Ved International Journal of Arts, Commerce and Technology (VIJACT)
© VIJACT |Vol.2, Issue 9, Sep 2026 (Special Issue), pp. 76-83 | ISSN: 3139-1656 (Online)
DOI: https://doi.org/10.65785/vijact.v2i9.96
Available online at: https://vijact.com/

Neural Network Procure-to-Pay and Supply Chain Intelligence -Augmented Cloud ERP Architecture for Predictive

Manykandaprebou Vaitinadin
Independent Researcher, California, USA | SAP Enterprise Architect
ORCID ID: 0009-0005-7386-5077

Article Info

Article history:

Received Aug 10, 2026

Accepted Aug 24, 2026

Published Sep 08, 2026

Keywords:

cloud ERP

Neural Networks

SAP S/4HANA

SAP BTP

Predictive Supply Chain, Procure-to-pay

Late-delivery risk

Machine Learning

Industry 4.0

ABSTRACT

Cloud ERP programs increasingly expose operational events to external analytics and AI services, but the practical question is how a predictive model should be placed inside a governed transaction process. This paper proposes a neural network-augmented cloud ERP architecture for predicting late-delivery risk in procure-to-pay and supply-chain execution. A reproducible simulation study was conducted on 60,000 synthetic ERP-like purchase and shipment records using only pre-outcome features available at or before an operational decision point. A multilayer perceptron was compared with logistic regression and random forest baselines using a chronological 80/20 split. The neural model achieved an ROC-AUC of 0.818 (95% bootstrap interval 0.810–0.826) and 0.700 accuracy; random forest achieved 0.816 ROC-AUC and the highest F1-score of 0.759. The small performance gap is itself operationally relevant: model complexity should be justified by measurable benefit rather than assumed. The architecture maps ERP events, cloud integration, feature services, model inference, workflow controls, and human approval into a single deployment pattern. The study contributes a leakage-aware evaluation protocol and a practical blueprint for integrating predictive intelligence into SAP S/4HANA-style cloud ERP landscapes without allowing the model to become the system of record

Corresponding Author:
Manykandaprebou Vaitinadin
Email: mkprebou1@gmail.com

1. Introduction:

Enterprise resource planning systems remain the transactional backbone of procurement, inventory, logistics, and financial control. Their core strength is deterministic execution: a purchase order is released only when configuration, authorization, pricing, and workflow conditions are satisfied. Predictive analytics introduces a different operating logic. Instead of recording what has already happened, a model estimates what is likely to happen next and asks the organization to intervene before an exception becomes visible in the ERP document flow.

This distinction matters in procure-to-pay (P2P). A buyer may know the requested delivery date, supplier history, shipping mode, order value, material criticality, and inventory coverage when a purchase order is created or confirmed. The actual late-delivery outcome is not yet known. If a model can convert those pre-outcome attributes into a calibrated risk score, the score can be used to prioritize follow-up, request an earlier supplier confirmation, expedite a critical item, or route an exception to a planner. The prediction has business value only when it is integrated into the transaction process rather than left in a separate data-science dashboard.

Research on supply-chain analytics has established the value of predictive methods for demand, logistics, and risk [3]–[8]. At the same time, data quality and leakage remain persistent concerns in supply-chain modeling [4], [8]. Enterprise deployment adds another layer of difficulty: the prediction service must coexist with master-data governance, authorization, workflow, audit logging, model monitoring, and fallback procedures. Cloud platforms make this technically feasible, but a deployable reference architecture needs to connect model inference with operational controls.

The study addresses three questions. RQ1 asks whether a compact neural network can discriminate late-delivery risk using features available before the outcome occurs. RQ2 asks how its predictive performance compares with conventional logistic-regression and random-forest baselines. RQ3 asks how the model can be embedded into a cloud ERP architecture while preserving ERP transaction authority and human accountability. The objective is not to claim a universal best algorithm. It is to connect a reproducible predictive experiment to an enterprise architecture that can be implemented and governed.

2. REVIEW OF RELATED WORK

Predictive analytics in supply chains has moved from descriptive reporting toward forecasting, classification, and prescriptive decision support. Schoenherr and Speier-Pero describe the growing role of data science and predictive analytics in supply-chain management [3], while Seyedan and Mafakheri survey machine-learning and neural-network approaches for supply-chain forecasting [5]. These studies emphasize that predictive value depends on the availability, quality, and timeliness of operational data.

Data quality is especially important in ERP environments because fields are created by different organizational units and at different points in a process. Hazen et al. show that poor data quality can undermine predictive analytics in supply chains [4]. A related methodological problem is target leakage: a model can appear highly accurate when it is allowed to use fields that become available only after the business outcome. Recent work using the DataCo supply-chain benchmark explicitly demonstrates the importance of removing leakage variables before evaluating late-delivery models [8].

Neural methods are increasingly used for logistics and delivery prediction. Ahmed et al. combine self-organizing maps, principal-component analysis, artificial neural networks, and SHAP-based interpretation for supply-chain forecasting [6]. Faulkner et al. report a multi-task deep-learning approach for uncertainty-aware delivery-delay duration prediction on more than ten million industrial shipment records [7]. These results show that nonlinear models can capture interactions that are difficult to represent with a single linear decision surface. They do not, however, remove the need for simpler baselines or for deployment controls.

The deployment side is equally important. SAP Business Technology Platform (BTP) combines integration, application development, data, analytics, and AI capabilities [12]. SAP Event Mesh supports asynchronous communication and event-driven extensions for SAP S/4HANA landscapes [11]. SAP also documents machine-learning services in which historical business data is used for training on BTP and inference is integrated back into SAP S/4HANA through governed interfaces [13]. These platform capabilities motivate the architecture proposed in this paper, but the neural risk engine described here is a research prototype rather than a claim of standard SAP functionality.

3. RESEARCH DESIGN AND EXPERIMENT

A. Synthetic ERP Transaction Dataset

The evaluation uses a synthetic dataset because production purchase-order and shipment records normally contain commercially sensitive supplier, pricing, and operational information. The dataset contains 60,000 chronologically ordered records spanning three simulated years. Each record represents the state of a purchase or shipment at a decision point before delivery outcome is known. No named company, employee, customer, or supplier data is used.

The feature set was designed around fields commonly available in procurement and logistics processes: supplier on-time performance, supplier quality score, planned lead time, order value and quantity, category complexity, source-region risk, transport mode, expedite flag, supplier tenure, price variance, historical exception rate, inventory coverage, demand volatility, distance, contract status, critical-material status, and peak-season status. The binary target indicates whether delivery is late. The simulation deliberately includes nonlinear interactions such as long-distance ocean movements, high-complexity items from lower-quality suppliers, and volatile demand combined with low inventory coverage.

TABLE I: FEATURE GROUPS USED IN THE SIMULATION

Feature group	Representative variables
Supplier	On-time rate, quality, tenure, historical exceptions
Order	Value, quantity, category complexity, contract, criticality
Logistics	Planned lead time, transport mode, distance, region risk
Planning	Inventory coverage, demand volatility, peak-season flag
Control	Expedite flag, price-variance rate

B. Leakage Control and Temporal Split

Only information available before the delivery outcome is used as an input. Actual shipping duration, realized delivery date, final delivery status, and any post-delivery exception code are intentionally excluded. This leakage rule is stricter than a conventional random feature-selection exercise because an ERP model must be evaluated at the exact point at which it would be invoked in production. The first 48,000 records are used for training and the final 12,000 records for testing, preserving chronological order instead of randomly mixing future and past observations.

The late-delivery prevalence is 50.5% in training and 53.6% in the held-out period. Continuous variables are standardized for logistic regression and the neural network; transport mode is one-hot encoded. The same input features and test period are used for all three models.

C. Models and Evaluation

Three classifiers are evaluated. Logistic regression provides an interpretable linear baseline. Random forest represents a strong nonlinear tree-based baseline. The neural model is a multilayer perceptron with two hidden layers of 128 and 64 rectified-linear units and a sigmoid probability output. Training uses mini-batches of 256 observations, early stopping on a validation subset, and a fixed random seed for reproducibility. The experiment is implemented in Python with scikit-learn.

Performance is reported using accuracy, precision, recall, F1-score, and area under the receiver-operating-characteristic curve (ROC-AUC). Because the decision threshold can later be adjusted to the business cost of false negatives and false positives, ROC-AUC is treated as the primary discrimination metric. A 95% bootstrap interval is calculated from 400 resamples of the held-out test set.

4. NEURAL NETWORK-AUGMENTED CLOUD ERP ARCHITECTURE

Figure 1 maps the predictive workflow to a cloud ERP operating model. ERP events remain the authoritative source of business state. A cloud integration layer validates event schemas and decouples the transaction system from downstream services. A feature service combines current document attributes with approved historical context. The neural risk engine returns a probability rather than directly changing the ERP transaction. Workflow, authorization, and human review determine the operational action.

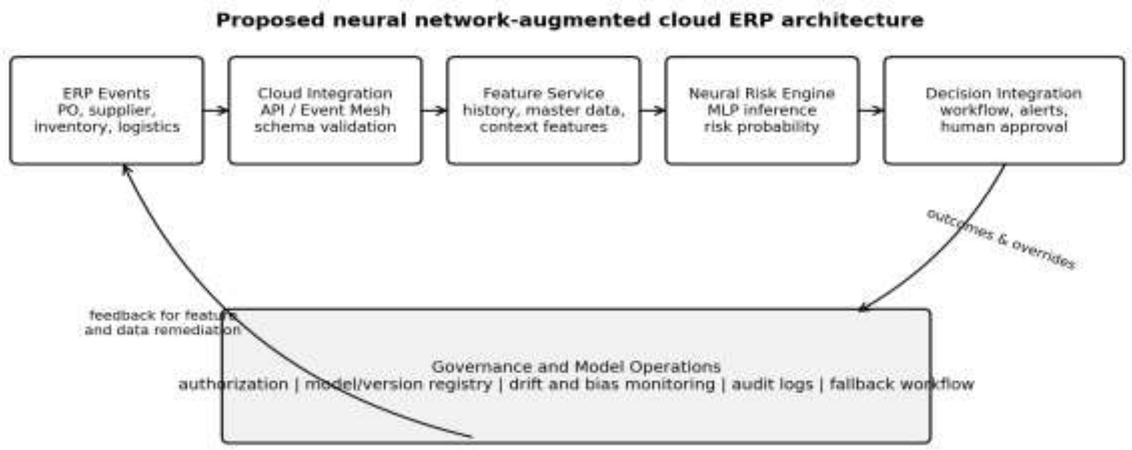

Fig. 1. Proposed neural network-augmented cloud ERP architecture for predictive P2P and supply-chain risk

A practical SAP implementation can map this pattern to SAP S/4HANA business objects, SAP BTP integration services, and an externally deployed model endpoint. SAP Event Mesh can be used where asynchronous business events are available [11], while APIs remain appropriate for synchronous inference or workflow updates. This mapping is intentionally modular: the same architecture can be implemented on another enterprise cloud platform provided that event identity, authorization context, audit traceability, and fallback processing are preserved.

A. Event and Feature Flow

Inference can be triggered when a purchase order is released, when a supplier confirmation changes, or when a material becomes operationally critical. The event payload should contain business keys and a minimal approved feature set rather than an unrestricted copy of the ERP document. The feature service then enriches the event with governed historical measures such as supplier on-time rate or exception frequency. This separation reduces tight coupling and allows feature definitions to be versioned independently from the ERP core.

B. Bounded Decision Authority

The model output is a risk probability, not a posting instruction. Low-risk transactions can remain in the standard process. Medium-risk items can create a buyer work item or supplier follow-up. High-risk items can trigger escalation, but price changes, source changes, purchase-order cancellation, and payment-related decisions remain subject to configured authorization and approval controls. This bounded-autonomy principle keeps the prediction service outside the system-of-record role.

C. Model Operations and Governance

Production use requires a model registry, versioned features, threshold ownership, monitoring for drift, and an audit record linking a risk score to the model version and input snapshot that produced it. The NIST AI Risk Management Framework provides a useful governance structure through its Govern, Map, Measure, and Manage functions [14]. A fallback path must allow the conventional ERP workflow to continue if the model or cloud service is unavailable.

5. RESULTS

A. Predictive Performance

Table II shows the held-out results. The neural network produced the highest ROC-AUC at 0.818, narrowly above random forest at 0.816 and logistic regression at 0.808. Random forest achieved the highest F1-score at 0.759. The neural model therefore improved ranking discrimination, but it did not dominate every threshold-dependent metric. This is a useful result for enterprise design because additional model complexity should be accepted only when the operational benefit justifies the monitoring and support burden.

TABLE II

HELD-OUT TEST PERFORMANCE (N = 12,000)

Model	Acc.	Prec.	Recall	F1	ROC-AUC
Logistic reg.	0.731	0.743	0.761	0.752	0.808
Random forest	0.739	0.751	0.768	0.759	0.816
Neural network	0.733	0.743	0.768	0.755	0.818

The bootstrap ROC-AUC intervals were 0.800–0.816 for logistic regression, 0.809–0.824 for random forest, and 0.810–0.826 for the neural network. The substantial overlap between the latter two intervals indicates that the experiment should not be interpreted as proof that the neural network is universally superior. Instead, it shows that a compact neural model is competitive under a leakage-controlled deployment scenario.

Fig. 2. ROC curves for the three late-delivery classifiers on the chronological test period.

B. Feature Sensitivity

Model-agnostic permutation analysis was used to test which variables most affected neural-network discrimination. Supplier on-time performance produced the largest decrease in ROC-AUC when permuted, followed by transport mode, category complexity, critical-material status, and supplier quality. Historical exception rate, demand volatility, contract status, distance, and source-region risk formed a second tier. The result is directionally consistent with how a planner would reason about delivery exposure: persistent supplier behavior matters, but logistics mode and material context change the meaning of that history.

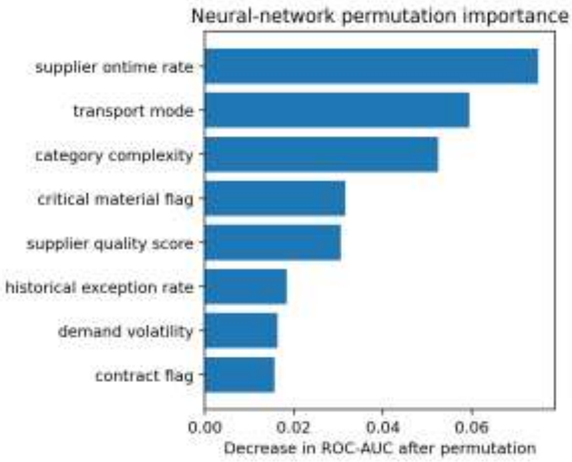

Fig. 3. Top neural-network features by permutation importance on a 3,000-record test subset.

6. INDUSTRY DEPLOYMENT SCENARIO

Consider a manufacturer or retailer that creates thousands of purchase orders each week. Today, buyer attention is often allocated by value, supplier familiarity, or manual exception lists. In the proposed architecture, every released purchase order can be scored at creation and rescored when a relevant business event occurs. A critical material with weak supplier history and long-distance ocean transport may move into a high-risk queue even though no delivery failure has yet occurred. Conversely, a low-risk standard replenishment order can remain untouched.

The business intervention should be chosen separately from the predictive model. A high risk score can initiate supplier confirmation, alternate-source review, inventory reallocation, expediting analysis, or planner escalation. The ERP workflow still enforces who may change a source, quantity, price, or delivery date. This separation makes the predictive service portable across industries while keeping industry-specific decision rights inside the ERP and workflow configuration.

For life sciences, the same pattern can incorporate batch criticality, quality status, and controlled-material constraints. In retail, it can include promotion windows, store or e-commerce demand, and seasonal peaks. In manufacturing, component criticality and production-order dependency can be added. The neural architecture therefore provides a common technical pattern, while the feature definitions and intervention rules remain domain-specific.

The methodology helps to explain from its initial stage; how research problems were identified and to its completion of the study. It includes the Research design, Data source, Research instrument, Population, Sample size and Sampling technique.

7. DISCUSSION

A. What the Experiment Shows

The experiment supports two practical conclusions. First, useful predictive discrimination is possible using only pre-outcome ERP features; post-delivery fields are not necessary to create a meaningful risk ranking. Second, the neural network and random forest are close enough that deployment should not be based on algorithm branding. A simpler model may be preferable when interpretability or supportability is the dominant requirement. A neural model becomes more attractive when the production feature space contains stronger nonlinear interactions, embeddings, sequence history, or high-dimensional contextual data.

B. Threats to Validity

The principal limitation is that the experiment uses synthetic rather than enterprise transaction data. The coefficients and interactions were selected to create realistic ranges and a nontrivial classification problem, but the measured AUC values are properties of the simulation and should not be quoted as expected production performance. The dataset also simplifies supplier networks, route dependencies, currency effects, calendar disruptions, and feedback between procurement decisions and later outcomes. A real deployment would require temporal back-testing on organization-specific data, calibration analysis, cost-sensitive threshold selection, and monitoring after go-live.

External validity is also limited because procurement data distributions differ across retail, life sciences, manufacturing, energy, and public-sector environments. The architecture is more general than the experiment, but the feature service, model, and intervention thresholds must be retrained and validated for each context. Finally, cloud latency and availability were not benchmarked in a production BTP or hyperscaler environment. The study evaluates the logical deployment pattern, not a certified performance service-level agreement.

C. Reproducibility and Responsible Use

The simulation uses a fixed seed, explicit feature definitions, a chronological split, and documented model hyperparameters. Reproducibility is important because enterprise AI evaluation can otherwise be distorted by leakage, repeated test-set tuning, or undocumented preprocessing. The model should be treated as decision support. Supplier risk scores can influence commercial relationships, so organizations should review proxy bias, false-positive burden, override patterns, and whether disadvantaged suppliers are systematically escalated without defensible operational evidence.

8. CONCLUSION

This paper connects neural-network prediction, cloud computing, ERP transaction control, and supply-chain operations in one deployable pattern. A leakage-aware simulation showed that a compact multilayer perceptron can provide competitive late-delivery discrimination using information available before the outcome, achieving an ROC-AUC of 0.818 on the held-out period. Random forest remained equally competitive and produced a slightly better F1-score, reinforcing the need to compare model complexity against operational value.

The larger contribution is architectural. Predictive intelligence should sit beside the ERP core, not replace it. Events and APIs expose controlled business context; cloud services create features and inference; workflow and authorization decide what can happen next; monitoring and human review close the loop. Future work should replace the synthetic experiment with multi-industry longitudinal ERP datasets, compare calibration and decision cost under different intervention budgets, and evaluate sequence models that learn directly from purchase-order and supplier-event histories.

Reference

1. A. J. van Weele, Purchasing and Supply Chain Management, 6th ed. Andover, U.K.: Cengage Learning, 2014.
2. [2] M. Christopher, Logistics and Supply Chain Management, 5th ed. Harlow, U.K.: Pearson, 2016.

3. [3] T. Schoenherr and C. Speier-Pero, “Data science, predictive analytics, and big data in supply chain management: Current state and future potential,” J. Bus. Logist., vol. 36, no. 1, pp. 120–132, 2015, doi: 10.1111/jbl.12082.
4. [4] B. T. Hazen, C. A. Boone, J. D. Ezell, and L. A. Jones-Farmer, “Data quality for data science, predictive analytics, and big data in supply chain management,” Int. J. Prod. Econ., vol. 154, pp. 72–80, 2014, doi: 10.1016/j.ijpe.2014.04.018.
5. [5] M. Seyedan and F. Mafakheri, “Predictive big data analytics for supply chain demand forecasting: Methods, applications, and research opportunities,” J. Big Data, vol. 7, art. 53, 2020, doi: 10.1186/s40537-020-00329-2.
6. [6] K. R. Ahmed, M. E. Ansari, M. N. Ahsan, A. Rohan, M. B. Uddin, and M. A. H. Rivin, “Deep learning framework for interpretable supply chain forecasting using SOM ANN and SHAP,” Sci. Rep., vol. 15, art. 26355, 2025, doi: 10.1038/s41598-025-11510-z.
7. [7] S. Faulkner, R. Zandehshahvar, V. E. Akhlaghi, S. Ouellet, C. Jordan, and P. Van Hentenryck, “Uncertainty-aware delivery delay duration prediction via multi-task deep learning,” arXiv:2602.20271, 2026.
8. [8] A. Katangoori, “An empirical analysis of data-driven supply chain optimization in retail and logistics,” Supply Chain Analytics, vol. 14, art. 100206, 2026, doi: 10.1016/j.sca.2026.100206.
9. [9] F. Constante, F. Silva, and A. Pereira, “DataCo SMART Supply Chain for Big Data Analysis,” Mendeley Data, 2019, doi: 10.17632/8gx2fvg2k6.5.
10. [10] I. Goodfellow, Y. Bengio, and A. Courville, Deep Learning. Cambridge, MA, USA: MIT Press, 2016.
11. [11] SAP SE, “SAP Event Mesh,” SAP Business Technology Platform. [Online]. Available: https://www.sap.com/products/technology-platform/event-mesh.html. Accessed: Aug. 7, 2026.
12. [12] SAP SE, “SAP Business Technology Platform,” SAP Help Portal. [Online]. Available: https://help.sap.com/docs/BTP/65de2977205c403bbc107264b8eccf4b/sap-business-technology-platform. Accessed: Aug. 7, 2026.
13. [13] SAP SE, “Architecture of the Machine Learning Service for Reconcile GR/IR Accounts,” SAP Help Portal. [Online].Available:https://help.sap.com/docs/SAP_S4HANA_CLOUD/0fa84c9d9c634132b7c4abb9ffdd8f06/68bc6c6930634f859c3293d458a8f155.html. Accessed: Aug. 7, 2026.
14. [14] E. Tabassi, Artificial Intelligence Risk Management Framework (AI RMF 1.0), NIST AI 100-1. Gaithersburg, MD, USA: National Institute of Standards and Technology, 2023, doi: 10.6028/NIST.AI.100-1.
15. [15] G. Hohpe and B. Woolf, Enterprise Integration Patterns: Designing, Building, and Deploying Messaging Solutions. Boston, MA, USA: Addison-Wesley, 2003.

Cite this Article:

Manykandaprebou Vaitinadin,"Neural Network Procure-to-Pay and Supply Chain Intelligence -Augmented Cloud ERP Architecture for Predictive"*, Ved International Journal of Arts, Commerce and Technology (VIJACT), ISSN: 3139-1656 (Online), Volume 2, Issue 9, pp. 76-83, Sep 2026.*
Journal URL: https://vijact.com
DOI: https://doi.org/10.65785/vijact.v2i9.96