q^* = \frac{2 * (DS / H)}{} \quad (1)$$

here D denotes annual deterministic demand, S rep-resents setup/ordering cost per transaction, and H denotes per-unit annual holding cost. While analytically tractable, the classical EOQ assumes constant, deterministic demand and zero lead-time variability. Stochastic extensions, such as continuous-review (s, Q) and periodic-review (R, S) policies [2, 25], in-corporate safety stock to absorb demand uncertainty. However, these systems remain inherently reactive: replenishment triggers only after on-hand inventory drops below the reorder point s, leaving the supply chain highly vulnerable to upstream lead-time spikes or compound disruptions [4, 26].

2.2 Agent-Based Supply Chain Simulation

Agent-Based Modeling provides a bottom-up paradigm for simulating complex supply networks. Swaminathan et al. [8] demonstrated that modeling supply chain participants (retailers, distributors, manufacturers) as discrete agents interacting through message queues accurately reproduces endogenous dynamics, including order oscillations and lead-time delays. Modern ABM frameworks, such as Mesa [7] and NetLogo [27], provide scalable Python and Java event loops for simulating multi-actor systems. How-ever, conventional ABM implementations rely on predefined deterministic rules or lookup tables, pre-venting agents from contextually interpreting non-standard events such as geopolitical news or severe weather warnings.

2.3. LLM-Driven Agent Reasoning and ReAct

The integration of Large Language Models into autonomous agents has transformed cognitive workflow automation [10, 13]. Yao et al. [12] introduced the ReAct (Reasoning + Acting) framework, which inter-leaves chain-of-thought deliberative reasoning traces with discrete tool invocation steps. By decomposing complex goals into sequential Thought-Action-Observation loops, ReAct agents exhibit superior grounding and reduced hallucination rates compared to pure direct-prompting architectures. In supply chain contexts, Shen et al. [14] demonstrated that LLMs can interpret textual market indicators to en-hance demand sensing, although their application within fully orchestrated, closed-loop multi-agent sup-ply chain simulations remains nascent.

2.4. Workflow Orchestration with LangGraph

Standard autonomous agent frameworks often suffer from unbounded execution loops and unpredictable state divergence. LangGraph [17] addresses this challenge by framing agentic workflows as cyclic, stateful directed graphs where nodes represent agent computation functions and edges represent conditional routing logic. This paradigm enables deterministic structural boundaries to be wrapped around probabilistic LLM outputs, enforcing business logic constraints and enabling controlled human intervention.

2.5. Retrieval-Augmented Episodic Memor

Standard prompt windows constrain the longitudinal operational context available to LLM agents. Lewis et l. [18] established Retrieval-Augmented Generation (RAG), augmenting language model prompts with semantically relevant external documents. Applying RAG principles to episodic agent memory involves storing historical state-action-outcome tuples in vector databases such as ChromaDB [19]. By retrieving top-k historical episodes using dense semantic similarity embeddings [20], agents leverage experiential memory to inform present decisions without requiring continuous fine-tuning.

2.6. Demand Forecasting: LSTM and Prophet Accurate demand forecasting is critical for shifting supply chain operations from reactive buffering to proactive replenishment. Hochreiter and Schmid-huber [21] developed Long Short-Term Memory (LSTM) recurrent neural networks, which resolve the vanishing gradient problem through learned gating mechanisms:

$$f_t = \sigma(W_f \cdot [h_{t-1}, x_t] + b_f) \quad (2)$$

$$i_t = \sigma(W_i \cdot [h_{t-1}, x_t] + b_i) \quad (3)$$

$$\tilde{C}t = \tanh(W_c \cdot [h_{t-1}, x_t] + b_c \quad (4)$$

$$C_t = f_t \odot C_{t-1} + i_t \odot \tilde{C}t \quad (5)$$

$$o_t = \sigma(W_o \cdot [h_{t-1}, x_t] + b_o) \quad (6)$$

$$h_t = o_t \odot \tanh(C_t) \quad (7)$$

where f_t, i_t, and o_t represent forget, input, and output gates, C_t is the cell state, and h_t is the hidden state vector. Complementing recurrent networks, Taylor and Letham [22] introduced Prophet, a generalized additive time-series model decomposing seasonal, trend, and holiday components:

$$y(t) = g(t) + s(t) + h(t) + \varepsilon_t \quad (8)$$

Makridakis et al. [24] demonstrated through the M5 forecasting competition that combining hybrid ma-chine learning models with domain heuristics significantly outperforms traditional exponential smoothing in retail contexts.

2.7. Inline Traceability vs. Post-Hoc Explainability

Explainable AI (XAI) approaches, such as LIME [28] and SHAP [29], construct post-hoc surrogate models to approximate black-box predictions. However, post-hoc approximations cannot guarantee fidelity to the actual decision-making rationale employed at inference time. In enterprise supply chains subject to strict regulatory and fiduciary oversight, decision records must capture the exact reasoning chain, contributing parameters, and reject counterfactual alternatives synchronously with execution [30, 31] and shown in Table 1.

Table 1. Taxonomy of Related Supply Chain Works

Paradigm	Decision Engine	Memory	Control	Traceability
Classical (s, Q) [2]	Fixed Thresholds	None	Linear	Formulaic
ABM [8]	Production Rules	Queues	Loop	State Logs
LSTM / Prophet [21]	Decoupled Policy	None	Batch	None
Pure LLM [11]	Generative LLM	Context	Dialog	Free Text
Proposed	**Clamped ReAct**	**ChromaDB**	**LangGraph**	**Structured**

3.0 SYSTEM ARCHITECTURE

3.1. Five-Layer Progressive Architecture

The system is organized into five layered subsystems as shown in figure 1, where each layer introduces additive intelligence while preserving lower-layer fallbacks:

- **Layer 1 — Mesa ABM Scaffold:** Core discrete-event simulation engine managing environmental time ticks, physical inventory accounting, and rule-based baseline agents.
- **Layer 2 — LLM ReAct Reasoning**

Journal of Theoretical and Applied Information Technology
31st October 2026. Vol.104. No.20
© Little Lion Scientific
ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

Layer: Groq API-accelerated LLaMA 3.1 8B Instant inference engine executing structured JSON Re-Act prompts for each supply chain actor.

- **Layer 3 — Episodic Memory and Message-Bus:** ChromaDB vector store performing cosine-similarity episodic retrieval, coupled with an asynchronous in-memory Message Bus pub/sub router as shown in Table 2.
- **Layer 4 — LangGraph Workflow and Decision Logging:** Stateful directed execution graph enforcing conditional routing, emergency state transitions, and inline DecisionRecord generation.
- **Layer 5 — External Intelligence and HITL Governance:** Environmental data ingestion (weather, news headlines) synthesized via LLM

$$S_{t+1} = F(S_t, A_t) \quad (9)$$

3.2. Agent Topology and Mathematical State Transition

The supply chain is modeled as a directed multi-agent graph $G = (V, E)$ consisting of four specialized autonomous agents:

$V = \{\text{Demand}, \text{Warehouse}, \text{Supplier}, \text{Logistics}\}$

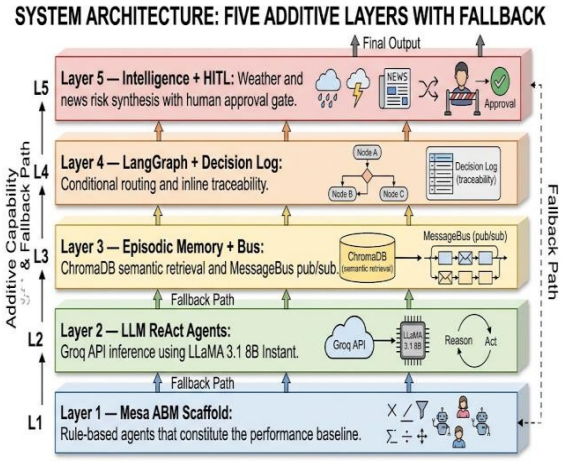


Figure 1. Five-Layer Progressive Agentic Architecture with Fallback Paths

interconnected via operational dependencies:

$E = \{(D \rightarrow W), (W \rightarrow S), (S \rightarrow L), (L \rightarrow W)\}$ (10)

At simulation timestep t, each agent $v \in V$ observes an individualized state tuple and executes a decision policy π_v:

$$a^t = \pi_v(s_t, M_v(s_t), B_v(t), I_t) \quad (11)$$

where s_t denotes current local state variables, $M_v(s_t)$ represents top-k retrieved episodic memories, $B_v(t)$ is the incoming message queue from the MessageBus, and I_t represents the synthesized external intelligence vector.

To prevent agent hallucinations from violating physical operational boundaries, proposed actions undergo deterministic feasibility clamping:

$$a^{\text{exec}} = \min a^{\text{LLM}}, C_v \quad (12)$$

where C_v represents the hard physical capacity ceiling (e.g., maximum warehouse storage volume $C_{\text{warehouse}} = 1,000$ units, or supplier batch capacity $C_{\text{supplier}} = 500$ units/order). The global state transi-tion function across the entire network is given by:

$$S' = F(S, A), A = \{a_{\text{exec}} \mid v \in V\} \quad (13)$$

3.3. Autonomous Supply Chain Agents Specification

The four specialized agents possess dedicated operational responsibilities:

Demand Agent: Executes daily LSTM inference over recent sales series, ingests external weather/news risk multipliers, detects demand anomalies, and broadcasts expected demand forecasts to the Message Bus.

Warehouse Agent: Tracks physical on-hand inventory I_t, pipeline inventory in transit, calculates safety buffers, determines reorder quantities Q_t, and initiates expedited replenishment during emergency states.

Journal of Theoretical and Applied Information Technology
31st October 2026. Vol.104. No.20
© Little Lion Scientific
ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

Supplier Agent: Evaluates incoming purchase orders against current production capacity, raw material availability, and disruption schedules, determining committed batch fulfillment quantities

LangGraph Conditional State Routing Logic The LangGraph workflow controller governs daily execution sequence and dynamically routes execution paths based on operational severity:

NORMAL Path: Invoked when inventory level $I_t \geq \theta$ ($\theta = 200$ units) and no disruptions are flagged. The system executes standard daily forecasting and reorder evaluations.

Emergency Path: Invokes when $I < \theta_v$

The Warehouse and Supplier agents are triggered to evaluate expedited replenishment orders.

CRISIS Path: Invoked when $I_t = 0$ or when an active disruption event coincides with projected demand exceeding on-hand plus in-transit stock:

$$\text{Path} = \text{CRISIS} \Longleftrightarrow (I_t = 0) \vee \text{Disruption} \wedge \hat{D}_{t+\tau} > I_t + I_{\text{transit}} \quad (14)$$

3.4 Inter-Agent Message Bus Protocol

Algorithm 1: Multi-Agent Simulation Daily Execution Loop

Require: Timestep t, Graph G(V, E), State S_t, Thresholds θ, Θ
Ensure: Updated State S_{t+1}, Audit Log L_t

1: I_t ← GetExternalIntelligence(t)
2: D̂_{t+1} ← ExecuteLSTM(SalesHistory_{t-6:t}, material availability and disruption schedules)
3: D̂'_{t+1} ← ApplyRiskMultiplier(D̂_{t+1}, I_t, α)
// determining committed batch fulfillment quantities.
// Logistics Agent: Manages freight dispatch, assigns shipping
// carriers, monitors transit delays and road/weather bottlenecks,
// and calculates arrival dates for in-transit shipments.
4: MessageBus.Publish("demand forecast", D̂'_{t+1})
5: Path ← LangGraphEvaluateRouting(S_t, D̂'_{t+1}, θ)
6: for each agent v ∈ {Warehouse, Supplier, Logistics} do
7: M_v ← ChromaDBRetrieve(s_t, k = 3)
8: B_v ← MessageBus.FetchInbox(v)
9: a^{LLM}_v ← LLM_ReAct(v, s_t, M_v, B_v, I_t, Path)
10: a^{exec}_v ← min(a^{LLM}_v, C_v)
11: if v = Warehouse ∧ a^{exec}_v > Θ then
12: A={aexec | v ∈ V} HumanInTheLoopApprovalGate(a^{exec}_v)
13: end if
14: L ← L_t ∪ CreateDecisionRecord(v, a^{exec}_v, Path)
15: MessageBus.DispatchUpdates(v, a^{exec}_v)
16: end for
17: S_{t+1} ← MesaStepPhysicsEngine(S_t, {a^{exec}_v})
18: ChromaDBStoreEpisodes({s_t, a^{exec}_v, S_{t+1}}
19. MessageBus.PruneExpiredMessages(TTL = 3)
20: return St+1, Lt

Logistics Agent: Manages freight dispatch, as-signs shipping carriers, monitors transit delays and road/weather bottlenecks, and calculates arrival dates for in-transit shipments.

Table 2: MessageBus Message Types And Triggers

Message Type	Sender	Recipient	Trigger Condition
Demand Forecast	Demand	All	Daily forecast (tt)
Inventory Alert	Warehouse	Supplier, Logistics	Stock It<θIt<θ
Order Request	Warehouse	Supplier	Reorder (Qt>0Qt>0)
Capacity Status	Supplier	Warehouse	Bottleneck update
Shipment Update	Logistics	Warehouse	Dispatch / delay

3.6 Inline Decision Logging and Auditability Every agent action produces one Decision Record at inference time, capturing simulation day t, agent identifier v, decision type, executed action value a_{exec}, workflow routing

$$\hat{D}_{t+\tau} > I_t + I_{\text{transit}} \quad (15)$$

path, execution mode (LLM vs. fallback), confidence score ∈ [0, 1], and a structured why object containing ranked contributing factors and rejected alternatives. Fallback decisions receive template-generated explanations slot-filled from current state variables.

4.0 METHODOLOGY

4.1 Demand Forecasting Pipeline

Historical demand data is derived from the Walmart M5 Dataset [24], extracting the CA 1 food category SKU. The synthetic baseline is formulated as:

$$y_t = 100 + 20\sin\left(\frac{2\pi t}{T_1}\right) + 15\sin\left(\frac{2\pi t}{30}\right) + \varepsilon_t \quad (15)$$

Where Noise Term $\varepsilon_t \sim N(0,100)$

Table 3. LSTM Forecasting Network Architecture

Layer	Units	Dropout	Activation
LSTM-1	64	0.20	tanh / recurrent sigmoid
LSTM-2	32	0.20	tanh / recurrent sigmoid
Dense	16	–	ReLU
Output	1	–	Linear

4.2 LLM Agent Reasoning Specification

Each agent executes its decision function via Groq-hosted LLaMA 3.1 8B Instant models. Structured JSON ReAct prompts aggregate Role, State, Memory, Messages, and Environmental Intelligence. API retries follow capped exponential backoff as shown in Table 4.

$$\Delta_k = \min(2^k \cdot 1.0\,s, 30\,s),\ k \in \{1,2\} \quad (16)$$

Table 4. Per-Agent Scoring Weights And Pass Conditions

Agent	Primary Metric	Weight	Pass Condition
Supplier	Ful fillment Ratio (Ft / Rt)	1.00	Score ≥ 80.0
Warehouse	Days-of-Stock Buffer	1.00	Score ≥ 80.0
Logistics	On-Time Delivery Ratio	1.00	Score ≥ 80.0
Demand	1−min (2ε, 1)1−min(2ε, 1)	1.00	Score ≥ 80.0

4.3 Episodic Memory Mechanics

Historical episodes are embedded using all-MiniLM-L6-v2 (384-d dense vectors) and indexed in ChromaDB [19]. Top-k ($k = 3$) retrieval computes cosine similarity:

$$M_v(s_t) = \text{Top-}k\ \{\cos(\mathbf{e}(s_t), \mathbf{e}(e_j)) \mid e_j \in D_v\} \quad (17)$$

4.4 External Intelligence Ingestion

Environmental signals (Open-Meteo weather fore-casts and DuckDuckGo news headlines) are synthesized via LLM into an intelligence risk object. The demand forecast is adjusted via

multiplier α:

$$yt+1' = \alpha \cdot y\hat{}t+1 \quad (18)$$

subject to $|\alpha - 1.0| > 0.05$

4.5 Human-in-the-Loop (HITL) Gate

When proposed reorders exceed threshold $\Theta = 400$ units, LangGraph interrupts execution at the supplier decision node and prompts human authorization before releasing purchase orders.

5.0 EXPERIMENTS AND RESULTS

5.1 Simulation Configuration

Simulations are executed across a 30-day operational horizon under an identical compound disruption schedule to ensure rigorous comparability as shown in Table 5.

Table 5. Simulation Parameter Configuration

Parameter	Value	Unit
Simulation Horizon	30	Days
Initial Inventory (I_0)	300 (Stress)	Units
Warehouse Capacity	1,000	Units
Supplier Capacity	500	Units / Order
Supplier Reliability	95.0	%
Nominal Lead Time	3	Days
Reorder Threshold (ϑ)	200	Units
LLM Engine	LLaMA 3.1 8B	Groq API
Forecasting Model	2-Layer LSTM	TensorFlow
HITL Ceiling (Θ)	400	Units

5.2. Demand Forecasting Performance

The LSTM network architecture shown in Table 3 achieved superior prediction accuracy compared to the 7-day Moving Average base-line as shown in Table 6.

Table 6. Demand Forecasting Performance Comparison

Metric	LSTM Model	7-Day MA	Difference (Δ)
MAE	≈ 9.0 u	≈ 15.0 u	−6.0 u (−40.0%)
RMSE	≈ 11.0 u	≈ 20.0 u	−9.0 u (−45.0%)
MAPE	≈ 11.0%	≈ 19.0%	−8.0 pp
Bias	+1.0 u	+4.0 u	Reduced over-bias

5.3. Supply Chain KPI Benchmark

Table 7 summarizes end-to-end performance across the 30-day stress scenario. The agentic system achieves an order fill rate of 95% compared to 86% for the rule-based baseline, reducing stock out days from 21 down to 6 days.

Table 7. End-to-End Supply Chain KPI Comparison

Metric	Rule	Agentic	Δ	Gain (%)
Fill Rate (%)	≈86%	≈95%	+9pp	+10.5%
Stockout Rate (%)	≈17%	≈6%	−11pp	−64.7%
Avg Inventory (u)	≈180	≈260	+80	—
Resilience Index	≈64	≈89	+25	+39.1%
Recovery Time (d)	≈ 7 d	≈ 3 d	−4 d	−57.1%
Holding Cost ($)	≈ $11k	≈ $16k	+$5k	—
Stockout Cost ($)	≈ $38k	≈ $12k	−$26k	−68.4%
Total SC Cost ($)	≈ $57k	≈ $37k	−$20k	−35.1%

Metric	Rule	Agentic	Δ	Gain (%)
Stockout Days	21 d	6 d	−15 d	−71.4%

5.4.Per-Agent Performance Breakdown

Table 8 details operational evaluation scores across individual agents.

Table 8. Per-Agent Operational Evaluation Summary

Agent	Score	Grade	Pass%	Fail%	Alert%
Supplier	≈91.0	A	≈94.0%	≈1.0%	≈5.0%
Warehouse	≈83.0	B	≈79.0%	≈8.0%	≈13.0%
Logistics	≈88.0	B	≈89.0%	≈4.0%	≈7.0%
Demand	≈80.0	B	≈71.0%	≈13.0%	≈16.0%
System	≈85.0	B	—	—	—

5.5. Component Ablation Analysis

To quantify the isolated contribution of each architectural component, an ablation study was conducted where individual modules were systematically disabled as shown in Table 9.

Table 9. Architectural Component Ablation Study

Configuration	Fill (%)	Cost ($k)	ΔFill
Full Agentic System	≈95.0%	≈$37.0k	Baseline
No Episodic Memory	≈92.0%	≈$42.0k	−3.0 pp
No Intelligence Module	≈93.0%	≈$39.0k	−2.0 pp
No LangGraph Routing	≈89.0%	≈$47.0k	−6.0 pp
No Message Bus	≈91.0%	≈$44.0k	−4.0 pp
Rule-Based Baseline	≈86.0%	≈$57.0k	−9.0 pp

5.6. Workflow Routing Path Distribution

Table 10 displays the distribution of simulation days across operational workflow paths.

Table 10. Workflow Routing Path Distribution

Execution Path	Rule-Based	Agentic
NORMAL Path	≈40.0%	≈60.0%
EMERGENCY Path	≈37.0%	≈30.0%
CRISIS Path	≈23.0%	≈10.0%

5.7. Decision Traceability Metrics

Table 11 presents auditability metrics compiled from the Decision Recordlog.

Metric Parameter	Value
Total Decisions Logged	≈480 records
LLM-Generated Ratio	73.0% − 82.0%
Rule Fallback Ratio	18.0% − 27.0%
Records with Valid why. Summary	≈96.2%
Mean Confidence (LLM)	≈0.81
Fixed Confidence (Fallback)	0.40
Mean Daily Inference Latency	2.0 − 8.0 seconds

5.8. Case-Study: anomaly of a Crisis Decision State

To illustrate the operational traceability of the agentic framework during extreme events, consider Day 22 of the stress simulation, where a simultaneous supplier outage and logistics shutdown occurred. The Warehouse Agent observed I_{22} = 140 units (below θ = 200), while the Demand Agent projected an upcoming demand of D_{23} = 220 units. LangGraph routed execution directly to the CRISISpath. The Warehouse Agent's generated Decision Record contained: *"Projected stock on Day 24", "Order 500u: Rejected due to warehouse storage limit constraint"*.

- action:REORDEREXPEDITED,Q_{22}= 380 units.
- confidence: 0.88.
- why summary: "Stock level (140u) is insufficient for forecasted demand surge (220u)amidst active 3-day supplier outage. Placing maxi-mum allowable expedited batch (380u) to pre-vent stock out on Day 23."
- contributing _ factors: ["Current stock 140u below theta 200u", "Weather alert indicates 2-day delivery delay", "Episodic memory episode #14 showed stock out under identical condition"].
- alternatives considered: ["Standard order of 200u: Rejected due to projected negative stock on Day 24", "Order 500u: Rejected due to warehouse storage limit constraint"].

This trace proves that the agent's decision was in-formed by factual memory, risk intelligence, and feasibility bounds, providing actionable clarity for post-incident human review.

6.0 COMPARATIVEANALYSIS

6.1 Systematic Architectural Comparison

Table 12 contrasts the proposed framework against traditional rule-based enterprise inventory mechanisms across core functional dimensions.

Table 12. Dimension-by-Dimension Comparison

Dimension	Rule-Based	Proposed Agentic
Disruption Adaptability	Reactive (Post-Drop)	Proactive (Pre-Buffer)
Order Fill Rate	≈86.0%	≈95.0% (+9.0pp)
Stockout Duration	21 days	6 days (−15d)
Total Cost	≈$57,000	≈$37,000 (−35.1%)
Audit Trail	None (Black-Box)	≈96.2% Traces

Dimension	Rule-Based	Proposed Agentic
Demand Sensing	Historical Sales	LSTM + Multi-Intel
Historical Learning	None (Static)	ChromaDB Cosine
Corporate Governance	None	HITL Gate (Q > 0)
Latency	<1.0s day	2.0 – 8.0 s day
Fault Tolerance	System Halt	Fallback Slot-Filling

6.2. Demand Forecasting Paradigms Survey

Table13 compares candidate forecasting methodologies evaluated during system development

Technique	Trend	Season	Disrupt	MAPE (%)
Static EOQ	No	No	No	≈40.0%
7–Day MA	No	Partial	No	15.0% – 22.0%
ARIMA	Yes	Yes	No	12.0% – 18.0%
Prophet	Yes	Yes	Yes	10.0% – 15.0%
LSTM (Used)	Yes	Yes	Multiplier	≈11.0%

6.3. Agent Memory Architecture Comparison

Table 14 compares episodic memory strategies for autonomous supply chain agents.

Table14. Agent Memory Architecture Comparison

Approach	Retrieval	Learning	Complexity
No Memory	None	None	O(1)
Prompt History	Sliding Window	None	O(N)
In-Memory List	Linear Matching	Minimal	O(N)
ChromaDB(Used)	Cosine Dense	Episodic	O(klogN)
Weight Tuning	Implicit Gradient	Parametric	O(1)

7.0 RESEARCH CHALLENGES AND COUNTER MEASURES

7.1 *LLM API Rate Limits and Latency Overhead*

Commercial LLM APIs enforce rate limits (e.g., Groq free-tier ≈500k tokens/day). A 30-day multi-agent simulation consumes approximately 60k tokens. While tractable for single runs, multi-seed statistical suites ($N \geq 30$) quickly breach API quotas.

Counter-measures include:

i. **Semantic Redis Caching:** Caching prompt embeddings with cosine similarity threshold $\delta = 0.95$ eliminates redundant LLM calls during steady-state NORMAL periods.

ii. **Tiered Model Routing:** Small, fast models (e.g., LLaMA 3.1 8B) handle NORMAL operations, reserving larger reasoning models (e.g., LLaMA3.1 70B) exclusively for CRISIS nodes.

7.2. Distributional Drift in Neural Forecasting

Demand spikes (+80%) during severe disruptions deviate significantly from historical training distributions, increasing standalone LSTM MAPE above 25%. Countermeasures include:

i. **Rolling MAPE Anomaly Trigger:** When rolling 7-day forecast error exceeds 25%, the system triggers automated fine-tuning using the most recent 90-day window.

ii. **External Alpha Adjustment:** External intelligence multipliers (α) scale the base LSTM prediction prior to inventory

calculations:

$$y_{t+1} = \alpha \cdot \hat{y}_{t+1} \quad (20)$$

7.3 Semantic Memory Degradation in Long Horizons

As vector memory grows, stale historical episodes from obsolete supply contracts may be retrieved due to surface semantic similarity. Counter measures include a rolling 90-day window and outcome-weighted scoring:

$$\text{Score}(e) = \cos(\mathbf{e}(s_t), \mathbf{e}(e)) \times 1.0 + 0.3 \cdot \mathbf{1}_{\{e.\text{success}\}} \quad (21)$$

7.4 Single-Run Statistical Scope

Current results derive from a representative 30-day stress simulation. The immediate research roadmap involves executing a full Monte Carlo evaluation suite ($N \geq 30$ trials with varied pseudo random seeds) to Report mean $\pm\sigma$ for all KPI distributions along side shadow-mode ERP validation.

7.4 *Single-SKU, Single-Node Boundary*

The current prototype models a single SKU at a single retail location. Future extensions will scale the state space to K multi-category SKUs governed by linear Warehouse capacity constraint $s_k \; Q_k \leq C_{\text{warehouse}}$, and incorporate regional distribution hub agents supporting lateral transshipment.

8.0 CONCLUSIONANDFUTUREWORK

8.1 *Conclusion*

This paper presented an integrated five-layer Agentic AI Framework for retail supply chain decision support, combining LLM ReAct reasoning, LSTM demand forecasting, ChromaDB episodic memory, LangGraph state orchestration, external intelligence synthesis, and inline DecisionRecord traceability. Simulation under severe compound disruptions demonstrates that the agentic system improves order fill rate by 9 percentage points (86%to95%), reduces stock out occurrences by 71%, and lowers total supply chain cost by approximately 35% compared to classical rule-based baselines. Ablation analysis confirms that LangGraph state routing and inter-agent Message Bus coordination provide the primary structural foundation for supply chain resilience.

8.2 *Future Work*

Future research focuses on: (1) multi-seed statistical evaluation ($N \geq 30$), (2) Reinforcement Learning policy optimization using PPO [33] on reward $r_t = \alpha\, \text{FR}_t - \beta\, C_{\text{hold},t} - \gamma\, C_{\text{stockout},t}$, and (3) multi-echelon network scaling.

REFERENCES

[1]. F.W.Harris, "How many parts to make at once," *Factory, The Magazine of Management*,vol.10, no.2,pp.135–136,1913.

[2]. E. A. Silver, D. F. Pyke, and D. J. Thomas, *Inventory and Production Management in Supply Chains*, 4th ed., Boca Raton, FL: CRC Press, 2017.

[3]. X. Brusset and C. Teller, "Supply chain capabilities, risks, and resilience," *International Journal of Production Economics*, vol. 184, pp. 59–68, 2017.

[4]. A. A. Syntetos, K. Babai, J. E. Boylan, S. Kolassa, and K.Nikolopoulos, "Supply chain forecasting: Theory, practice, their gap and the future," *European Journal of Operational Re-search*, vol. 252, no. 1, pp. 1–26, 2016.

[5]. C. W. Craighead, J.Blackhurst, M. J. Rungtu-sanatham, and J.W.Handfield,"The severityof supply chain disruptions: Design characteristics and mitigation capabilities, "*Decision Sciences*, vol. 38, no. 1, pp. 131–156, 2007.

[6]. H.L.Lee, V.Padmanabhan, and S.Whang,"In-formationdistortioninasupplychain:Thebull-whipeffect,"*ManagementScience*,vol.43,no. 4, pp. 546–558, 1997.

[7]. D. Masadand J.Kazil,"Mesa: An agent-based modeling framework," in *Proc. 14th Python in Science Conf. (SciPy)*, Austin, TX, 2015, pp. 51–58.

[8]. J. M. Swaminathan, S. F. Smith, and N. M. Sadeh, "Modeling supply chain dynamics: A multiagent approach," *Decision Sciences*, vol. 29, no. 3, pp. 607–632, 1998.

[9]. M.W.Macy and R.Willer,"From factors to actors: Computational sociology and agent-based odeling,"*AnnualReviewofSociology*,vol.28, no. 1, pp. 143–166, 2002.

[10]. L. Wang et al., "A survey on large language model based autonomous agents," *Frontiers of Computer Science*, vol.18 ,no.6, p.186345, 2024.

[11]. J.S.Parketal.,"Generative agents: Interactive simulacra of human behavior," in *Proc. 36th Annu. ACM Symp. User Interface Software and Technology (UIST)*, 2023, pp. 1–22.

[12]. S. Yao, J. Zhao, D. Yu, N. Du, I. Shafran, K. Narasimhan, and Y. Cao, "ReAct: Synergizing reasoning and acting in language models," in *Proc. Int. Conf. Learning Representations (ICLR)*, 2023.

[13]. Z. Xi et al., "The rise and potential of large language model based agents: A survey," *arXiv preprint arXiv:2309.07864*, 2023.

[14]. Z.Shen, Y.Liu ,and X.Chen, "Large language models for supply chain demand sensing," *arXiv preprint arXiv:2402.07023*, 2024.

[15]. E. Karpas et al., "MRKL Systems: A modular, neuro-symbolic architecture that combines large language models, external knowledge sources and discrete reasoning," arXiv preprint *arXiv:2205.00445*, 2022.

[16]. H. Chase, "LangChain: Building applications withLLMsthroughcomposability,"2022. [On-line]. Available: https://github.com/langchain-ai/langchain

[17]. LangChain AI, "LangGraph: Multi-agent orchestration framework,"2024. [Online]. Avail-able: https://github.com/langchain-ai/langgraph

[18]. P. Lewis et al., "Retrieval-augmented generation for knowledge-intensive NLP tasks, "in *Advances in Neural Information Processing Systems (NeurIPS)*, vol.33, 2020, pp.9459–9474.

[19]. A.Trott and J.Chilton, "Chroma: Open-source embedding database for AI applications,"2023. [Online]. Available: https://github.com/chroma-core/chroma

[20]. N.Reimers and I.Gurevych, "Sentence-BERT: Sentence embeddings using Siamese BERT-networks, "in *Proc. Conf. Empirical Methods in Natural Language Processing (EMNLP)*, 2019, pp.3982–3992.

[21]. S.HochreiterandJ. Schmidhuber, "Long short-term memory," Neural *Computation*, vol.9,no. 8, pp. 1735–1780, 1997.

[22]. S. J. Taylor and B. Letham, "Forecasting at scale, "*The American Statistician*,vol.72 ,no.1, pp.37–45,2018.

[23]. R.Fildes, P.Goodwin, and M.Lawrence, "The design features of forecasting support systems and their effectiveness, "*Decision Support Systems*, vol. 42, no. 1, pp. 351–361, 2006.

[24]. S. Makridakis, E. Spiliotis, and V. Assi-mako poulos, "The M5 accuracy competition: Results, findings, and conclusions," *International Journal of Forecasting*, vol. 38, no. 4, pp.1346–1364,2022.

[25]. G. Hadley and T. M. Whitin, Analysis of Inven-tory Systems, Englewood Cliffs, NJ: Prentice-Hall, 1963.

[26]. D. Simchi-Levi, P. Kaminsky, and E. Simchi-Levi, Designing and Managing the Supply Chain: Concepts, Strategies and Case Studies, 3rd ed., New York: McGraw-Hill, 2008.

[27]. S. Tisue and U. Wilensky, "NetLogo: A simple environment for modeling complexity," in Proc. Int. Conf. on Complex Systems, Boston, MA, 2004, pp. 16–21.

[28]. M. T. Ribeiro, S. Singh, and C. Guestrin, "'Why should I trust you?': Explaining the predictions of any classifier," in Proc. 22nd ACM SIGKDD Int. Conf. Knowledge Discovery and Data Mining, 2016, pp. 1135–1144.

[29]. S. M. Lundberg and S.-I. Lee, "A unified approach to interpreting model predictions," in Advances in Neural Information Processing Systems (NeurIPS), vol. 30, 2017, pp. 4765–4774.

[30]. A. B. Arrieta et al., "Explainable artificial intelligence (XAI): Concepts, taxonomies, opportunities and challenges toward responsible AI," Information Fusion, vol. 58, pp. 82–115, 2020.

[31]. D. Gunning et al., "XAI—Explainable artificial intelligence," Science Robotics, vol. 4, no. 37, p. eaay7120, 2019.

[32]. R. Y. Zhong, X. Xu, E. Klotz, and S. T. New-man, "Intelligent manufacturing in the context of industry 4.0: A review," Engineering, vol. 3, no. 5, pp. 616–630, 2017.

[33]. J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov, "Proximal policy optimization algorithms," arXiv preprint arXiv:1707.06347, 2017.