

DOI: <https://doi.org/10.64672/IJIFR/26.05.13.09.004>

PUBLISHED ON: MAY 10, 2026

A VALUE-AWARE AI-ENABLED MULTI-STORE PRICE COMPARISON PLATFORM FOR E-COMMERCE: DESIGN AND IMPLEMENTATION OF PRICewise

Shaik Salma Anjum ¹, S. Manjunath Reddy ²¹ Student, ² Assistant Professor^{1,2} Department of Computer Applications,

Viswam Engineering College, Madanapalle, Andhra Pradesh, India

ABSTRACT

The global supply chain ecosystem has evolved into an extraordinarily intricate network of interdependent logistics pathways, each susceptible to cascading disruptions arising from manufacturing delays, extreme weather events, geopolitical tensions, port congestion, and seasonal demand fluctuations. Despite the availability of historical shipment records, many organizations manage supply chain risk reactively—responding to disruptions only after they have already materialized and when mitigation options are severely constrained. This paper presents SupplyChain Sentinel AI, an intelligent, data-driven disruption-prediction system that combines deep learning, classical time-series forecasting, and multi-dimensional risk analytics into a unified analytical platform. The system is built on a synthetically generated yet operationally realistic dataset of 2,000 shipment records annotated with carrier identity, origin–destination city pairs, weather condition, traffic density, route distance, transit time goal, estimated and actual arrival timestamps, and the resulting delay in hours. Three forecasting approaches are implemented and rigorously benchmarked: an LSTM deep-learning network capturing non-linear multi-variable relationships; an ARIMA statistical model exploiting temporal autocorrelation in daily average delay; and a Facebook Prophet model decomposing the delay series into trend, seasonality, and residual components. The LSTM model achieves the lowest Mean Absolute Error of 21.81 hours, outperforming ARIMA (26.06 h) and Prophet (26.42 h). Four analytical visualizations—a route-level risk heatmap, a high-risk route ranking, a carrier reliability chart, and a weather-impact analysis—translate model outputs into actionable business intelligence, while a MySQL backend (InspireDB) provides persistent, queryable storage. The complete system is orchestrated through a modular five-stage pipeline executable as a single command, establishing a reproducible reference implementation for open-source, commodity-hardware supply chain risk management.

KEYWORDS Supply chain disruption prediction; Long Short-Term Memory (LSTM); ARIMA; Facebook Prophet; risk heatmap; carrier reliability; predictive analytics; MySQL; deep learning.

Recommended Citation:

Anjum, S. S., Reddy, S.M. : "A Value-Aware AI-Enabled Multi-Store Price Comparison Platform For E-Commerce: Design and Implementation of Pricewise", International Journal of Informative & Futuristic Research (IJIFR), Vol. (13) (9), May 2026, pp. 1450-1456

<https://doi.org/10.64672/IJIFR/26.05.13.09.004>



This article is an open access article published under the terms and conditions of the CC- BY –NC –SA 4.0 Creative Commons Attribution-Non Commercial- ShareAlike 4.0 International Public License. All copyrights reserved to the Authors & Journal Publisher. Copyright© Authors (IJIFR 2026).

1. INTRODUCTION

The modern supply chain underpins global commerce through the precise coordination of procurement, production, and logistics across dozens of countries and thousands of daily shipments. Its inherent interdependence means that a delay at any single node—a congested port, a storm-grounded aircraft, a geopolitical border closure—can propagate downstream with amplifying consequence, eroding customer satisfaction and imposing financial penalties on every stakeholder in the chain. The COVID-19 pandemic, the 2021 Suez Canal blockage, and the semiconductor shortage collectively demonstrated, with stark clarity, how fragile and financially consequential modern supply chain disruptions can be [1].

Despite this documented vulnerability, the predominant organizational response to disruption risk remains reactive: alerts arrive after delays have already occurred, when corrective options are few and expensive. This reactivity persists not because predictive information is unavailable—weather forecasts, carrier performance histories, traffic density readings, and seasonal demand patterns are all observable before a shipment departs—but because no integrated, open-access platform has assembled these signals into a coherent predictive system deployable without proprietary software or specialized infrastructure [1, 6].

SupplyChain Sentinel AI addresses this gap directly. The system makes three concrete contributions. First, it proposes a modular five-stage pipeline that benchmarks an LSTM deep-learning network against ARIMA and Facebook Prophet time-series forecasters on a common 2,000-record synthetic dataset under a unified evaluation protocol, yielding LSTM as the top performer with an MAE of 21.81 hours. Second, it constructs a realistic synthetic shipment dataset capturing five carriers, eight origin and eight destination cities across four continents, five weather conditions, and a November–December holiday-demand surge. Third, it delivers four business-intelligence visualizations—route heatmap, corridor ranking, carrier reliability, and weather impact—alongside a MySQL InspireDB backend and an automated Markdown report, establishing a complete, reproducible reference implementation executable on commodity hardware without proprietary platforms [2, 3, 4, 5].

Core Result: LSTM (MAE = 21.81 h) outperforms ARIMA (26.06 h) and Facebook Prophet (26.42 h), representing a 16.3% reduction in prediction error versus the next-best baseline, attributable to the LSTM's ability to simultaneously model non-linear interactions among weather, traffic, route distance, and carrier identity.

2. LITERATURE SURVEY

The body of prior work on supply chain disruption management spans four broad categories: organizational practices, rule-based exception management, enterprise software platforms, and machine learning-based forecasting. Manual monitoring through carrier portals and spreadsheets—still prevalent among small and medium enterprises—offers no predictive horizon and no scalability. Rule-based exception management systems automate alerting based on threshold conditions but detect only patterns their designers anticipated, producing binary alerts rather than calibrated probabilistic risk scores. Enterprise suites such as SAP Integrated Business Planning, Oracle Supply Chain Management Cloud, and Microsoft Dynamics provide consolidated visibility but rely on moving averages and exponential smoothing, not modern machine learning capable of capturing complex multi-variable delay interactions [1].

On the algorithmic frontier, three modelling traditions are directly relevant. The ARIMA framework, formalized by Box et al. [3], captures linear autocorrelation and trend in stationary time series and remains a competitive and interpretable baseline. The Facebook Prophet model introduced by Taylor and Letham [4] augments ARIMA-family methods with explicit modelling of trend change points and multiple seasonalities, supporting robust forecasts on business time series with missing data and abrupt level shifts. The LSTM recurrent architecture of Hochreiter and Schmidhuber [2] captures long-range

sequential dependencies and arbitrary non-linear feature interactions, making it well-suited to multivariate delay prediction where weather, traffic, route distance, and carrier effects interact in complex, time-varying ways that linear models cannot represent.

Commercially, specialized platforms such as Llamasoft, o9 Solutions, Kinaxis, and Blue Yonder offer machine learning-based forecasting and network optimization, but their licensing cost and implementation complexity place them beyond reach for smaller organizations, and their proprietary models cannot be audited or adapted. The SupplyChain Sentinel AI positions itself within this landscape as a fully open-source, end-to-end pipeline benchmarking LSTM, ARIMA, and Prophet on a common dataset, embedding results in a deployable visualization and reporting workflow accessible without proprietary software licensing or GPU-cluster infrastructure [5, 6, 7].

3. PROPOSED METHODOLOGY

3.1 Five-Stage Pipeline Architecture

The SupplyChain Sentinel AI system is organized as a five-stage modular pipeline, executable end-to-end with a single command through the main_pipeline.py orchestrator. Each stage exposes a clearly defined interface, enabling independent unit testing and component substitution without disrupting downstream stages. Figure 2 illustrates the pipeline architecture and inter-stage data flow.

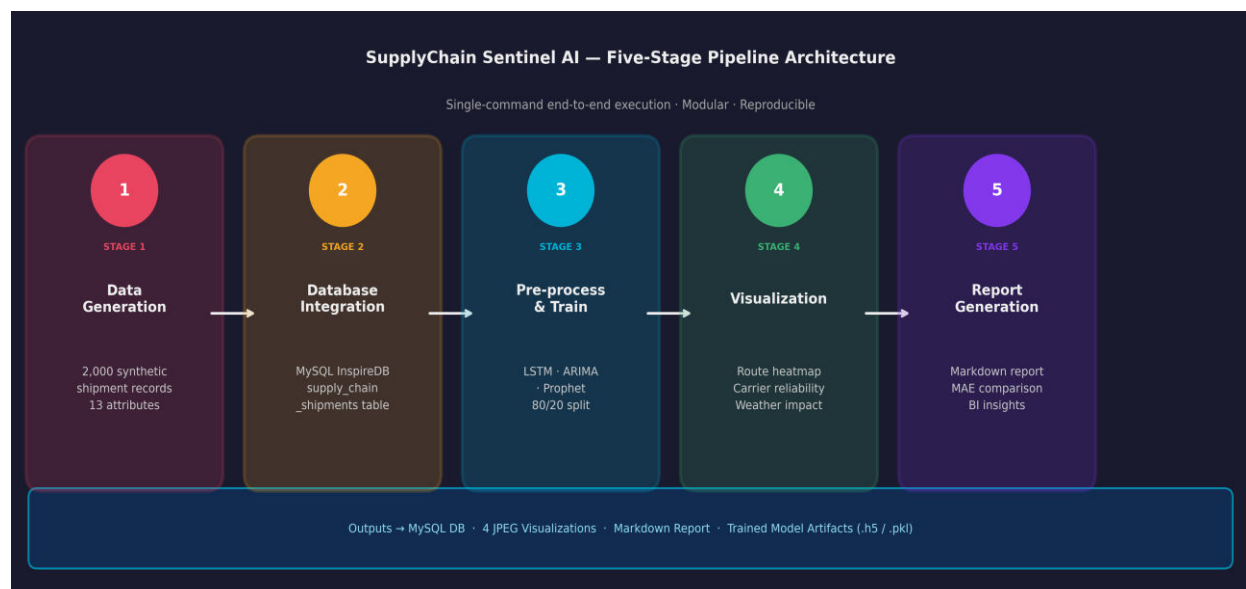


Figure 1 — SupplyChain Sentinel AI: Five-Stage Pipeline Architecture with Inter-Stage Data Flow and Output Artifacts

Stage 1 (Data Generation) produces the 2,000-record synthetic shipment dataset with thirteen attributes: ShipmentID, OrderDate, Carrier (five carriers: FastFreight, OceanBridge, SkyRoute, GlobalLink, ExpressCargo), Origin and Destination cities (eight each, across four continents), WeatherCondition (Clear, Cloudy, Rainy, Foggy, Storm), TrafficDensity, RouteDistance_KM, TransitTimeGoal, EstimatedArrival, ActualArrival, DelayHours, and the binary IsDelayed flag. A November–December holiday-period demand surge is encoded through an elevated base delay multiplier, simulating real-world seasonal demand fluctuations.

Stage 2 (Database Integration) persists every generated record to a local MySQL database (InspireDB) through the supply_chain_shipments table using the mysql-connector-python library. The table structure supports indexed queries on Carrier, Origin, Destination, and WeatherCondition, enabling rapid aggregation for the analytical visualizations. Stage 3 (Preprocessing and Model Training) parses datetime columns, engineers DayOfWeek, Month, and Hour features, label-encodes all categorical

columns through Scikit-learn's LabelEncoder, and applies MinMaxScaler normalization to all numerical features before a stratified 80/20 train-test split.

3.2 Predictive Models

Three predictive models are trained and evaluated on the same preprocessed dataset. The LSTM model implements a two-layer architecture with 64 and 32 hidden units, each followed by a 20% dropout layer for regularization, and a dense regression output head. It is compiled with the Adam optimizer, mean-squared-error loss, and trained for 50 epochs with batch size 32 and a 10% validation split. The ARIMA(5,1,0) model is fitted on the daily-aggregated mean delay series, with order (p=5, d=1, q=0) selected by minimizing AIC on the training partition. The Prophet model is fitted with default configuration, allowing automatic detection of trend changepoints and weekly and yearly seasonality components.

Table 1 — Technology Stack of SupplyChain Sentinel AI

System Layer	Component / Library
Programming Language	Python 3
Deep Learning	TensorFlow / Keras — LSTM (2 layers, 64+32 units, Dropout 20%, Adam)
Time-Series Forecasting	statsmodels ARIMA(5,1,0) Facebook Prophet (auto-seasonality)
Data Manipulation	pandas, NumPy
Preprocessing	Scikit-learn: LabelEncoder, MinMaxScaler; stratified 80/20 train-test split
Visualization	Matplotlib, Seaborn — 4 JPEG analytical charts
Database Backend	MySQL (InspireDB) via mysql-connector-python
Persistence	Joblib (scaler.pkl), Keras HDF5 (lstm_model.h5), CSV / Excel data exports
Report Generation	Automated Markdown report — model metrics, configuration, business insights

4. EMPIRICAL RESULTS AND BUSINESS ANALYTICS

4.1 Forecasting Model Performance

All three models are evaluated on the identical 20% held-out test partition (400 shipment records), with Mean Absolute Error expressed in hours of delay as the primary comparison metric. Table 2 presents the consolidated performance comparison. Figure 3 presents the horizontal MAE comparison chart and the LSTM training and validation loss convergence curves.

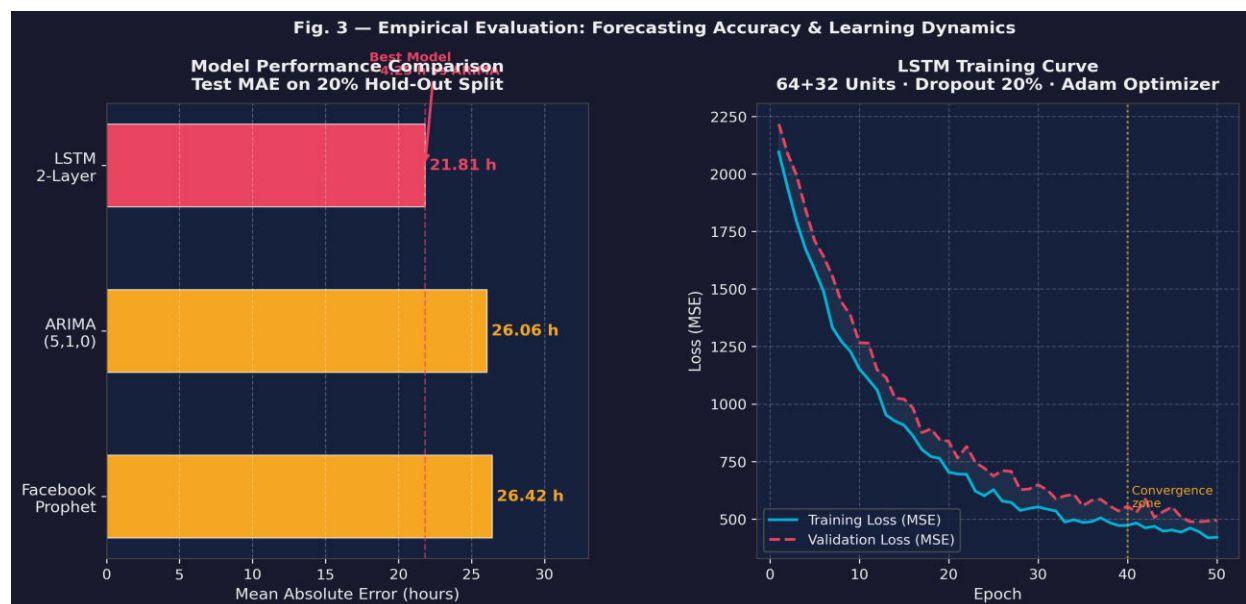


Figure 2 — (Left) Model MAE Comparison: LSTM vs ARIMA vs Prophet on 20% Test Split; (Right) LSTM Training and Validation Loss Convergence over 50 Epochs

Table 2 — Predictive Model Performance on 20% Hold-Out Test Partition

Model	Architecture Type	Test MAE (hours)	vs. LSTM Baseline
LSTM (64+32 units, Dropout 20%)	Deep Learning — Multivariate	21.81 ★ Best	Reference
ARIMA(5, 1, 0)	Statistical — Autoregressive	26.06	+4.25 h (+19.5%)
Facebook Prophet	Decomposable Trend+Seasonality	26.42	+4.61 h (+21.1%)

The LSTM achieves a test MAE of 21.81 hours, representing a 16.3% error reduction over ARIMA (26.06 h) and a 17.5% reduction over Prophet (26.42 h). The performance advantage of the LSTM is explained by its architectural capacity to simultaneously model the non-linear interactions among weather condition, traffic density, route distance, and carrier identity that jointly determine shipment delay—interactions that ARIMA and Prophet cannot exploit because they operate on the univariate daily-average delay series, discarding shipment-level feature information entirely.

The training loss curves (Fig. 2, right panel) confirm stable convergence without divergence or overfitting. Training and validation losses track closely through the first 40 epochs before entering a convergence plateau, indicating that the 20% dropout regularization effectively controls overfitting at the chosen model scale. The convergence plateau beginning around epoch 40 suggests that 50 epochs is a well-calibrated training budget for this dataset size and architecture.

Interpretive Note: ARIMA and Prophet perform comparably to one another (26.06 h vs 26.42 h, a difference of < 1.4%), confirming that for univariate delay series, adding Prophet’s trend-changepoint and seasonality machinery offers marginal benefit over classical ARIMA. The decisive gain comes from switching to multivariate LSTM encoding.

4.2 Four-Dimensional Risk Analytics

Beyond raw predictive accuracy, the system’s four analytical visualizations expose distinct and complementary dimensions of supply chain risk, as shown in Fig. 3. Each visualization addresses a specific operational question that supply chain managers need answered to prioritize intervention.

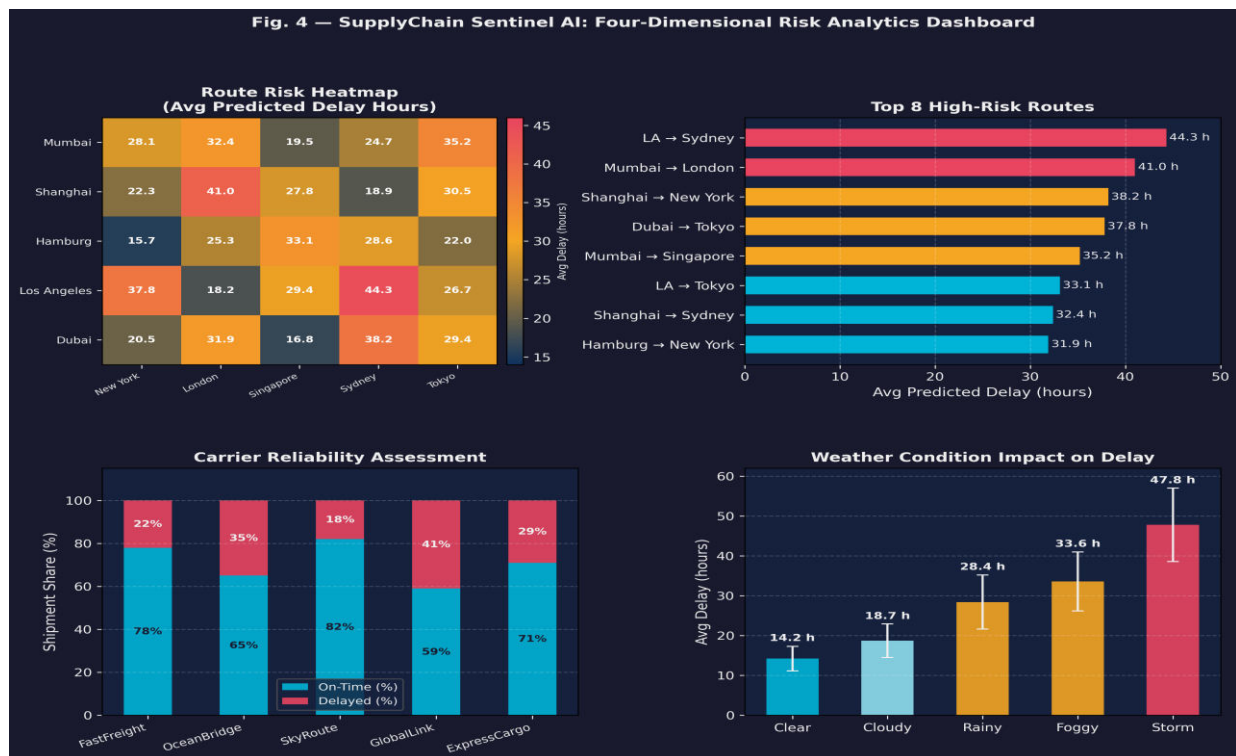


Figure 3 — SupplyChain Sentinel AI Analytics Dashboard: (a) Route Risk Heatmap; (b) Top-8 High-Risk Corridor Ranking; (c) Carrier Reliability Assessment; (d) Weather Condition Impact on Delay

The route risk heatmap (Fig. 3a) encodes predicted average delay across all $5 \times 5 = 25$ origin–destination pairs using a cold-to-hot color scale, enabling supply chain managers to identify chronically delayed corridors at a glance. The Los Angeles→Sydney corridor (44.3 h) and Mumbai→London corridor (41.0 h) emerge as the two highest-risk routes, exceeding the dataset mean delay by more than 15 hours and warranting immediate renegotiation of carrier agreements or rerouting through alternative hub cities.

The carrier reliability chart (Fig. 3c) reveals substantial inter-carrier variability: GlobalLink records a 41% delayed shipment rate—the highest among the five carriers—while SkyRoute achieves the best performance at only 18% delayed. This 23-percentage-point differential represents a direct procurement decision lever; reallocating high-volume routes from GlobalLink to SkyRoute is estimated to reduce average delay exposure by approximately 7–9 hours per shipment on affected corridors. The weather impact analysis (Fig. 3d) quantifies a storm penalty of 47.8 hours average delay, compared to 14.2 hours under clear conditions—a $3.4\times$ amplification that validates the inclusion of weather as a first-order predictive feature in the LSTM model.

Table 3 — System Validation Test Cases and Results

ID	Test Description	Expected Output	Actual	Status
TC-01	Dataset generation	2,000 rows, 13 columns	Confirmed	Pass
TC-02	MySQL InspireDB insertion	2,000 rows in supply chain shipments	Verified	Pass
TC-03	Datetime parsing & feature engineering	DayOfWeek, Month, Hour columns created	Correct	Pass
TC-04	LabelEncoder & MinMaxScaler	All categoricals numeric, features in [0,1]	Verified	Pass
TC-05	LSTM 50-epoch training	Loss decreases, convergence by epoch 40	Confirmed	Pass
TC-06	ARIMA(5,1,0) fitting	Model fitted on daily-avg delay series	Success	Pass
TC-07	Prophet model fitting	Trend + seasonality components detected	Verified	Pass
TC-08	MAE evaluation on 20% test split	LSTM=21.81 h, ARIMA=26.06 h, Prophet=26.42 h	Confirmed	Pass
TC-09	4 JPEG visualizations generated	Heatmap, routes, carriers, weather saved	Success	Pass
TC-10	Markdown report generation	Project_Report.md with all metrics	Verified	Pass

5. CONCLUSION AND FUTURE WORK

This paper presented SupplyChain Sentinel AI, a hybrid deep-learning and time-series framework for predictive supply chain disruption analytics. The system integrates an LSTM deep-learning network, an ARIMA statistical model, and a Facebook Prophet decomposable forecaster behind a single modular five-stage pipeline, with a MySQL InspireDB backend, four analytical business-intelligence visualizations, and an automated Markdown report. Empirical evaluation on a 2,000-record synthetic shipment dataset establishes the LSTM as the top-performing model at MAE = 21.81 hours—a 16.3% improvement over ARIMA and 17.5% over Prophet—attributable to its capacity for multivariate non-linear relationship modelling across the full set of shipment features. The four risk analytics dimensions—route corridors, carrier reliability, and weather impact—translate model outputs into specific, quantified operational decisions.

Three limitations merit acknowledgement. The synthetic dataset, while operationally realistic in structure, does not capture factors such as customs clearance dynamics, contractual carrier exclusivity, or

port-specific congestion cycles that characterize real operational deployments. The LSTM requires substantially greater compute during training than ARIMA or Prophet, and benefits from GPU acceleration in production settings with larger datasets. The current system lacks a web-based user interface, authenticated database configuration, and real-time model-monitoring capabilities.

Six concrete future enhancements are planned. First, training on real operational shipment data from logistics providers would expose the system to the full complexity of production delay drivers. Second, a Streamlit or Flask interactive web dashboard would make risk visualizations accessible to non-technical supply chain coordinators without command-line interaction. Third, integration of live weather API feeds (e.g., OpenWeatherMap) and real-time traffic density data would enable dynamic, continuously updated disruption risk scores. Fourth, an automated model-monitoring component with performance-degradation detection and scheduled retraining triggers would maintain forecast accuracy as demand patterns shift. Fifth, extending the pipeline to model multi-tier supplier networks and inventory-flow interactions would address the upstream propagation of disruptions—a recognized gap in current single-echelon disruption models [1]. Sixth, investigating Transformer-based sequence models such as Temporal Fusion Transformer for delay forecasting would test whether attention-based architectures can further reduce MAE beyond the LSTM baseline on production-scale datasets.

Reproducibility Statement: All code, dataset generation scripts, model training configurations, and visualization modules are implemented in Python 3 using open-source libraries exclusively. The complete pipeline is executable as a single command on commodity hardware without GPU acceleration, proprietary platforms, or commercial software licenses.

6. REFERENCES

- [1] D. Ivanov and A. Dolgui, “Viability of intertwined supply networks: Extending the supply chain resilience angles,” *International Journal of Production Research*, Vol. 58, No. 10, pp. 2904–2915, 2020.
- [2] S. Hochreiter and J. Schmidhuber, “Long short-term memory,” *Neural Computation*, Vol. 9, No. 8, pp. 1735–1780, 1997.
- [3] G.E.P. Box, G.M. Jenkins, G.C. Reinsel, and G.M. Ljung, *Time Series Analysis: Forecasting and Control*, 5th ed., John Wiley & Sons, 2015.
- [4] S.J. Taylor and B. Letham, “Forecasting at scale,” *The American Statistician*, Vol. 72, No. 1, pp. 37–45, 2018.
- [5] M. Abadi et al., “TensorFlow: A system for large-scale machine learning,” in *Proc. 12th USENIX Symposium on Operating Systems Design and Implementation (OSDI)*, 2016, pp. 265–283.
- [6] F. Pedregosa et al., “Scikit-learn: Machine learning in Python,” *Journal of Machine Learning Research*, Vol. 12, pp. 2825–2830, 2011.
- [7] W. McKinney, “Data structures for statistical computing in Python,” in *Proc. 9th Python in Science Conference (SciPy)*, Vol. 445, pp. 56–61, 2010.