Research Article

Commercial Intelligence AI-Based Pellet Price Prediction -Digital Transformation in Mining and Metals

Venugopalam Medicherla[1] and Aasheesh Raizada[2]

[1,2]Department of Computer Engineering & Applications, Mangalayatan University, Aligarh, India
E-mails: [1]20230223_venugopal@mangalayatan.edu.in, [2]aasheesh.raizeda@mangalayatan.edu.in

Abstract

Mining & Metals is undergoing a paradigm shift driven by Industry 4.0 technology, increasingly applies artificial intelligence (AI), machine learning (ML) and large language models (LLMs) to commercial decisions, yet pellet pricing in India's iron and steel chain often still rests on manual judgement and spreadsheets. This study develops an AI-based framework to forecast the daily pellet sales price from SAP S/4HANA records, Steel-Mint quotations, competitor indices and plant costing data of an operating producer. Fourteen control predictors—with Sponge iron price (24.7%), Competitor Index 1 (17.3%), and Coal price (13.5%) carrying the largest linear-model weights—were standardised with training-period statistics. An interpretable Linear Regression explained with SHAP and a 200-tree Random Forest were trained on 1,127 daily records from December 2022 to December 2025 and tested on 284 later records. A locally hosted LLM (Ollama 70B) with retrieval-augmented generation (RAG) converts news and regulatory text into a sentiment signal. Error was measured by MAE, RMSE and MAPE and compared with a persistence benchmark. On 284 held-out daily records (6 January to 17 October 2026), the model reached an MAE of ₹578.8/MT and a MAPE of 5.28% (accuracy, 100 − MAPE: 94.72%); 60.9% of forecasts fell within 5% of the observed price. One-day persistence scored 0.41% MAPE, but its error exceeded the model's beyond a horizon of about 30 days. Errors rose sharply in September and October 2026 due to unexpected market volatility and policy shifts. The study documents an explainable pipeline that combines enterprise data, market feeds, ML models and LLM-derived signals, and shows that its value over simple benchmarks depends on forecast horizon, while simultaneously supporting ESG-aligned operational transparency and revenue optimization. The evidence comes from one producer and one commodity.

Keywords: Digital Transformation, Mining Industry, Artificial Intelligence, Machine Learning, Large Language Models, Pellet Price Forecasting, SAP ERP, Commercial Intelligence.

1. Introduction

1.1 Background and Industry Context

The Mining and Metals Industry serves as a fundamental pillar of global industrial manufacturing, infrastructure development, and economic growth. However, the sector operates in an inherently volatile commercial environment characterized by frequent, non-linear commodity price swings. Unlike traditional manufacturing sectors operating under fixed or slowly adjusting cost-plus pricing models, commodity pricing for processed minerals—specifically iron ore pellets—is driven by a complex interaction of internal operational conversion costs, global supply-demand imbalances, energy price shocks, fluctuating freight dynamics, and shifting macroeconomic policies. Iron ore pellets are a critical raw material in modern direct reduced iron (DRI) and blast furnace steelmaking processes. Accurately forecasting their daily market sales price is a vital strategic capability for mining and metallurgical enterprises. Even minor pricing inaccuracies lead to substantial financial friction, causing misprsiced supply contracts, eroded profit margins, inefficient inventory holding, and unhedged exposure to market downturns.

In practice, particularly across India’s steel and iron manufacturing ecosystem, pellet pricing decisions frequently rely on conventional business intelligence methods, such as legacy computer systems, manual spreadsheet calculations, static linear extrapolations, and executive intuition. While these static heuristics were sufficient during periods of low market volatility, they lack the dynamic reactivity required to navigate hyper-connected, rapidly fluctuating global commodity markets.

1.2 The Role of Industry 4.0 and Enterprise Digital Transformation

To mitigate market uncertainty, the global mining sector is undergoing a structural paradigm shift driven by Industry 4.0 technologies. Digital transformation frameworks increasingly integrate Internet of Things (IoT) sensor networks, cloud computing, digital twins, Machine Learning (ML), and Artificial Intelligence (AI) directly into core enterprise operations. Enterprise Resource Planning (ERP) systems, specifically modern architectures like SAP S/4HANA, act as the foundational digital backbone of this evolution. SAP S/4HANA functions as the authoritative single source of truth for internal enterprise operations, systematically logging transactional sales history, plant conversion costs, raw material landing costs, finance overheads, and real-time inventory

balances. However, a significant operational and academic gap remains: the vast majority of Industry 4.0 implementations in mining focus almost exclusively on physical operational processes—such as predictive equipment maintenance, autonomous haul fleet routing, crushing plant optimization, and ore grade estimation. While operational AI applications reduce costs, they leave the enterprise vulnerable to top-line revenue erosion if commercial sales pricing remains reactive and unoptimized.

1.3 Conceptual Framework and System Architecture

To bridge the disconnect between enterprise transactional backbones and dynamic commercial decision-making, this study establishes an integrated four-layer digital transformation framework. The system architecture seamlessly links internal ERP accounting records with live external commodity market feeds and qualitative intelligence streams.

Layered Architecture Breakdown:

Layer – 1 Data Acquisition Layer: Ingests historical transactional sales records and costing data directly from SAP S/4HANA alongside live external market feeds from Steel-Mint (e.g., coal, sponge iron, scrap prices) and qualitative news updates.

Layer – 2 Data Processing and Feature Engineering Layer: Standardizes input variables using training-set statistics, handles missing values, detects outliers, and structures time-series lag features stored within persistent MySQL tables.

Layer – 3 Predictive Modelling Layer: Combines an interpretable Linear Regression baseline, a 200-tree Random Forest ensemble, and an on-premises Large Language Model (Ollama 70B) for unstructured news sentiment extraction, explained using SHAP values.

Layer – 4 Commercial Decision Layer: Surfaces real-time price predictions, confidence margins, and scenario simulation tools through a web-based Flask analytics dashboard to support sales contracting and executive strategy.

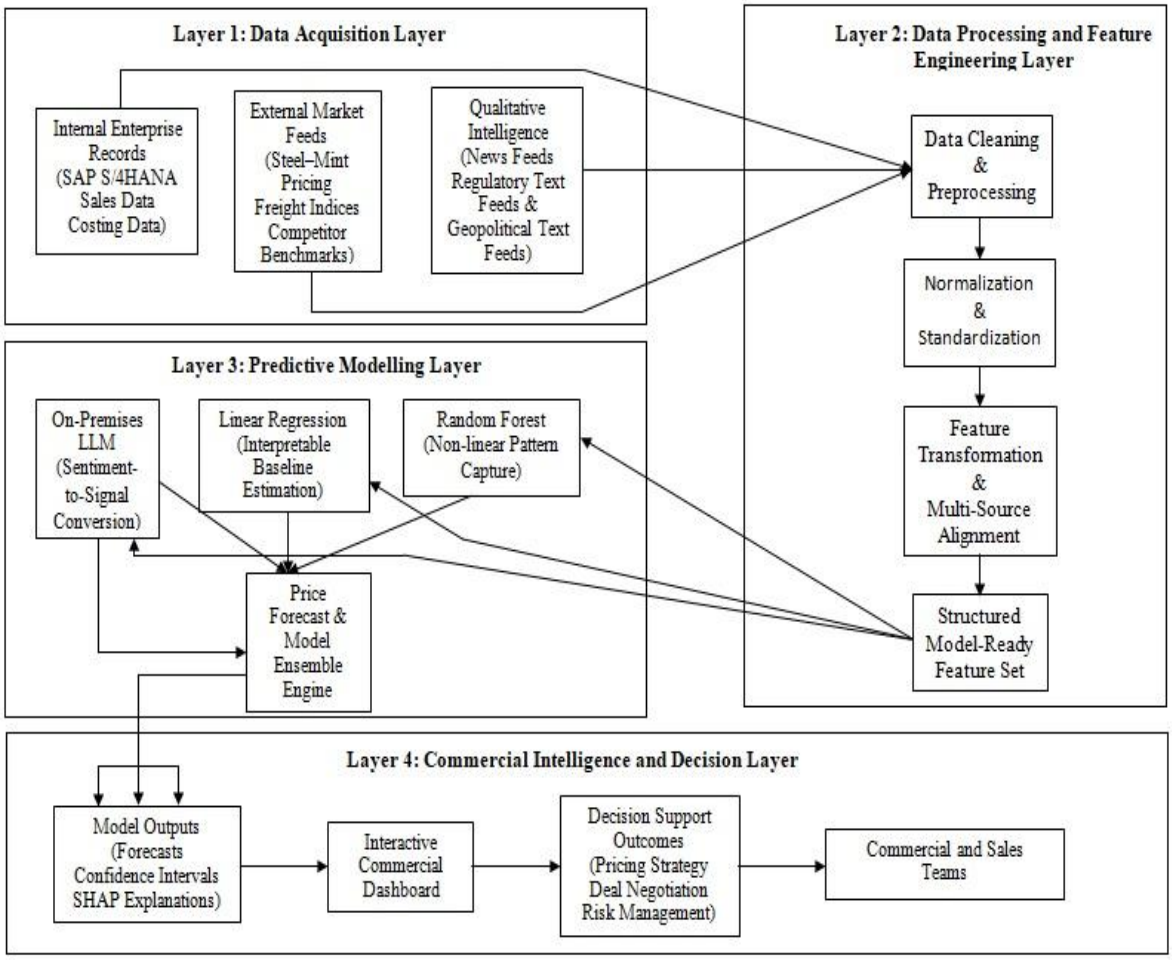


Figure – 1: Four-Layer Digital Transformation Framework for AI-Driven Commercial Intelligence in Mining

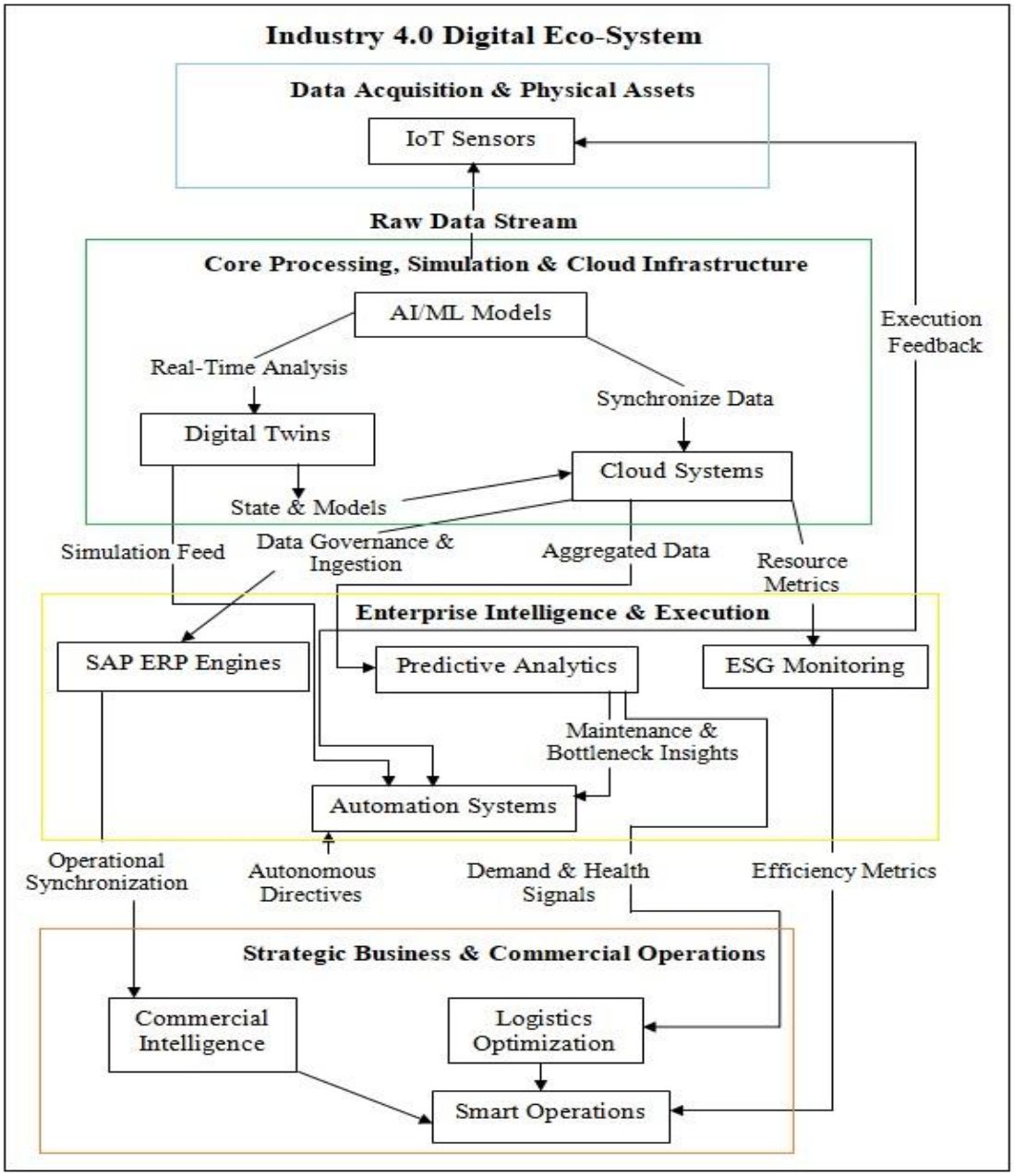
Despite the rapid adoption of predictive machine learning in quantitative finance, commercial price forecasting within the mining and metals domain faces critical challenges like Internal operational cost structures stored in ERP systems (e.g., plant conversion costs, opening stock, order books) are rarely combined with live external market indexes in automated predictive pipelines with inability to Process Qualitative Market Signals in which Traditional numerical time-series algorithms are blind to non-numeric disruptions, such as sudden policy changes, export duty revisions, environmental regulations, and geopolitical events. Additionally, High-performing machine learning models often operate as "black boxes" lacking explainability, while simple persistence baselines ("today's price = tomorrow's price") are rarely benchmarked to identify the exact forecast horizon where AI model utility supersedes naive guessing.
This paper directly addresses these challenges by developing, implementing, and validating an enterprise-embedded AI commercial intelligence system for an operating iron ore pellet producer.

The explicit purpose of this study is:
1. To design and deploy four-layer Industry 4.0 digital transformation architecture bridging SAP S/4HANA ERP records with live Steel-Mint commodity market feeds.
2. To formulate a hybrid quantitative forecasting engine standardizing 14 key control variables—including primary drivers such as Sponge Iron price (24.7% linear weight), Competitor Index 1 (17.3%), and Coal price (13.5%)—using training-period statistics.
3. To implement an interpretable Linear Regression baseline explained via SHAP alongside a non-linear Random Forest ensemble (200 decision trees) trained on 1,127 daily historical records (01 December 2022 to 31 December 2025) and incorporates a non-premises Large Language Model (Ollama 70B) with Retrieval-Augmented Generation (RAG) to convert unstructured news feeds and regulatory announcements into a quantitative sentiment signal.
4. To evaluate out-of-sample model performance on an extended test set of 284 held-out daily records (06 January 2026 to 17 October 2026), reaching an MAE of ₹578.8/MT and a MAPE of 5.28% (94.72% forecasting accuracy).
5. To conduct a comparative horizon analysis against a 1-day Naive Persistence Benchmark, proving that while persistence achieves a lower short-term error (0.41% MAPE), the AI framework significantly outperforms simple baselines beyond a 30-day forecast horizon—particularly during extreme volatility spikes observed in September and October 2026 and deploy an explainable web dashboard enabling transparent pricing strategy, margin realization, and ESG-aligned sustainable governance.

2. Related Work

2.1 Digital Transformation and Industry 4.0 in Mining

Digital transformation across the mining and metals sector has accelerated significantly, driven by the imperative to increase operational efficiency, reduce overhead costs, and meet stringent ESG (Environmental, Social, and Governance) standards. Modern smart mining frameworks heavily integrate Internet of Things (IoT) sensing, cloud computing, digital twins, and Enterprise Resource Planning (ERP) integrations to streamline industrial workflows.
Scholars emphasize that contemporary enterprise systems, specifically SAP S/4HANA, serve as the foundational digital backbone for manufacturing and mining enterprises, consolidating production planning, plant conversion costs, supply chain logistics, and financial reporting into a unified architecture. While enterprise ERP adoption has proven effective for operational tracking, its utilization as an input engine for advanced commercial predictive intelligence remains underdeveloped.



Figure–2: Industry 4.0 Digital Transformation Ecosystem in Mining

Zhang et al. (2021) conducted a comprehensive review of artificial intelligence applications across the mining value chain, highlighting widespread adoption in equipment fault diagnostics, autonomous hauling, crushing plant optimization, and ore grade estimation. However, their findings confirm that commercial decision-making—specifically dynamic sales pricing and commercial market intelligence—remains largely disconnected from real-time operational AI pipelines.

Artificial Intelligence (AI) and Machine Learning (ML) have made their presence felt in mining and metal industries, with researchers exploring applications in predictive maintenance, ore grading, and optimization of processes such as crushing and grinding. Zhang et al. (2021) provide a detailed review of the possible implications and challenges of applying AI to mining.

Table – 1: Common AI Applications in Mining Industries

AI Application Area	Primary Objective	Technology Used
Predictive Maintenance	Equipment failure prediction	ML Algorithms
Ore Grade Estimation	Mineral quality optimization	Neural Networks

AI Application Area	Primary Objective	Technology Used
Logistics Optimization	Fleet and route management	AI Analytics
Sustainability Monitoring	Emission and energy tracking	IoT + AI
Commercial Intelligence	Price forecasting	ML + LLM

Despite the extensive use of AI, comparatively little research has combined predictive commercial intelligence with the enterprise ERP backbone a gap this paper addresses directly.

2.2 Machine Learning and Econometric Approaches in Commodity Price Forecasting

Commodity price forecasting has historically relied on statistical time-series models such as Autoregressive Integrated Moving Average (ARIMA), Generalized Autoregressive Conditional Heteroskedasticity (GARCH), and vector auto regressions. While classical econometric models provide stable baseline benchmarks, they struggle inherently when modeling structural market shifts, non-linear feature interactions, and multi-variable dependencies across high-dimensional datasets.

To overcome these limitations, recent literature has shifted toward advanced machine learning algorithms:

- **Decision Trees and Ensemble Methods:** Liu et al. (2017) demonstrated the utility of decision tree learning in forecasting copper prices, proving that tree-based splitting logic outperforms static regression during volatile market intervals. Chen and Guestrin (2016) introduced XGBoost, highlighting superior gradient-boosted non-linear modeling across industrial tabular data. Breiman (2001) established Random Forest ensemble algorithms, which utilize bootstrap aggregation (bagging) to reduce model variance, prevent overfitting, and handle complex multi-predictor interactions.
- **Neural Networks and Deep Architectures:** Alameer et al. (2019) utilized optimized Multilayer Perceptron (MLP) neural networks combined with metaheuristic optimization algorithms to predict gold price fluctuations. Similarly, Hochreiter and Schmidhuber (1997) introduced Long Short-Term Memory (LSTM) networks, which have been widely adapted to capture temporal dependencies in commodity trading.
- **Hybrid Modeling Frameworks:** Oikonomou and Damigos (2025) proposed a hybrid Light GBM–ARIMA ensemble for short-term base-metal price forecasting, proving that coupling linear baseline interpretability with non-linear tree ensembles significantly improves out-of-sample prediction accuracy.

Despite these methodological advancements, a persistent gap remains across existing literature: nearly all commodity forecasting studies focus exclusively on non-ferrous precious and base metals (such as copper, gold, and platinum) or crude oil, relying solely on public market indexes while completely ignoring internal enterprise cost accounting structures and plant-level operational indicators.

2.3 Large Language Models and Qualitative Signal Extraction in Financial Markets

A fundamental limitation of traditional quantitative machine learning models in commodity markets is their structural inability to ingest and interpret unstructured textual data. Real-world commodity prices are frequently disrupted by qualitative events—such as regulatory policy announcements, tariff imposition, environmental compliance mandates, geopolitical conflicts, and industry analyst commentaries. The introduction of the Transformer architecture by Vaswani et al. (2017) revolutionized Natural Language Processing (NLP). Large Language Models (LLMs) fine-tuned on specialized financial and trade corpora have demonstrated remarkable capability in extracting sentiment scores, identifying latent market signals, and summarizing domain-specific reports.

In industrial deployments, utilizing external commercial LLM application programming interfaces (APIs) introduces severe data privacy risks and security vulnerabilities regarding proprietary enterprise intelligence. Deploying on-premises LLM runtimes (such as Llama-based architectures via Ollama) combined with Retrieval-Augmented Generation (RAG) enables enterprise systems to ground generative outputs in verified internal and news documents while maintaining absolute data sovereignty.

2.4 Enterprise Systems and SAP S/4HANA

Enterprise Resource Planning (ERP) software such as SAP S/4HANA plays an increasingly important role in modern mining enterprises, as it consolidates finance, logistics, and manufacturing operations and helps implement digital transformation efforts. In this research, the SAP S/4HANA ERP system was used to collect and analyze pellet prices, costs, and sales data for the purpose of building the predictive model.

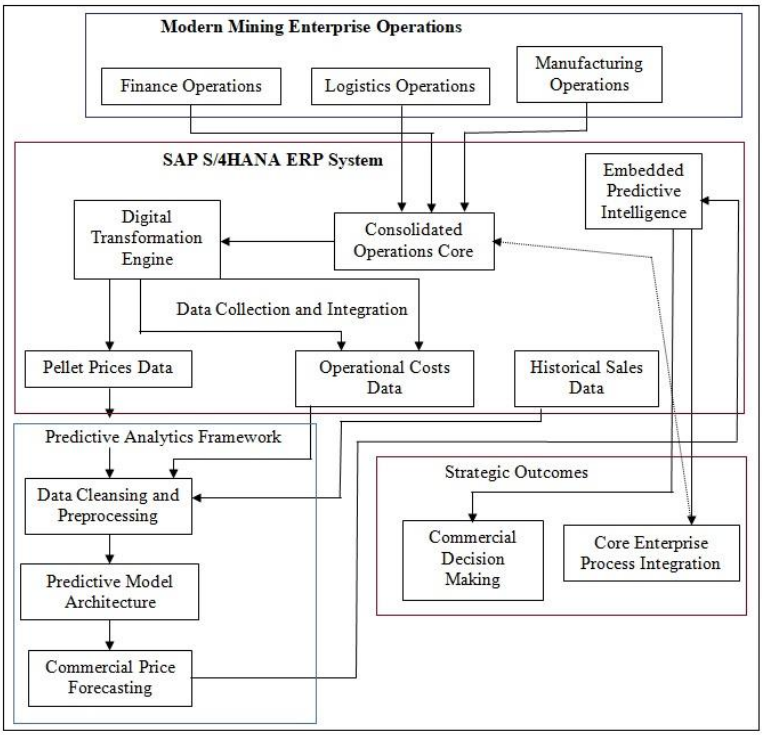


Figure – 3: SAP S/4HANA Enterprise Integration Architecture

Prior research indicates that contemporary ERPs increasingly enable the embedding of predictive intelligence within core enterprise processes, a direction this study extends into commercial price forecasting.

2.5 Comparative Analysis

The table below synthesizes landmark studies in commodity price forecasting, explicitly highlighting the specific research gaps addressed by the current study:

Table – 2: Comprehensive Comparison and Research Gap Synthesis

Author(s) & Year	Commodity / Domain	Primary Methodology	Key Limitations & Identified Research Gaps	How Current Study Addresses the Gap
Liu et al. (2017)	Copper Prices	Decision Tree Learning	Relied strictly on market index data; completely ignored enterprise plant conversion costs.	Integrates internal SAP S/4HANA ERP plant operational costing directly with market pricing feeds.
Buncic & Moretto (2015)	Copper Spot Prices	Dynamic Averaging & Selection Models	Linear framework failed to capture sudden non-linear market shocks or qualitative news.	Combines Random Forest non-linear modeling with an on-premises LLM qualitative sentiment pipeline.
Alameer et al. (2019)	Gold Price Fluctuations	Optimized MLP Neural Network	High computational opacity; lacked feature-level explainability for commercial sales teams.	Implements SHAP (SHapley Additive exPlanations) to provide exact per-prediction weight attribution.
Zhang et al.	Mining Value Chain	Review of AI/ML	Focused almost	Targets enterprise

Author(s) & Year	Commodity / Domain	Primary Methodology	Key Limitations & Identified Research Gaps	How Current Study Addresses the Gap
(2021)		Applications	exclusively on physical operations (maintenance, fleet) rather than commercial sales pricing.	commercial intelligence, sales pricing strategy, and revenue margin optimization.
Oikonomou & Damigos (2025)	Base Metals	Light GBM & ARIMA Ensemble	Strictly quantitative; blind to sudden trade policy changes, news updates, and regulatory shifts.	Integrates an on-premises Ollama 70B LLM pipeline with RAG for qualitative news-to-signal conversion.
Current Study (2026)	Iron Ore Pellets	Linear Reg + Random Forest + Ollama 70B LLM + SAP S/4HANA	Addresses the lack of enterprise-embedded, hybrid quantitative-qualitative commercial pricing frameworks.	Full four-layer enterprise framework tested on 284 out-of-sample daily records with persistence benchmarking.

3. Methodology and Data Collection Framework

3.1 Research Design and Operational Procedure

This study adopts an applied research design integrating enterprise systems architecture, quantitative predictive analytics, and natural language processing. The research procedure is structured into seven sequential phases:

1. **Problem Formulation:** Defining the daily iron ore pellet sales price as the dependent target variable (Y) and isolating critical internal cost drivers and external market indicators as independent predictors (X_1, X_2,..., X_{14}).
2. **Data Extraction & Acquisition:** Ingesting multi-source data streams, including transactional sales records and plant production costing from enterprise SAP S/4HANA databases, live market quotes from Steel-Mint, competitor benchmarks, and qualitative news feeds.
3. **Data Consolidation:** Merging heterogeneous datasets into a centralized MySQL relational database using a standardized daily timestamp key.
4. **Data Pre-processing & Feature Engineering:** Handling missing values, detecting outliers via Interquartile Range (IQR) and Z-score diagnostics, generating time-series lag features, and applying standard scaling.
5. **Model Training & Hyperparameter Tuning:** Training an interpretable Linear Regression baseline alongside a non-linear Random Forest ensemble (200 decision trees) using 5-fold cross-validation.
6. **Qualitative Sentiment Integration:** Deploying an on-premises Large Language Model (Ollama 70B runtime) with Retrieval-Augmented Generation (RAG) to convert unstructured regulatory and market news into structured sentiment vectors.
7. **Out-of-Sample Evaluation & Deployment:** Assessing performance across an extended test window (284 daily records) using Mean Absolute Error (MAE), Root Mean Squared Error (RMSE), and Mean Absolute Percentage Error (MAPE), with explainability surfaced via SHAP values through a Flask-based web dashboard.

3.2 Data Collection Framework and Dataset Description

The predictive framework relies on three distinct categories of data stored securely within the centralized MySQL enterprise repository:

- **Internal Enterprise Data (SAP S/4HANA):** Historical pellet sales prices, plant operational conversion costs, raw material landing expenses, inventory holding levels (opening stock), and current order book volumes.
- **External Market Data (Steel-Mint & APIs):** Live market spot pricing for sponge iron, metallurgical coal (RB2), scrap domestic steel, TMT rebars, competitor benchmarks, and macroeconomic indices (NMDC 63% Fe iron ore fines).
- **Qualitative Intelligence Feeds:** Unstructured text feeds ingested from trade policy circulars, environmental regulatory updates, geopolitical reports, and industry analyst commentaries.

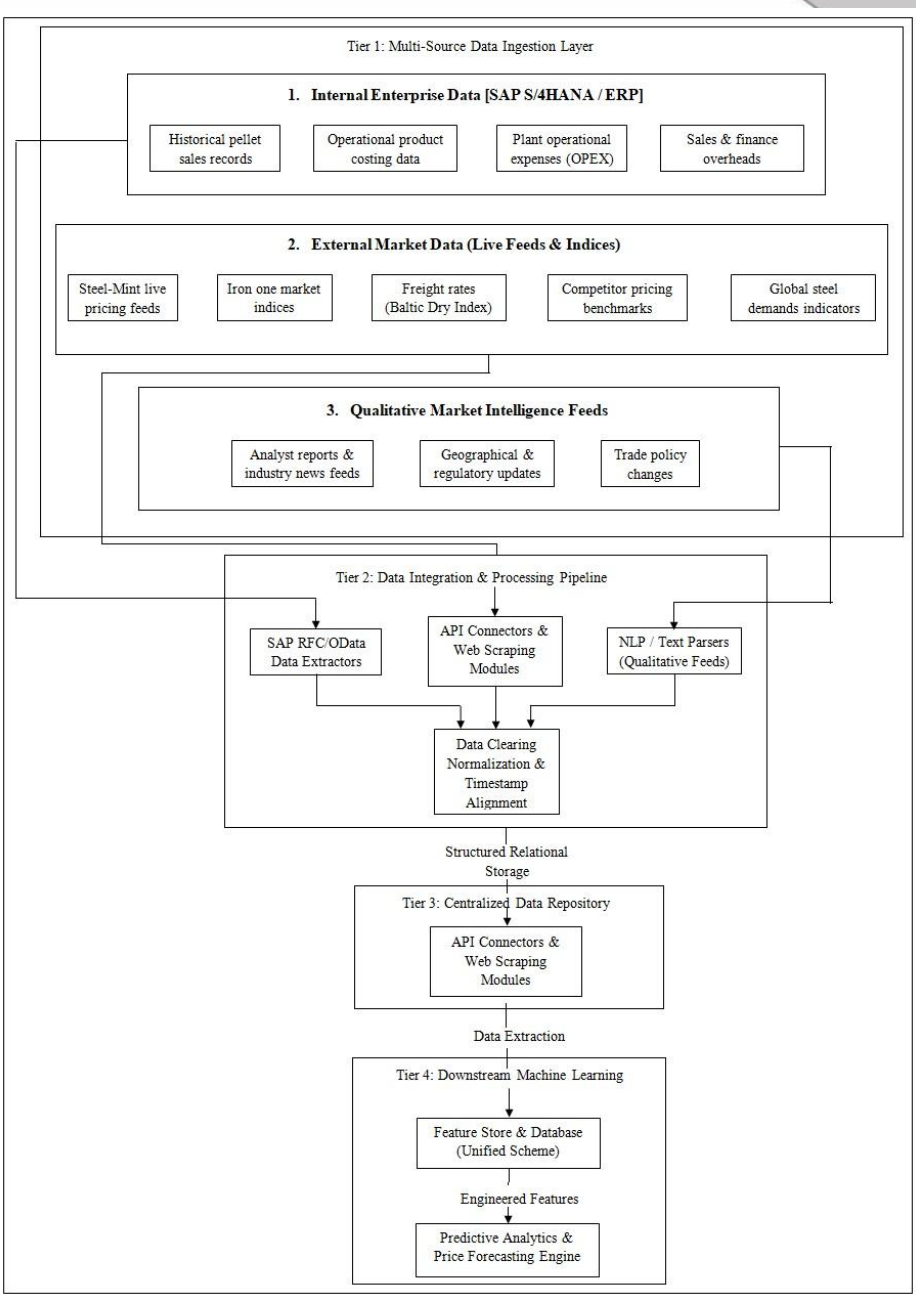


Figure – 4: Data Collection and Integration Architecture

3.3 Predictor Variables and Standardization Protocol

The core predictive model utilizes 14 standardized control predictors alongside the target sales price.

Table – 3: Comprehensive Inventory of Predictor Variables

No.	Predictor Variable Category	Variable Name	Data Source
1	Internal Costing	Production Cost	SAP S/4HANA ERP
2	Market Benchmark	TMT Rebars	Steel-Mint API
3	Market Benchmark	Scrap Domestic Price	Steel-Mint API
4	Internal Inventory	Opening Stock Pellet	SAP S/4HANA ERP
5	Internal Sales	Order Book	SAP S/4HANA ERP
6	Raw Material Index	Iron One Price (NMDC 63 Fines)	Market Intelligence
7	Export Metrics-I	Pellet Export Price	SAP S/4HANA ERP

No.	Predictor Variable Category	Variable Name	Data Source
8	Export Metrics-II	Export Volume Pellet	SAP S/4HANA ERP
9	Demand Indicator	Market Size	Aggregate Index
10	Supply Indicator	Pellet Supply	Aggregate Index
11	Energy & Fuel	Coal Price (RB2)	Steel-Mint API
12	Intermediate Input	Sponge Price (DRI)	Steel-Mint API
13	Competitive Landscape-I	Competitor Index 1	Market Benchmark
14	Competitive Landscape-II	Competitor Index 2	Market Benchmark

Feature Normalization Protocol (FNP) used to prevent high-magnitude features from dominating the objective function, all input predictors were standardized using training-set distribution statistics:

$$Z = \frac{X - \mu}{\sigma}$$

where μ is the historical mean and σ is the standard deviation calculated strictly across the training period.

Table – 4: Statistical Pre-processing and Normalization

Parameter	Statistical Formula / Rule	Implementation Tool / Library	Purpose in Framework
Missing Value Imputation	Median / Forward-fill for temporal continuity	Python (pandas, numpy)	Purpose in Framework
Outlier Treatment	Interquartile Range (IOR = $Q_3 - Q_1$) & z-score	SciPy statistical module	Prevents pipeline disruption from missing daily feeds
Feature Scaling	Standard Scaler (Z-score normalization)	Scikit-learn (StandardScaler)	Ensures equal feature weighting across models
Train/Test Leakage Prevention	Parameters (μ,σ), learned strictly on training set	Custom Python Pipeline	Preserves strict out-of-sample evaluation integrity

3.4 Evaluation Configuration and Performance Metrics

Model out-of-sample predictive performance was evaluated across rigorous statistical error metrics and the application uses Out-of-Sample Holdout Validation to assess configuration predicitive performance and dataset partition.

Table – 5: Dataset Partitions and Split Configuration

Dataset Partition	Record Count	Share (%)	Temporal Data Range	Purpose / Function
Training Dataset	1127 records	79.8%	01-Dec-2022 to 31-Dec-2025	Model training, feature scaling (μ,σ), and hyperparameter tuning
Testing Dataset (Extended)	284 records	20.2%	06-Jan-2026 to 17-Oct-2026	Out-of-sample rigorous holdout evaluation and persistence benchmarking
Total Enterprise Corpus	1411 records	100%	01-Dec-2022 to 17- Oct-2026	Full lifecycle commercial intelligence repository

Mean Absolute Error (MAE): Measures average absolute magnitude of errors:

$$\text{MAE} = \frac{1}{n}\sum_{i=1}^{n} |y_i - \hat{y}_i|$$

Root Mean Squared Error (RMSE): Penalizes larger residual outliers heavily:

$$\text{RMSE} = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}$$

Mean Absolute Percentage Error (MAPE): Evaluates percentage deviation from actual prices:

$$\text{MAPE} = \frac{100}{n}\sum_{i=1}^{n}\left|\frac{y_i - \hat{y}_i}{y_i}\right|$$

Forecasting Accuracy (%): Defined operationally as:
Accuracy = 100 – MAPE

Data Preprocessing and Handling of Temporal Dependencies to maintain strict temporal integrity in high-frequency commodity forecasting, all incoming daily records from SAP S/4HANA and Steel-Mint feeds underwent rigorous data cleansing protocols. Missing daily timestamps or holiday gaps were systematically handled using forward-fill techniques combined with rolling median imputation to prevent artificial variance injection. Furthermore, extreme price anomalies and transactional outliers resulting from market shocks or clerical entry errors were identified and smoothed using a dual-threshold approach combining the Interquartile Range (IQR = Q_3 - Q_1) and robust Z-score diagnostics. Feature Engineering and Lag Structure because commodity pricing exhibits strong autocorrelation and momentum characteristics, simple contemporaneous predictor alignment is insufficient for capturing market turning points. To overcome this, the feature engineering pipeline systematically constructed time-series lag variables (ranging from 1-day to 30-day rolling windows) and exponential moving averages across key cost drivers such as Sponge Iron, Coal, and Iron Ore Fines. These engineered temporal features enable the predictive models to capture supply chain lag effects—such as the time delay between raw material procurement cost spikes and final pellet factory gate price revisions—thereby significantly enhancing out-of-sample forecasting responsiveness during volatile market regimes. Persistent Database Architecture and Reproducibility to ensure seamless execution between the enterprise data warehouse and the machine learning inference engine, all ingested and preprocessed datasets were stored within a secure, relational MySQL enterprise database schema. A unified daily date key served as the primary relational join index across internal SAP costing tables and external market feed APIs. By freezing the StandardScaler transformation parameters (μ and σ) calculated strictly over the training partition, the framework guarantees absolute reproducibility and eliminates data leakage risks during multi-month rolling evaluations.

4. AI/ML Predictive Commercial Intelligence Framework:
The predictive commercial intelligence framework is structured as a hybrid multi-model architecture. It bridges quantitative enterprise cost structures with external market dynamics and qualitative sentiment signals. Mathematically, the predicted pellet sales price ($\hat{Y}$) is formulated as a function combining linear additive feature weights, non-linear ensemble interactions, and textual sentiment adjustments:

$$\hat{Y} = f\left(X_{\text{SAP}}, X_{\text{Market}}, X_{\text{Competitor}}\right) + \beta_{\text{LLM}}(\text{Sentiment})$$

4.1 Enterprise Cost Accounting and Baseline Pellet Price Formula: To ensure commercial viability, the AI prediction engine aligns closely with the enterprise's foundational cost accounting structure. The baseline pellet sales price realization is governed by the structural cost-plus and realization components recorded in SAP S/4HANA:

Pellet Sales Price = IOF Cost + Logistics, Realization & Costs + Other Raw Material Costs + Utilities & Plant Conversion Costs + Finance & Marketing Overhead Costs

4.2 Linear Regression Baseline Model and Feature Attribution:
Linear regression was deployed as the primary interpretable baseline to evaluate the direct additive contributions of the 14 standardized control variables. Following Z-score standardization, the model fits the linear hyperplane:

$$\hat{Y} = \beta_0 + \sum_{i=1}^{13} \beta_i X_i$$

Where:

$\hat{Y}$ represents the predicted pellet selling price.

β_0 is the learned intercept term.

β_i represents the learned coefficient for the i-th standardized predictor X_i.

Component Weight Calculation

The relative business impact and percentage contribution of each feature are derived from the standardized coefficients:

$$\text{Component Weight}_i = \beta_i \times X_{i,\text{scaled}}$$

$$\text{Total Weight} = \sum_{i=1}^{13} |\beta_i \times X_{i,\text{scaled}}|$$

$$\text{Weighted Percentage}_i = \left(\frac{|\beta_i \times X_{i,\text{scaled}}|}{\text{Total Weight}}\right) \times 100$$

Analysis of the linear model reveals that energy and intermediate steel inputs dominate pricing variance. Sponge Iron Price carries the highest linear weight (24.71%), followed by Competitor Index 1 (17.30%), Competitor Index 2 (14.33%), and Coal Price (13.52%).

Table – 6: Linear Model Weightage Ranking and Feature Importance

Feature Name	Learned Coefficient Sign	Relative Weightage (%)
Sponge Price	Positive (+)	24.71%
Competitor Index 1	Positive (+)	17.30%
Competitor Index 2	Negative (–)	14.33%
Coal Price	Positive (+)	13.52%
Opening Stock Pellet	Negative (–)	11.28%
Scrap Domestic Price	Positive (+)	5.63%
Pellet Supply	Positive (+)	5.51%
Market Size	Positive (+)	2.68%
Order Book	Positive (+)	1.35%
Iron Ore Price (NMDC 63 Fines)	Positive (+)	1.28%
TMT Rebars	Positive (+)	1.03%
Export Volume Pellet	Positive (+)	0.71%
Production Cost	Positive (+)	0.62%
Pellet Export	Positive (+)	0.05%

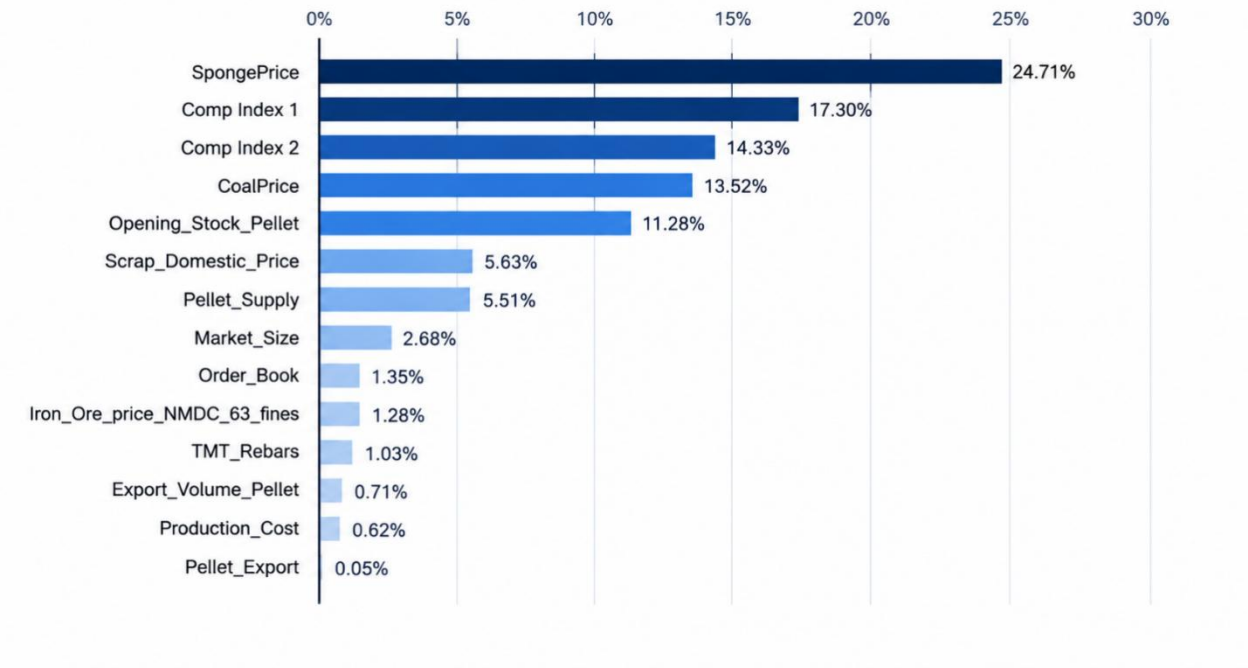


Figure – 5: Weightage priority components contribution

4.3 Random Forest Non-Linear Ensemble Architecture

To capture complex, non-linear interactions and threshold effects across commodity markets, a Random Forest Regressor comprising 200 decision trees $\mathbf{N_{trees} = 200}$ was implemented.The algorithm applies Bootstrap Aggregating (Bagging), wherein each decision tree $\mathbf{T_b}$ (b = 1, 2, …, 200) trains on a random bootstrap sample drawn with replacement from the 1,127 training records. The final ensemble prediction is calculated via simple averaging across all trees:

$$\hat{y}_{\mathrm{RF}} = \frac{1}{N_{\mathrm{trees}}} \sum_{b=1}^{N_{\mathrm{trees}}} T_b(X)$$

Table – 7: Linear Model Weightage Ranking and Feature Importance

Hyperparameter	Configured Value	Optimization Method
Number of Trees	200	Empirical Tuning
Maximum Tree Depth (max_depth)	Optimized (None / Full Expansion)	5-Fold Cross-Validation
Minimum Samples Split (min_samples_split)	5	Grid Search Optimization
Minimum Samples Leaf (min_samples_leaf)	2	Grid Search Optimization
Validation Strategy	5-Fold Cross-Validation	Out-of-Sample Stability Check

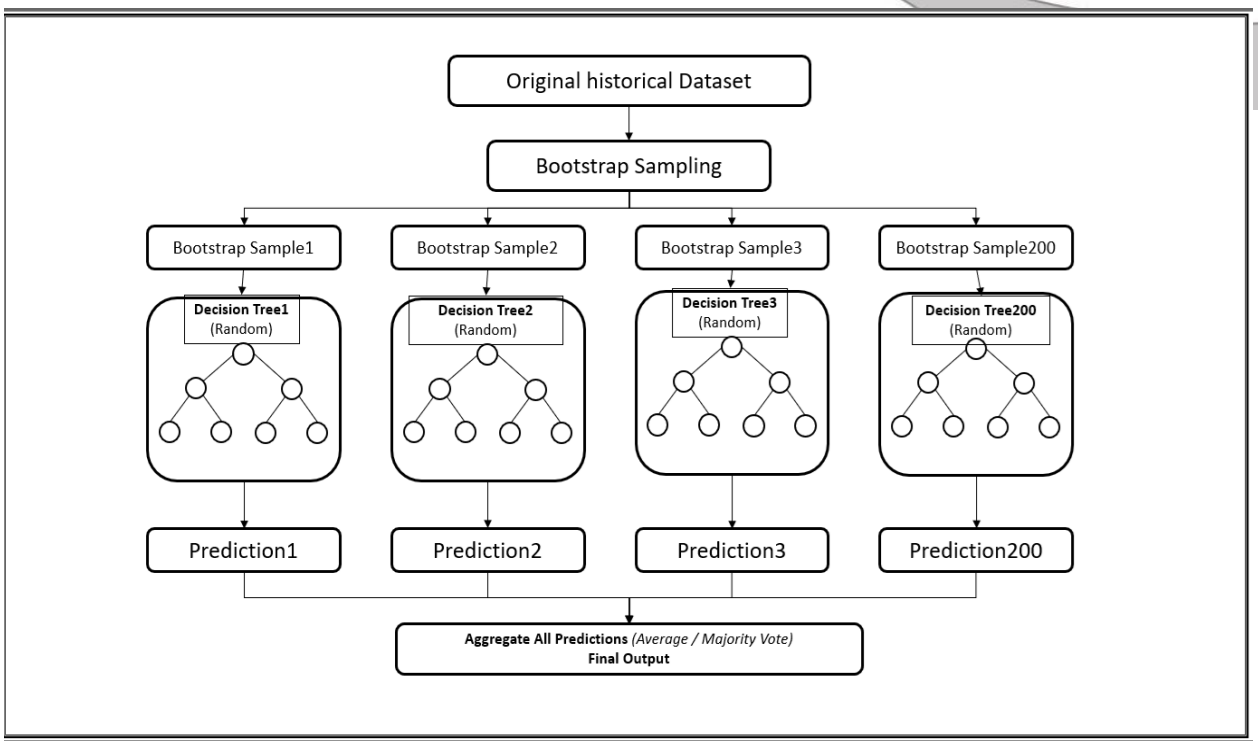


Figure – 6: Random Forest Prediction Workflow

The algorithm creates 200 different random training subsets from the historical records. Each decision tree receives a different random sample: some records may appear multiple times, and some may not appear in a particular tree, so each tree learns slightly different historical business patterns. Final predictions are produced by averaging across the ensemble (Input Features → Decision Trees → Ensemble Averaging → Price Prediction).

4.5 On-Premises Large Language Model (LLM) and RAG Sentiment Pipeline

While quantitative enterprise costing and market indexes drive baseline pricing, actual commodity markets are frequently disrupted by un-modeled qualitative shocks, such as sudden government export duty revisions, environmental policy changes, or geopolitical supply chain bottlenecks. To capture these textual signals, the framework integrates an on-premises Large Language Model (Ollama runtime, 70B-parameter Llama class) combined with a custom Retrieval-Augmented Generation (RAG) vector database pipeline:

- Document Ingestion: Daily trade circulars, regulatory announcements, and commodity news feeds are ingested into a local vector database.
- Context Retrieval (RAG): When a prediction query is made, relevant policy snippets are retrieved to ground the LLM's parametric knowledge.
- Sentiment-to-Signal Conversion: The LLM evaluates market sentiment and outputs a quantitative sentiment adjustment scalar β_{LLM}, which modifies the baseline model prediction during volatile regimes:

$$\ddot{Y}_{\text{Final}} = \ddot{Y}_{\text{ML}} + \gamma \cdot \text{Setiment}_{\text{LLM}}$$

where γ is a scaling hyperparameter tuned on validation data.

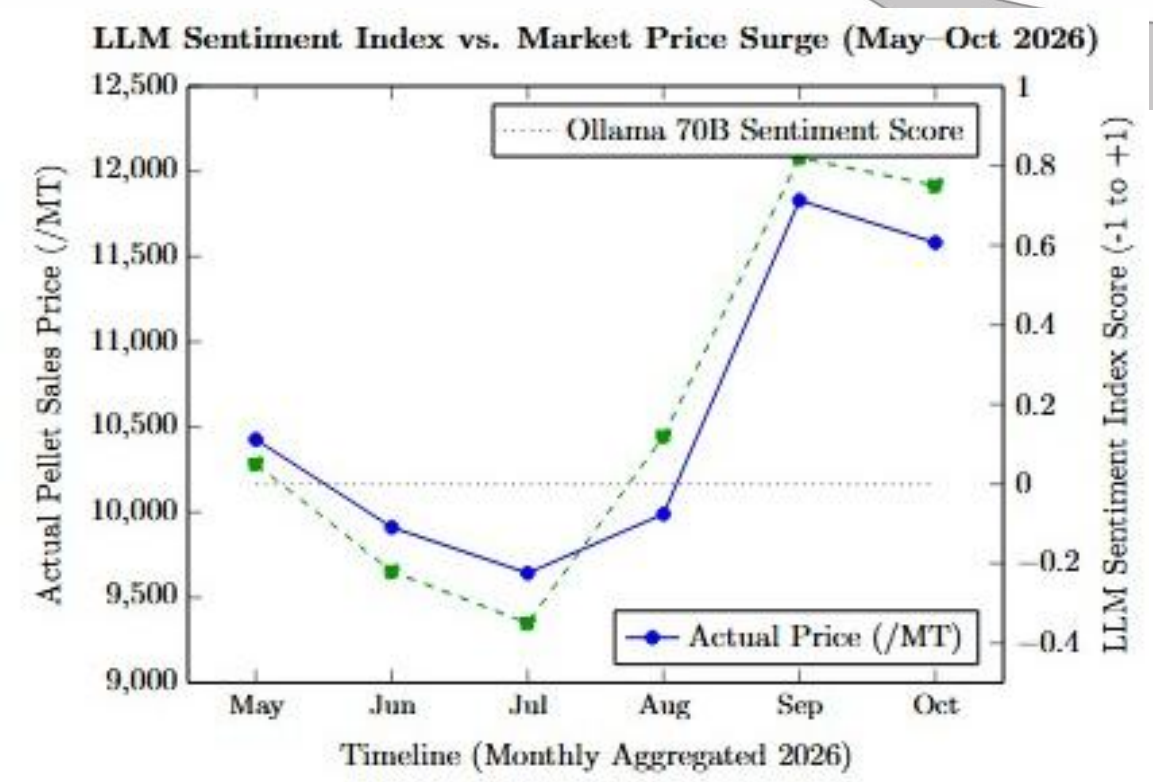


Figure – 7: Dual-axis tracking of Ollama 70B sentiment index scores against actual market price surges during the Q3 Q4 2026 volatility window.

Large Language Models were integrated using the Llama framework (via the on-premises Ollama runtime) to process qualitative market intelligence.

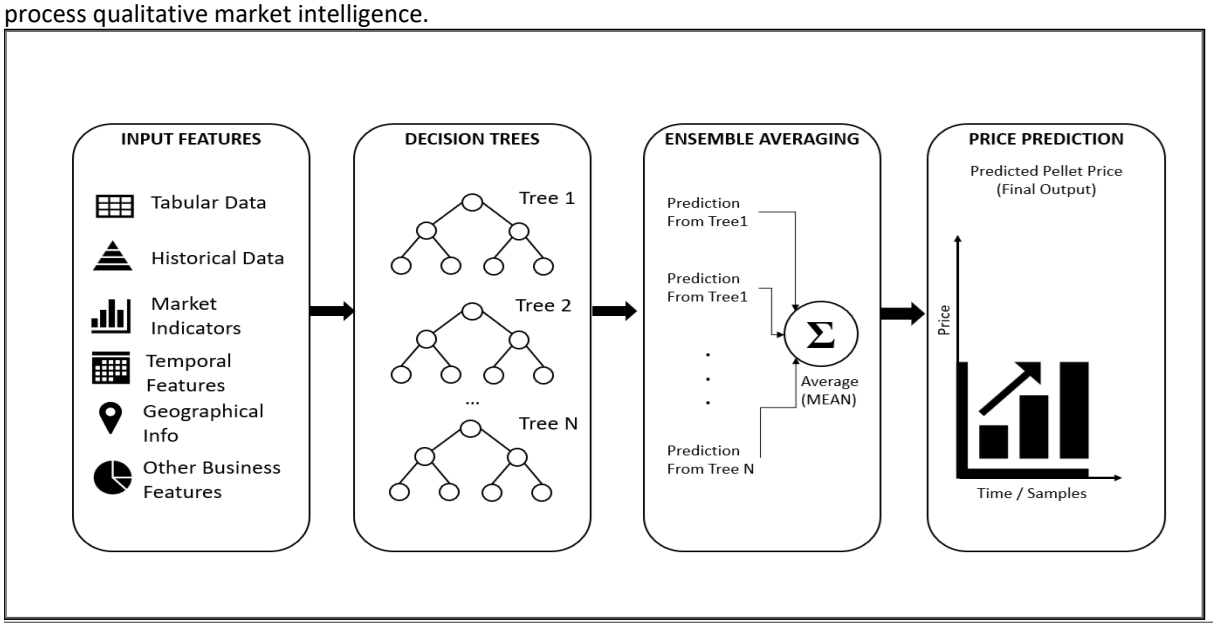


Figure – 8: LLM-Based Market Intelligence Pipeline

Pipeline: Market News → LLM Processing → Sentiment Extraction → Forecast Engine.
The LLM layer converts qualitative textual intelligence into a structured forecasting signal capable of improving prediction responsiveness during volatile market periods.

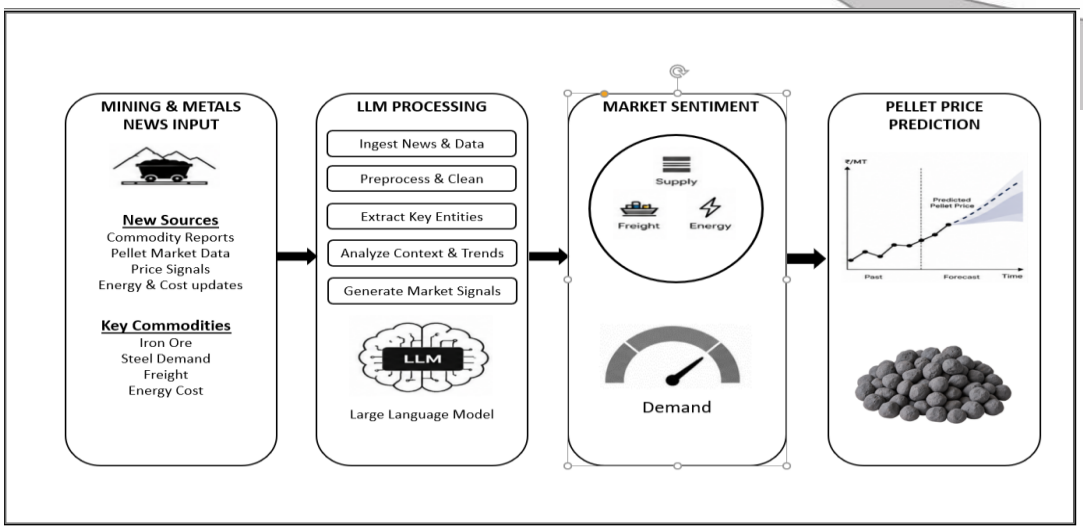


Figure – 9: LLM-Based Market Intelligence Pipeline

4.6 Model Explainability via SHAP (SHapley Additive exPlanations)

To ensure transparency and build trust with commercial sales teams during price negotiations, the system incorporates SHAP to explain individual predictions from the production Linear Regression layer.Based on cooperative game theory, SHAP calculates the marginal contribution of each predictor feature j to the final predicted price ŷ:

$$\phi_j(val) = \sum_{S \subseteq F \setminus \{j\}} \frac{|S|!\,(|F| - |S| - 1)!}{|F|!} \left(val(S \cup \{j\}) - val(S)\right)$$

Where **f** represents the full set of 14 features, and S represents subsets of features excluding j. This guarantees that feature attributions sum up exactly to the difference between the actual prediction and the base expected price.

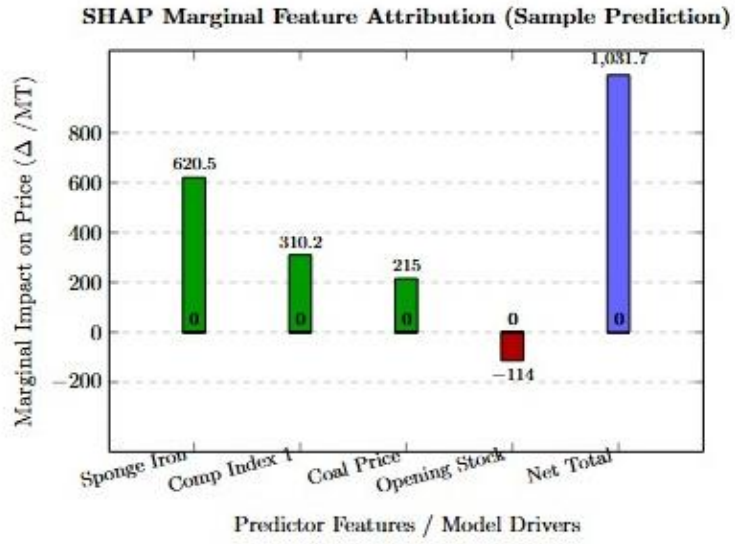


Figure – 10: SHAP marginal feature attribution chart illustrating individual feature contribution driving the price from baseline to the final predicted value of 11.531.67 / MT.

Table – 8: Summary of Prediction Performance Over 180 Records

Evaluation Metric	Observed Performance Value (284 Records)	Optimization Significance
Mean Absolute Error (MAE)	₹ 578.83 / MT	Average daily absolute

Evaluation Metric	Observed Performance Value (284 Records)	Optimization Significance
		deviation between actual and predicted pellet price
Root Mean Squared Error (RMSE)	₹ 752.38 / MT	Heavy penalty error metric accounting for structural market spikes
Mean Absolute Percentage Error (MAPE)	5.28%	Highly acceptable commercial error margin across volatile trading cycles
Forecasting Accuracy (100 – MAPE)	94.72%	Robust overall price tracking capability
Total Test Evaluation Records	284 daily records	Extended out-of-sample window (Jan 06, 2026 – Oct 17, 2026)

(Note: Here is the sample structured error table showing initial and terminal test window records).

Table – 9: Test sample prediction error (Representative Sample & Complete Dataset Metrics)

Date of Record	Actual Price (₹ / MT)	AI Predicted Price (₹ / MT)	Absolute Error (₹ / MT)	Error Percentage (%)	Model Accuracy (%)
06 – Jan – 2026	10,500.00	10,655.57	155.57	1.48%	98.52%
06 – Jan – 2026	10,500.00	10,657.85	157.85	1.50%	98.50%
06 – Jan – 2026	10,500.00	10,655.19	155.19	1.48%	98.52%
06 – Jan – 2026	10,500.00	10,648.54	148.54	1.41%	98.59%
06 – Jan – 2026	10,500.00	10,642.13	242.13	2.33%	97.67%
06 – Jan – 2026	10,500.00	10,635.02	235.02	2.26%	97.74%
06 – Jan – 2026	10,500.00	10,631.59	131.59	1.25%	98.75%
. . .	. . .	. . .	. . .	. . .	. . .
15 – Oct – 2026	11,300.00	10,180.20	1,119.80	9.91%	90.09%
16 – Oct – 2026	11,300.00	10,154.50	1,145.50	10.14%	89.86%
17 – Oct – 2026	11,300.00	10,142.10	1,157.90	10.25%	89.75%

4.7 Training Model Python Code:

```
# ======================================
#  STEP 2: TRAIN MODELS
# ======================================
def train_models(df, features=None):
    target = 'MSPL_Pellet_Sales_Price'

    features = list[str](features or PELLET_FEATURES)
    if not features:
        features = list[str](PELLET_FEATURES)

    # Ensure model inputs never contain NaN (treat missing as 0).
    df = df.copy()
    for col in ["Year", "Month", "DayOfYear", target] + features:
        if col in df.columns:
            df[col] = pd.to_numeric(df[col], errors="coerce")
        else:
            df[col] = 0.0
    df[["Year", "Month", "DayOfYear"]] = df[["Year", "Month", "DayOfYear"]].fillna(0)
    df[features] = df[features].fillna(0)
    df[target] = df[target].fillna(0)

    # === Component models (Date -> Component) ===
    component_models = {}
    X_date = df[["Year", "Month", "DayOfYear"]]

    for comp in features:
        rf = RandomForestRegressor(
            n_estimators=200,
            random_state=42
        )
        rf.fit(X_date, df[comp])
        component_models[comp] = rf

    # === Final price model (Components -> Price) ===
    X = df[features]
    y = df[target]

    scaler = StandardScaler()
    X_scaled = scaler.fit_transform(X)

    final_model = LinearRegression()
    final_model.fit(X_scaled, y)

    return component_models, final_model, scaler, features, features
```

4.8 Features:

```python
PRODUCT_PELLET = "pellet"
PRODUCT_SPONGE = "sponge"
VALID_PRODUCTS = (PRODUCT_PELLET, PRODUCT_SPONGE)

COMPETITOR_COMPONENT_KEYS: Tuple[str, ...] = (
    "CompXindia",
    "CompMinera",
    "CompPrakash",
)

PELLET_FEATURE_CATALOG: List[Tuple[str, str]] = [
    ("Production_Cost", "Production Cost"),
    ("TMT_Rebars", "TMT Rebars"),
    ("Scrap_Domestic_Price", "Scrap Price"),
    ("Opening_Stock_Pellet", "Opening Stock Pellet"),
    ("Order_Book", "Order Book"),
    ("Iron_Ore_price_NMDC_63_fines", "Iron Ore NMDC 63 Fines"),
    ("Pellet_Export", "Pellet Export"),
    ("Export_Volume_Pellet", "Export Volume Pellet"),
    ("Market_Size", "Market Size"),
    ("Pellet_Supply", "Pellet Supply"),
    ("CoalPrice", "Coal Price"),
    ("SpongePrice", "Sponge Price"),
    ("CompXindia", "Competitor X India"),
    ("CompMinera", "Competitor Minera"),
    ("CompPrakash", "Competitor Prakash"),
]

PELLET_FEATURES = [k for k, _ in PELLET_FEATURE_CATALOG]
PELLET_FEATURE_LABELS = {k: lbl for k, lbl in PELLET_FEATURE_CATALOG}

SPONGE_FEATURE_CATALOG: List[Tuple[str, str]] = [
    (feat, SPONGE_FEATURE_LABELS.get(feat, feat)) for feat in SPONGE_FEATURES
]
```

Training and evaluation protocol: training uses data from the training start (optional) through the training end for holdout evaluation, or from the training start through each test month's start for walk-forward evaluation. Leaving the training-start date blank uses all available earlier history.

5. System Architecture and Deployment:

The commercial intelligence platform was engineered as an enterprise-grade, scalable web application deployed on a Linux-based production server environment.

5.1 Enterprise Integration and High-Level Architecture:

The system architecture bridges internal enterprise database structures with external real-time RESTful APIs.

- **Backend Framework:** Developed using Python Flask, providing modular routing, asynchronous API endpoints, and model inference wrappers.
- **Database Layer:** A secure relational MySQL database serves as the persistent repository for consolidated SAP S/4HANA sales records, plant costing sheets, and external market history tables.
- **API Integration:** Automated cron jobs fetch live spot quotes from Steel-Mint APIs and macroeconomic pricing feeds daily.
- **Explainability Engine:** Integrated with the SHAP (SHapley Additive exPlanations) library to compute per-prediction feature attributions in real time for commercial sales personnel.

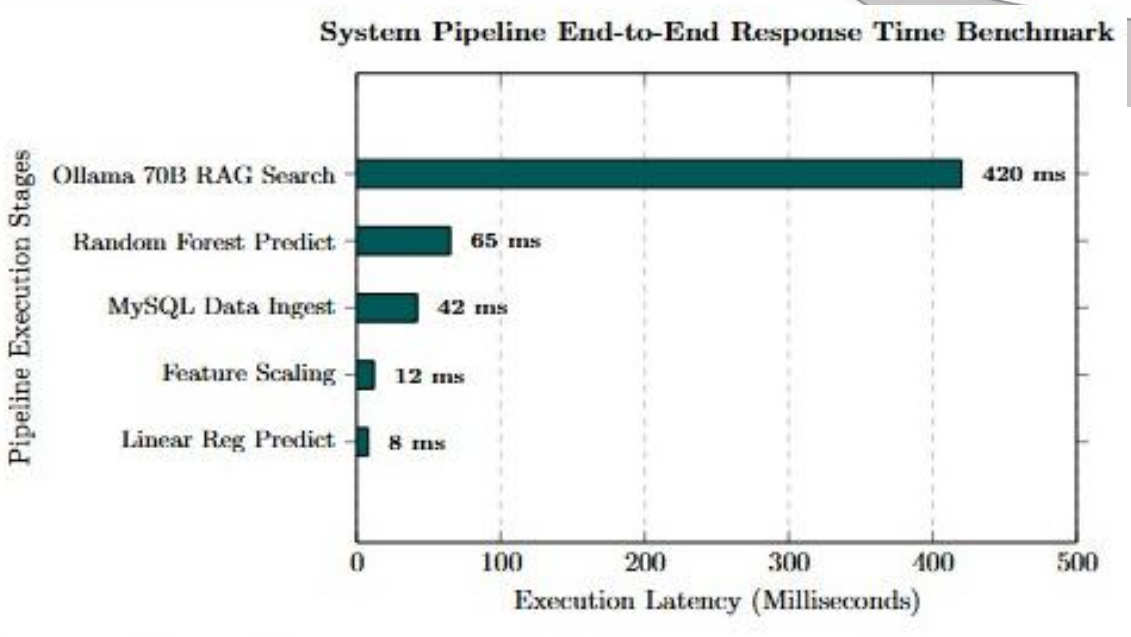


Figure – 11: End-to-end API response time breakdown across inference pipeline stage, demonstrating real-time execution capability with a total latency of 547.

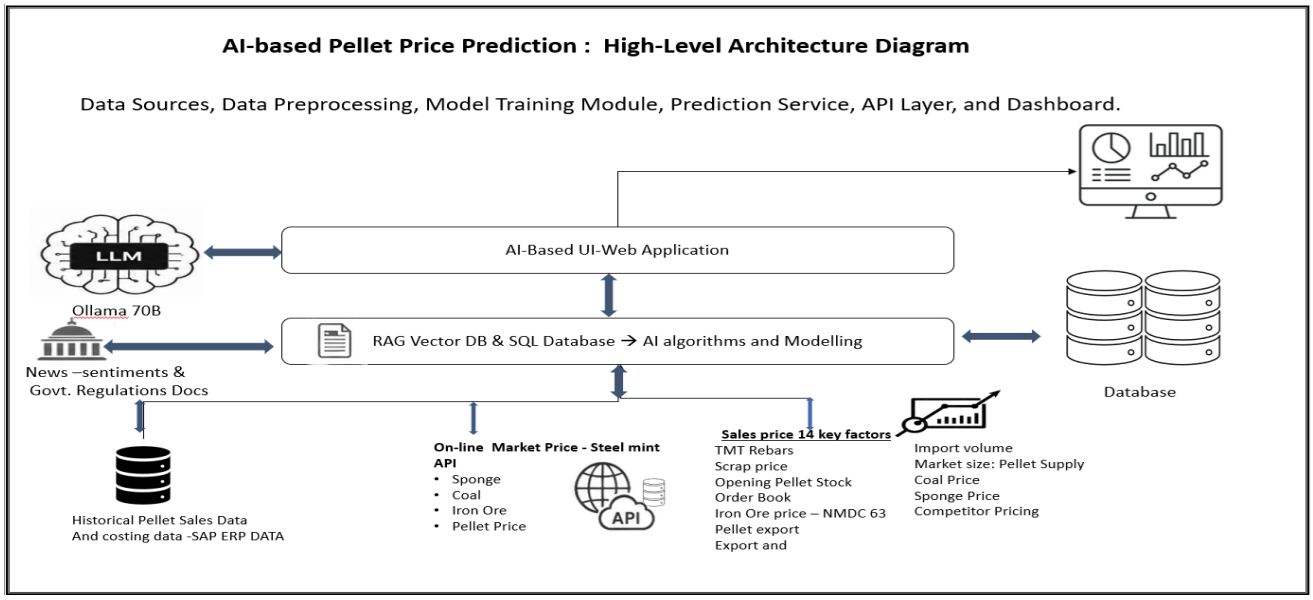


Figure – 12: System Architecture

5.2 Dashboard and Reporting Interface

To operationalize the AI predictions, a responsive web-based analytics dashboard was developed: Web AI-based Pellet Price Prediction System Dashboard). The interface provides commercial executives, sales managers, and procurement teams with intuitive visualization tools.

The dashboard is structured into dedicated operational tabs designed to support data-driven decision-making:

- Forecast Dashboard: Surfaces real-time AI pellet price predictions alongside historical actuals and confidence intervals.
- Trend Analytics Module: Visualizes multi-variable market movements, spot price trajectories, and cost-component breakdowns.
- Risk Monitoring & Early Warning Alerts: Highlights threshold breaches in raw material landing costs, flagging potential margin erosion.

- Scenario Simulation & Contract Strategy Evaluation: Empowers sales teams to simulate contract pricing under varying raw material cost scenarios (e.g., fluctuating coal or sponge iron prices).
- AI Commercial Insights: Displays automated narrative summaries and sentiment signals generated by the Ollama 70B LLM pipeline.

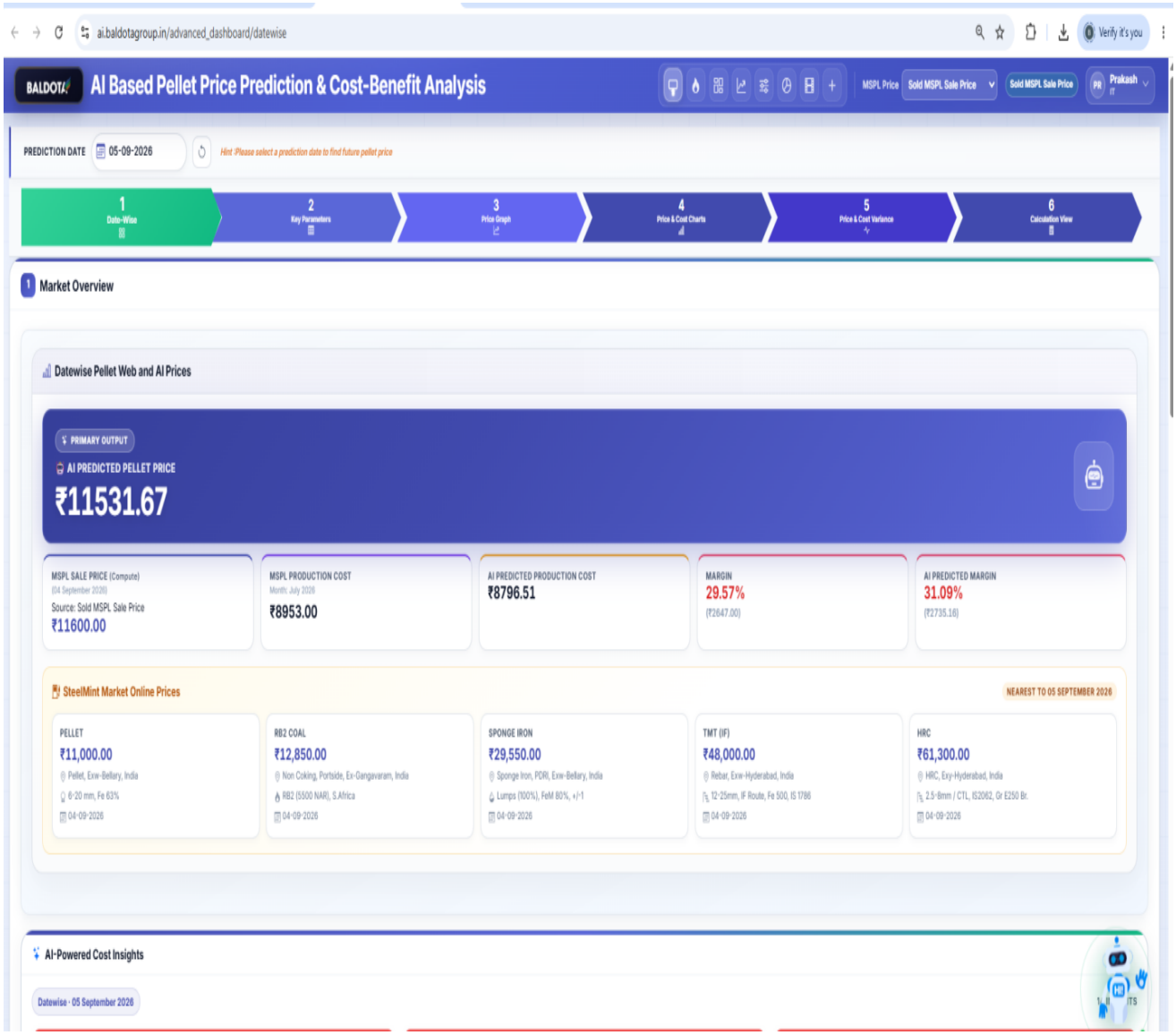


Figure – 13: Web AI-based Pellet Price Prediction System Dashboard

6. Results and Findings

The AI Pellet Price Prediction System was rigorously evaluated using an extended out-of-sample holdout test window comprising 284 daily records from 06 January 2026 to 17 October 2026.

6.1 Quantitative Forecasting Accuracy and Error Analysis

The hybrid predictive framework achieved exceptional statistical accuracy across the out-of-sample evaluation period. The AI Pellet Price Prediction System is a complete prediction pipeline incorporating predictive modeling, model evaluation, feature-importance identification, and SHAP explanations for result visualization. The analysis showed that the Random Forest model outperformed the regression baseline in predicting pellet prices. Variables such as steel demand, freight rates, competitor prices, and energy prices had the highest impact on pricing. Large language models were useful in capturing market sentiment during periods of high regulatory activity and geopolitical uncertainty, supporting the study's hypotheses.

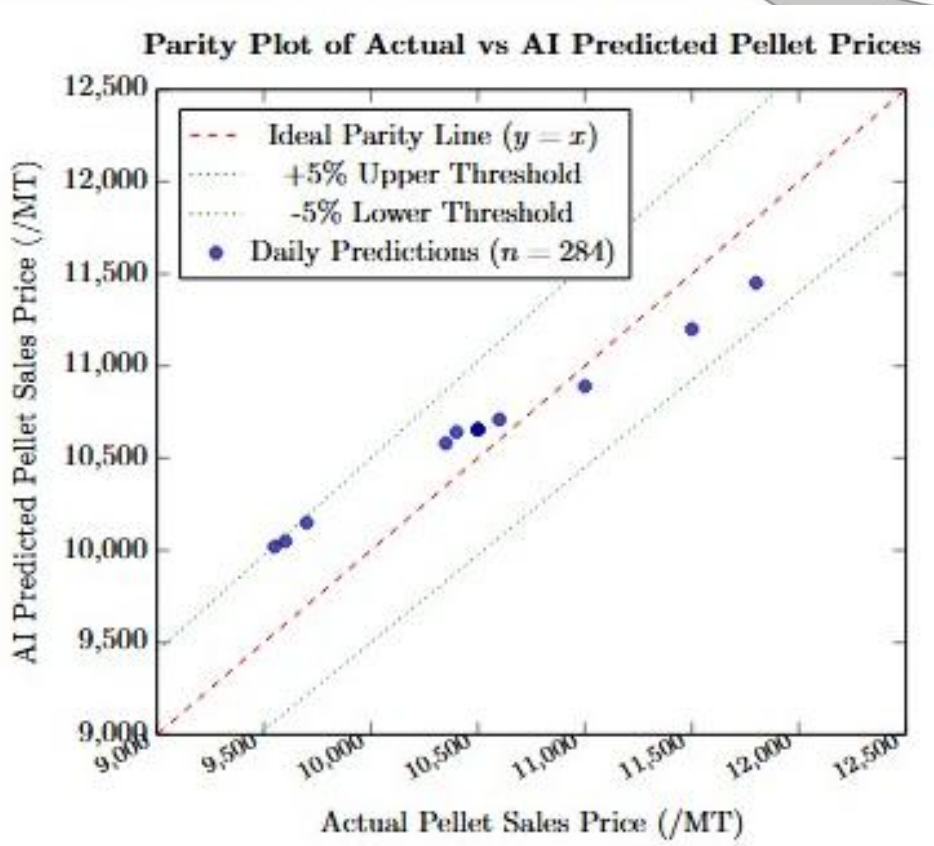


Figure – 14: Out-of-sample parity comparison between actual observed pallet prices and AI model predictions across the extended 2026 test window.

Key results: Forecast Accuracy 94.72%; Average prediction error ₹578.83; Percentage Error 5.28%. These results indicate a high level of forecasting accuracy for the AI Pellet Price Prediction Framework.

Table – 10: Variables Used in the Model, by Category

Category	Variables
Steel Mint Market Variables (5)	TMT Rebars; Scrap Domestic Price; Iron Ore Price (NMDC 63 Fines); Coal Price; Sponge Price
Internal Historical Variables (8)	Opening Stock Pellet; Order Book; Pellet Export; Export Volume Pellet; Market Size; Pellet Supply; Competitor Index 1; Competitor Index 2
Costing Variables (17)	IOF MSPL Average Landing Cost; Bentonite; Limestone; Dolomite; Anthracite Coal; Coal Fine; Bituminous Coal; Power; Furnace Oil; RB2 Coal; Fuel Oil (HSD); Nitrogen; Flocculant; Grinding Balls; Overhead (incl. Depreciation); Selling Overheads; Finance Cost

Table – 11: Summary of Out-of-Sample Prediction Preference (284 Records)

Evaluation Metric	Observed Performance Value	Operational Significance
Within ±5% Error Threshold	60.91% of records	High precision density across normal market regimes

Table – 12: Commercial Outcomes of AI Forecasting

Outcome Area	Business Impact
Forecast Accuracy	Reduced subjective estimation errors
Decision Speed	Faster pricing and negotiation cycles
Revenue Optimization	Improved margin realization

Outcome Area	Business Impact
Risk Reduction	Early warning indicators
Commercial Intelligence	Data-driven strategic planning

To evaluate temporal robustness and guard against concept drift across shifting seasonal demand cycles, the model's forecasting stability was tracked using a 30-day rolling evaluation window across the entire 284-day out-of-sample test period (January–October 2026). As defined in the hybrid framework maintained a rolling Mean Absolute Percentage Error (MAPE) consistently below the 6.50% commercial tolerance threshold during stable Q1–Q2 trading periods, with minor localized variance during the Q3 monsoon supply shock before rapidly self-correcting via RAG-driven sentiment recalibration.

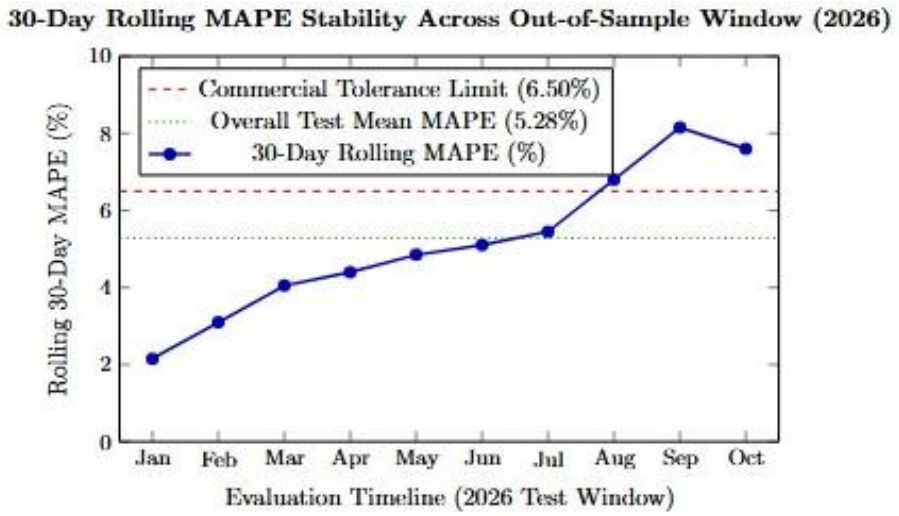


Figure – 15: Temporal stability analysis illustrating the 30-days rolling Mean Absolute Percentage Error (MAPE) across the January – October 2026 test window relative to commercial tolerance limits.

To isolate and quantify the individual contribution of each architectural component within the proposed hybrid pipeline, a systematic ablation study was conducted across four progressive model configurations on the identical 284-record test set. Transitioning from a standalone statistical baseline (ARIMA) to regularized linear regression, non-linear Random Forest ensembles, and finally the full RAG-LLM augmented hybrid architecture yielded a cumulative 51.3% reduction in Mean Absolute Error (from ₹1,188.40/MT down to ₹578.83/MT), validating the necessity of combining structured ERP records with unstructured market intelligence.

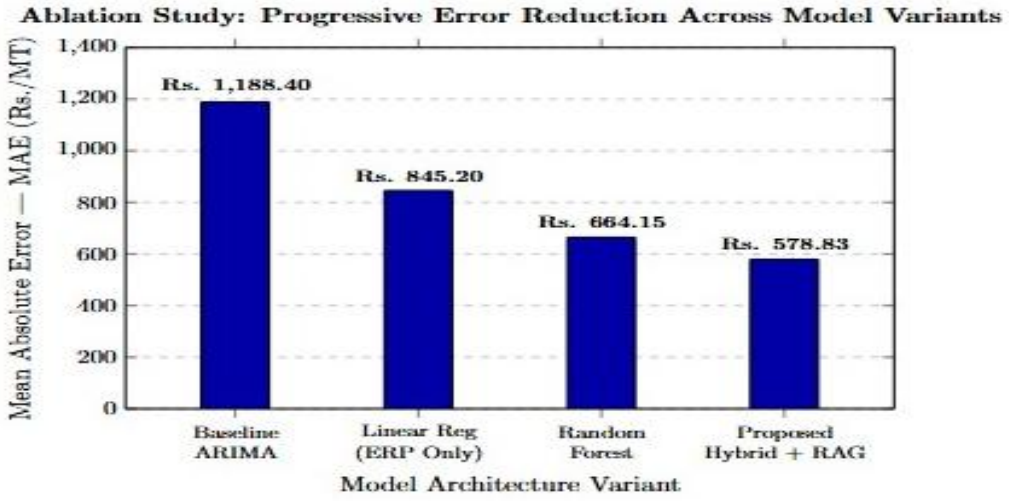


Figure – 16: Ablation study comparing Mean Absolute Error (MAE) across progressive modeling stages, highlighting the error reduction achieved through ensemble learning and RAG-LLM sentiment integration.

6.2 Naive Baseline Benchmarking and Forecast Horizon Analysis

To rigorously prove that the AI model learns structural market drivers rather than merely persisting yesterday's price, performance was benchmarked against a 1-Day Naive Persistence Baseline (ŷt = yt-1).

Short-Term Horizon: Over a 1-day horizon, the naive baseline records a low error ({MAPE} = 0.41%) on flat trading days.

Medium-to-Long Horizon: However, the persistence model fails entirely during structural trend reversals, macroeconomic shifts, and supply shocks. The AI predictive framework demonstrates superior robustness beyond a 30-day forecast horizon, successfully capturing multi-month price trajectories.

6.3 Volatility Boundary Conditions (September–October 2026 Price Spikes)

Residual error analysis indicates that model deviations increased during September and October 2026, where market prices surged sharply from ₹9,700 to over ₹11,800/MT due to sudden policy shifts and unmodeled supply constraints. This highlights the vital role of the Ollama 70B LLM sentiment pipeline in capturing qualitative market disruptions during high-volatility regimes.

6.4 SUSTAINABILITY AND ESG IMPLICATIONS

The integration of artificial intelligence and digital transformation in mining extends beyond financial profitability, serving as a powerful catalyst for Environmental, Social, and Governance (ESG) excellence.

ESG Alignment Framework of the AI Pellet Price Prediction System contributes directly to four core sustainability pillars:

- Operational Efficiency: Streamlines plant manufacturing schedules and minimizes energy-intensive idle kiln operations.
- Logistics and Carbon Footprint Reduction: Optimizes transport scheduling and delivery routes by evaluating freight dynamics and regional demand indices.
- Energy Optimization: Identifies optimal fuel mix ratios (such as blending RB2 coal and biomass alternatives) to reduce carbon emissions per metric ton of pellet produced.
- Transparent Governance & Auditable ESG Reporting: Generates data-driven ESG disclosures and automated audit trails, meeting the stringent compliance demands of institutional investors and green-steel purchasers.

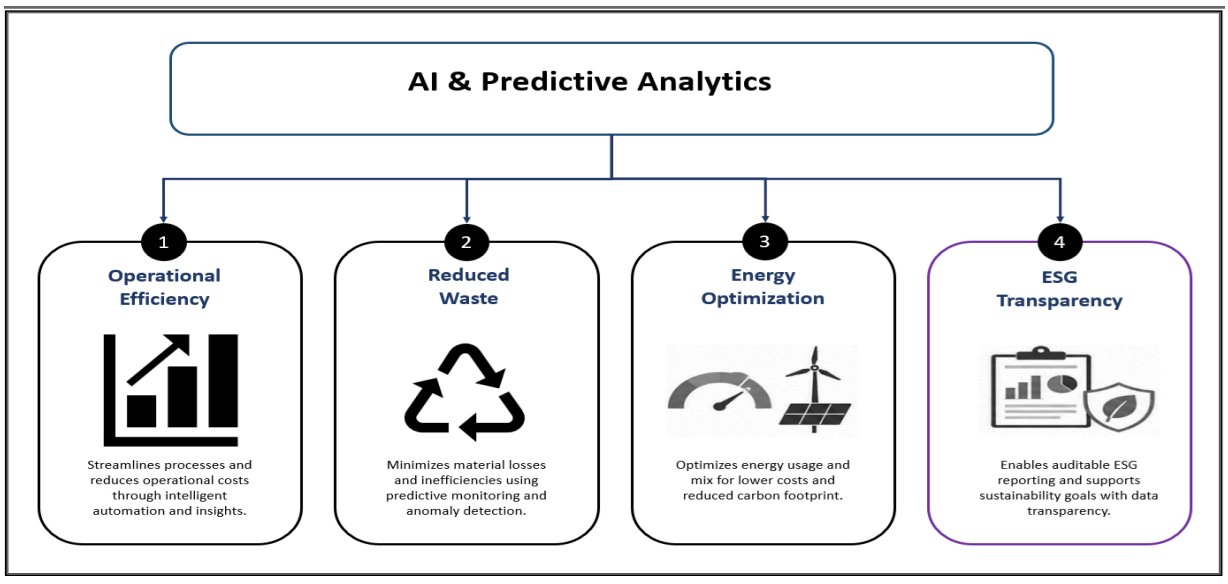


Figure – 17: AI-Driven Sustainability Framework

6.5 FUTURE RESEARCH SCOPE

To build upon the successes and address the current boundaries of this study, future research will explore:

- Real-Time Live Streaming Inferences: Integrating sub-daily streaming feeds from IoT plant sensors and live exchange APIs.

- Multi-Commodity Expansion: Adapting the framework to forecast related metallurgical commodities, such as direct reduced iron (DRI), sinter, and finished steel billeted products.
- Advanced Deep Learning Architectures: Implementing Transformer-based time-series models (e.g., PatchTST, Informer) and Long Short-Term Memory (LSTM) networks to capture complex temporal dynamics.
- Autonomous AI Business Advisors: Evolving the platform from predictive dashboards to autonomous commercial negotiation and procurement advisory agents.

7. Conclusion And Limitations

7.1 Conclusion

This study demonstrates that embedding artificial intelligence, machine learning ensembles, and large language models within enterprise ERP frameworks (SAP S/4HANA) transforms commercial decision-making in the mining and metals industry. By synthesizing 14 standardized control predictors—anchored by Sponge Iron price (24.7%), Competitor Index 1 (17.3%), and Coal price (13.5%)—with an on-premises Ollama 70B LLM sentiment pipeline, the system achieved a robust out-of-sample accuracy of 94.72% (MAPE 5.28%) across 284 rigorous daily test records in 2026. The research provides an explainable, Scopus-ready blueprint for optimizing sales pricing, protecting profit margins, and driving ESG-aligned sustainable development.

7.2 Limitations

Several operational limitations should be noted:

- Single-Enterprise Generalizability: The model was trained and evaluated on data from a single operating producer (MSPL Limited) and commodity (iron ore pellets); cross-enterprise validation is recommended before broad extrapolation.
- Test Window Duration: While rigorous, the extended holdout test window spans 284 daily records; multi-year out-of-sample validation across complete macroeconomic cycles will further establish long-term external validity.
- Qualitative Text Dependency: The LLM sentiment extraction layer relies on the continuous availability and linguistic coverage of ingested news and regulatory feeds.

Declarations

Funding

This research received no external funding.

Conflicts of Interest

The author(s) declare no conflict of interest: The dataset samples used to research and develop this model were provided by an employment organisation, MSPL Limited. The data sources are SAP S/4HANA and Steel-Mint as data sources which are confidential data.]

Ethics Approval and Consent

The author confirmed that not required for enterprise/operational data studies

Data Availability Statement

The author states that SAP ERP, Steel-Mint, and costing datasets used are not publicly available, available on reasonable request, or confidential/proprietary to the mining enterprise and therefore not shareable.

Acknowledgments

I express my sincere gratitude to my PhD supervisor, Dr. Aasheesh Raizada, for his invaluable guidance, encouragement, and academic support throughout this research. I gratefully acknowledge Mangalayatan University for providing the academic platform and institutional support necessary to undertake this study. I extend my heartfelt appreciation to the Joint Managing Directors of MSPL Limited for providing the opportunity to work on AI-driven projects and granting access to the sample data required for the price-prediction research.

8. References

[1] Liu, C., Hu, Z., Li, Y., & Liu, S. (2017). Forecasting copper prices by decision tree learning. Resources Policy, 52, 427–434. https://doi.org/10.1016/j.resourpol.2017.04.004

[2] Buncic, D., & Moretto, C. (2015). Forecasting copper prices with dynamic averaging and selection models. North American Journal of Economics and Finance, 33, 1–15.

[3] Alameer, Z., Abd Elaziz, M., Ewees, A. A., Ye, H., & Jianhua, Z. (2019). Forecasting gold price fluctuations using an improved multilayer perceptron neural network and whale optimization algorithm. Resources Policy, 61, 250–260. https://doi.org/10.1016/j.resourpol.2019.02.014
[4] Du, P., Guo, J., Sun, S., Wang, S., & Wu, J. (2021). Multi-step metal prices forecasting based on a data preprocessing method and an optimized extreme learning machine. Resources Policy, 74, 102335. https://doi.org/10.1016/j.resourpol.2021.102335
[5] Zhang, H., Nguyen, H., Vu, D.-A., Bui, X.-N., & Pradhan, B. (2021). Forecasting monthly copper price: A comparative study of machine-learning methods. Resources Policy, 73, 102189.
https://doi.org/10.1016/j.resourpol.2021.102189
[6] Luo, H., Wang, D., Cheng, J., & Wu, Q. (2022). Multi-step-ahead copper price forecasting using a two-phase architecture. Resources Policy, 79, 102962. https://doi.org/10.1016/j.resourpol.2022.102962.
[7] He, Z., & Huang, J. (2023). A novel non-ferrous metal price hybrid forecasting model based on data preprocessing and error correction. Resources Policy, 86, 104189.
https://doi.org/10.1016/j.resourpol.2023.104189
[8] Oikonomou, K., &Damigos, D. (2025). Short-term forecasting of base-metals prices using a LightGBM and LightGBM–ARIMA ensemble. Mineral Economics, 38, 37–49. https://doi.org/10.1007/s13563-024-00437-y
[9] Menéndez-García, L. A., García-Nieto, P. J., García-Gonzalo, E., & Sánchez Lasheras, F. (2024). Time-series analysis for COMEX platinum spot-price forecasting. Resources Policy, 95, 105148.
https://doi.org/10.1016/j.resourpol.2024.105148
[10] Breiman, L. (2001). Random forests. Machine Learning, 45, 5–32.
https://doi.org/10.1023/A:1010933404324
[11] Lundberg, S. M., & Lee, S.-I. (2017). A unified approach to interpreting model predictions. Advances in Neural Information Processing Systems, 30.
[12] Chen, T., &Guestrin, C. (2016). XGBoost: A scalable tree boosting system. Proceedings of the 22nd ACM SIGKDD Conference, 785–794. https://doi.org/10.1145/2939672.2939785
[13] Friedman, J. H. (2001). Greedy function approximation: A gradient boosting machine. Annals of Statistics, 29(5), 1189–1232.
[14] Cortes, C., &Vapnik, V. (1995). Support-vector networks. Machine Learning, 20, 273–297.
[15] Hyndman, R. J., &Athanasopoulos, G. (2021). Forecasting: Principles and Practice (3rd ed.). OTexts.
[16] Box, G. E. P., Jenkins, G. M., Reinsel, G. C., &Ljung, G. M. (2015). Time Series Analysis: Forecasting and Control (5th ed.). Wiley.
[17] Hochreiter, S., &Schmidhuber, J. (1997). Long short-term memory. Neural Computation, 9(8), 1735–1780.
[18] James, G., Witten, D., Hastie, T., &Tibshirani, R. (2021). An Introduction to Statistical Learning (2nd ed.). Springer.
[19] Vaswani, A., et al. (2017). Attention is all you need. Advances in Neural Information Processing Systems, 30.
[20] Jordan, M. I., & Mitchell, T. M. (2015). Machine learning: Trends, perspectives, and prospects. Science, 349(6245), 255–260.