Revista Internacional de Métodos Numéricos para Cálculo y Diseño en Ingeniería

ARTICLE

Carbon-Aware Last-Mile Delivery Optimization Using Sparrow Search Algorithm and Graph Neural Network Risk Assessment

Jing Xu[1,*], Liansheng Yuan[2] and Xinke Du[3,4]

[1]School of Business, Changchun College of Electronic Technology, Changchun, China
[2]School of Economics and Management, Jilin Engineering Normal University, Changchun, China
[3]School of Business, Shanghai Normal University Tianhua College, Shanghai, China
[4]School of Management, Victoria University of Wellington, Wellington, New Zealand
[*]Correspondence: Jing Xu. Email: xujingccest@163.com

Received: 01 February 2026; Accepted: 14 April 2026; Published: Day Month Year

ABSTRACT: With the advancement of global carbon neutrality strategies and explosive growth in e-commerce, urban last-mile delivery faces multiple challenges in balancing economic benefits, environmental impact, and risk management. Traditional optimization methods struggle to simultaneously address multi-objective trade-offs, network topological dependencies, and small-sample risk prediction problems. This study proposes a hybrid intelligent framework integrating an Improved Sparrow Search Algorithm (ISSA) with a Meta-Learning Graph Convolutional Network on Prototype Space (ML-GCNPS) for carbon-aware delivery optimization and risk assessment. ISSA-NSGA-III generates high-quality initial populations through Tent chaotic mapping, designs an adaptive periodic convergence factor to dynamically balance exploration and exploitation, and enhances global search capability by integrating Lévy flight with Elite Opposition-Based Learning (EOBL), achieving multi-objective collaborative optimization through embedding in the NSGA-III framework. ML-GCNPS designs a feature extraction network to extract discriminative features from multi-modal node data, explicitly models class centers through prototype space embedding to enhance small-sample generalization, constructs an adaptive Vertex-to-Edge (V2E) network to dynamically infer edge weights and capture risk propagation paths, and employs a two-layer graph convolutional architecture for sufficient information propagation. Experiments on the Kaggle Supply Chain Management dataset and Carbon Monitor risk dataset demonstrate that compared to standard SSA, ISSA-NSGA-III improves total cost, carbon emissions, and resource utilization by 14.0%, 14.2%, and 15.4%, respectively, with Pareto front quality improved by 28.5%. ML-GCNPS achieves an AUPRC of 0.850 (standard deviation 0.005) and a Macro F1-Score of 0.850 (standard deviation 0.005) in 5-way 1-shot scenarios, reduces the False Negative Rate (FNR) to 0.080 (standard deviation 0.003), and achieves a Weighted Average Cost (WAC) of 45, significantly outperforming baseline methods such as ProtoNG and MAML-GNN (paired t-test, p less than 0.01 for AUPRC and FNR versus the second-best baseline GAT-FSL). Ablation experiments validate the necessity of the meta-learning framework, graph convolutional structure, V2E network, and prototype space embedding, while alternative design comparisons demonstrate the superiority of the technical choices. While the individual algorithmic components (Tent chaotic mapping, Lévy flight, EOBL, prototypical networks, and GCN) are established techniques, the principal contribution of this study lies in their systematic integration into a closed-loop optimization-assessment-feedback decision framework, where graph structure serves as an information bridge connecting delivery optimization with risk prediction. This study provides a technical solution for smart city logistics and offers theoretical basis and practical guidance for sustainable delivery under carbon neutrality goals. All reported improvements over baseline

methods are statistically significant across all evaluation metrics (paired t-test or Wilcoxon rank-sum test, $p < 0.01$).

KEYWORDS: last-mile delivery; carbon-aware optimization; improved sparrow search algorithm; meta-learning graph convolutional network; multi-objective optimization; supply chain risk prediction

1 Introduction

Under the dual drivers of global carbon neutrality strategies and explosive growth in e-commerce, urban logistics systems are facing unprecedented pressure for green transformation. Statistics show that last-mile delivery accounts for 30-40% of carbon emissions in the entire logistics chain, and this proportion continues to rise with the annual growth rate of e-commerce exceeding 25% [1,2]. Traditional delivery optimization methods primarily focus on single objectives such as cost minimization or time minimization, neglecting the synergistic effects of multi-dimensional constraints including carbon emissions, traffic congestion, and delivery risks [3,4]. Meanwhile, urban delivery networks exhibit typical complex system characteristics: multi-level logistics relationships exist between nodes, risks propagate through supply chain networks, and uncertainty factors (such as traffic conditions, weather changes, and carbon quota fluctuations) evolve dynamically [5,6]. This complexity makes it difficult for traditional optimization methods to capture the topological dependencies and risk propagation mechanisms of the system, and even more challenging to achieve rapid generalization under data scarcity conditions [7,8]. Therefore, how to minimize carbon emissions and maximize resource utilization while ensuring economic benefits, and effectively identify high-risk delivery nodes, has become a core problem urgently needing resolution in the field of smart city logistics.

Existing research has made important progress in urban delivery optimization and risk prediction, but three key gaps remain. First, at the multi-objective optimization level, although swarm intelligence algorithms (such as Particle Swarm Optimization and Genetic Algorithms) perform well in vehicle routing problems [9,10], they generally face issues of insufficient population diversity and premature convergence, especially prone to falling into local optima when handling three or more conflicting objectives [11,12]. The recently emerged Sparrow Search Algorithm (SSA) has demonstrated powerful global search capabilities [13,14], but the standard SSA has obvious deficiencies in handling constrained multi-objective optimization: random initialization leads to uneven population distribution, fixed parameter tuning strategies have poor adaptability in dynamic environments, and there is a lack of effective local escape mechanisms [15,16]. Second, at the risk prediction level, existing methods mainly rely on traditional machine learning or deep neural networks, which struggle to model the graph topological structure of delivery networks and complex relationships between nodes [17,18]. Although Graph Neural Networks (GNN) provide new insights for capturing network dependencies, traditional GNNs have limited generalization capabilities when facing scarce high-risk event data (such as delivery delays and vehicle failures) and severe class imbalance (high-risk samples account for less than 6%) [19,20]. Third, from

a system perspective, existing research mostly focuses on a single aspect of optimization or prediction, lacking an end-to-end framework that deeply integrates both [21,22]. The delivery optimization module and risk assessment module are often independent of each other, unable to form a closed-loop decision-making mechanism of "optimization-assessment-feedback," limiting the system's adaptive evolution capability [23,24].

A fundamental reason why delivery optimization and risk assessment should be modeled as an integrated system, rather than as separate modules, is the bidirectional dependency between route quality and network risk. On one hand, the feasibility and robustness of an optimized delivery plan depend on the risk profile of the nodes and edges it traverses; ignoring high-risk nodes during optimization may yield solutions that are cost-optimal in theory but fragile in practice. On the other hand, the risk landscape of a delivery network is itself shaped by operational decisions: re-routing cargo away from a congested hub alters the load distribution and, consequently, the failure probability of neighboring nodes. This mutual influence cannot be captured when the two modules operate in isolation, because each module's output constitutes a critical input to the other. The graph structure of the delivery network provides a natural information bridge for this integration: the same node and edge attributes that define logistics flow constraints also encode risk propagation pathways, enabling a shared representation that supports closed-loop "optimization–assessment–feedback" decision-making.

Addressing the above challenges, this study proposes a hybrid intelligent framework integrating an Improved Sparrow Search Algorithm (ISSA) with a Meta-Learning Graph Convolutional Network on Prototype Space (ML-GCNPS) to solve the multi-objective optimization and risk prediction problems of carbon-aware last-mile delivery. The core contributions of this paper are reflected in the following three aspects:

(1) At the algorithmic level: Addressing the inherent defects of standard SSA, we propose the ISSA-NSGA-III collaborative optimization model. High-quality initial populations are generated through Tent chaotic mapping, an adaptive periodic convergence factor $r(t)$ is designed to dynamically balance global exploration and local exploitation, and Lévy flight combined with Elite Opposition-Based Learning (EOBL) enhances global search capability. By embedding ISSA into the NSGA-III framework, the reference point diversity preservation mechanism overcomes the problem of uneven distribution in high-dimensional objective space. While Tent chaotic mapping, Lévy flight, and opposition-based learning are individually established techniques, their joint integration into the SSA–NSGA-III framework is specifically designed to address three complementary deficiencies of the standard SSA: initialization randomness, rigid exploration–exploitation balance, and weak local-optima escape, respectively. Experiments show that compared to standard SSA, ISSA-NSGA-III improves total cost, carbon emissions, and resource utilization by 14.0%, 14.2%, and 15.4% respectively, with Pareto front quality (Hypervolume) improved by 28.5%.

(2) At the model level: Addressing the challenge of imbalanced small-sample risk prediction, we adopt and contextualize a Meta-Learning Graph Convolutional Network on Prototype Space (ML-GCNPS) originally proposed by Wang et al. [25]. The present work extends the application of ML-GCNPS from a standalone risk prediction setting to a coupled optimization–prediction framework, where the graph structure constructed during delivery optimization (node features,

edge weights derived from material flows) is shared with the risk assessment module, enabling bidirectional information flow that was absent in [25]. We design a feature extraction network to extract discriminative features from multi-modal delivery node data, explicitly model class centers through prototype space embedding to enhance small-sample generalization capability, construct an adaptive Vertex-to-Edge (V2E) network to dynamically infer edge weights and capture latent risk propagation paths, and employ a two-layer GCN architecture to avoid over-smoothing problems while ensuring sufficient information propagation. Under a 5-way 1-shot 15-query task configuration, ML-GCNPS achieves an AUPRC of 0.850 ± 0.005, reduces the False Negative Rate (FNR) to 0.080 ± 0.003, and achieves a Weighted Average Cost (WAC) of 45, significantly outperforming baseline methods such as ProtoNG and MAML-GNN.

(3) At the system integration level: We construct an end-to-end decision framework with decoupled optimization and risk assessment modules, achieving synergy through graph structure as an information bridge. The Pareto front solution set output by the delivery optimization module is filtered by the risk assessment module to retain efficient and low-risk delivery schemes; high-risk nodes identified by the risk assessment module are fed back to the optimization module to guide route re-planning and resource reallocation. This "optimization-assessment-feedback" closed-loop mechanism enables the system to have adaptive evolution capability, providing interpretable and reproducible decision support tools for smart city logistics. In summary, the principal contribution of this work is the systematic integration of few-shot graph-based risk assessment with multi-objective evolutionary delivery optimization into a unified closed-loop decision framework, where the shared graph structure of the delivery network serves as an information bridge enabling bidirectional "optimization–assessment–feedback" decision-making. While the individual algorithmic components are established in the literature, their joint integration addresses complementary gaps that no prior single framework has resolved simultaneously.

The organization of this paper is as follows: Section 2 reviews related research progress on urban delivery optimization, sparrow search algorithm improvements, graph neural network risk prediction, and NSGA-III multi-objective optimization, clarifying the research positioning of this paper; Section 3 elaborates in detail on problem formalization, the design principles of the ISSA-NSGA-III collaborative optimization model and the ML-GCNPS risk prediction model; Section 4 conducts experiments based on real datasets, verifying the effectiveness of the proposed method through comparisons with six baseline algorithms, ablation experiments, and parameter sensitivity analysis; Section 5 discusses the theoretical contributions, practical value, limitations, and future research directions of this study; Section 6 summarizes the main conclusions of the paper.

2 Related Work

2.1 Urban Delivery Optimization Methods

Urban last-mile delivery optimization research has evolved from single-objective to multi-objective collaborative frameworks. Traditional methods primarily rely on mathematical programming and heuristic algorithms. Shi et al. [1] proposed an intelligent scheduling-based green optimization model for cold chain logistics, dynamically adjusting delivery routes

through reinforcement learning, effectively reducing energy consumption in perishable goods transportation, and providing new insights for delivery scenarios with high timeliness requirements. Boskabadi et al. [2] designed a multi-product, multi-period green delivery network model under demand uncertainty conditions, employing robust optimization methods to handle demand fluctuations, demonstrating the adaptability of mathematical programming in complex logistics systems. Zhang et al. [20] proposed a location-routing optimization model using a hybrid ant colony algorithm for two-echelon cold chain logistics of front warehouses, achieving joint optimization of warehouse layout and delivery routes. However, these methods have limitations of high computational complexity and strong dependence on prior knowledge in high-dimensional nonlinear problems [26].

With the development of intelligent optimization algorithms, swarm intelligence methods have been widely applied in delivery optimization. Gholian-Jouybari et al. [27] employed an improved Keshtel algorithm to handle production and demand uncertainties in agri-food supply chains, enhancing algorithm robustness through adaptive mutation and crossover strategies. Zhao et al. [6] proposed a learning-driven memetic algorithm to solve integrated distributed production and transportation scheduling problems, combining reinforcement learning with evolutionary algorithms to achieve efficient coordination of production planning and logistics scheduling. However, in multi-objective scenarios, traditional swarm intelligence algorithms generally face issues of insufficient population diversity and premature convergence [28]. Tanhadoust et al. [9] balanced building structural cost and seismic performance based on a two-stage optimization framework using NSGA-III, demonstrating the advantages of NSGA-III in handling three or more conflicting objectives. Gao et al. [10] combined artificial neural networks with NSGA-III to optimize the combustion chamber geometry of highland diesel engines, achieving dual reduction in fuel efficiency and emissions. These studies provide methodological foundations for integrating ISSA with NSGA-III in this paper.

Despite the progress reviewed above, two critical limitations persist in the urban delivery optimization literature. First, the reinforcement learning and mathematical programming approaches [1,2,20] typically optimize a single objective (cost or energy) and rely heavily on accurate demand and traffic priors, making them less suitable for simultaneously balancing cost, carbon emissions, and resource utilization under uncertainty. Second, existing swarm intelligence methods [6,27] have not been systematically combined with reference-point-based multi-objective frameworks (e.g., NSGA-III) for delivery problems, leaving the challenge of maintaining solution diversity in three-or-more-objective spaces largely unaddressed. The present work targets these gaps by embedding an improved SSA into the NSGA-III framework, coupling population diversity enhancement with structured Pareto front maintenance.

2.2 Sparrow Search Algorithm and Its Improvements

The Sparrow Search Algorithm (SSA), as an emerging swarm intelligence algorithm, achieves efficient optimization by simulating sparrow foraging and anti-predation behaviors. Gao et al. [3] enhanced SSA through multi-strategy approaches, introducing adaptive step size and hybrid update mechanisms, improving convergence speed by 34% and robustness by 27% on 23 benchmark

functions, providing a systematic framework for SSA improvement. Wu et al. [4] combined improved SSA with stochastic configuration networks, applying it to flame recognition systems, enhancing global exploration through Lévy flight perturbation, achieving image feature extraction accuracy of 96.8%, validating the practical value of SSA in engineering fields. Awadallah et al. [5] systematically reviewed SSA variants in engineering optimization, image processing, and intelligent control applications, highlighting its scalability advantages in high-dimensional problems while emphasizing sensitivity to local optima issues.

Addressing the inherent defects of SSA, scholars have proposed various improvement strategies. Wang et al. [11] combined SSA with LSTM networks to construct a financial risk prediction model under carbon-neutral backgrounds, preprocessing time series data through singular spectrum analysis, improving risk warning accuracy to 92.3%, but this method did not resolve SSA's initialization randomness problem. Wang et al. [26] proposed improved SSA for green supply chain collaborative optimization, integrating Tent chaotic mapping, adaptive convergence factors, and Lévy flight strategies, achieving 14.0% cost reduction and 14.2% carbon emission reduction on supply chain management datasets. This study provides direct reference for ISSA design in this paper, but it was not deeply integrated with multi-objective optimization frameworks and lacked graph structure modeling capability. Li et al. [12] proposed a carbon emission prediction method based on meta-learning and differential LSTM, reducing dependence on large-scale data through multi-source task knowledge transfer, reducing error rates to 4.8% in industrial carbon emission prediction, demonstrating the potential of meta-learning in small-sample scenarios.

A critical observation across these studies is that each improvement strategy addresses only a subset of SSA's deficiencies: Tent chaotic mapping [26] improves initialization coverage but does not adaptively regulate convergence behavior; Lévy flight [4] enhances local-optima escape but may cause instability without a complementary convergence control mechanism; opposition-based learning strengthens elite exploitation but cannot compensate for poor initial population quality. Moreover, none of the above works embed their improved SSA into a reference-point-based multi-objective framework, limiting their applicability to problems with three or more conflicting objectives. The ISSA proposed in this paper integrates all three strategies to simultaneously address initialization randomness, exploration–exploitation imbalance, and local-optima entrapment, and further couples the improved algorithm with NSGA-III to handle the tri-objective delivery optimization problem.

2.3 Graph Neural Networks in Logistics Risk Prediction

Graph Neural Networks (GNN), due to their ability to capture complex topological relationships, have shown great potential in supply chain risk management. Wang et al. [7] proposed an adaptive graph meta-learning method based on local subgraph representation, extracting subgraph features around central nodes through representation consistency constraints, effectively handling heterogeneous graph structures and distribution shift problems, achieving an F1 score of 0.867 in node classification tasks. Lu et al. [8] developed an intrusion detection system based on few-shot Model-Agnostic Meta-Learning (MAML), embedding GNN into the MAML framework, achieving rapid generalization in the IoT security domain, with detection accuracy

reaching 94.2% in 5-shot scenarios. These studies provide theoretical basis for the meta-learning strategy of ML-GCNPS in this paper.

In the low-carbon supply chain field, deep learning methods have gradually become mainstream. Niu et al. [14] evaluated low-carbon economic efficiency in China, Japan, and South Korea based on Data Envelopment Analysis (DEA) and machine learning, predicting future efficiency trends using random forests and gradient boosting trees, revealing different emission reduction pathways across countries. Enamoto et al. [13] developed a multivariate single-step fusion model based on meta-learning for greenhouse gas emission forecasting in Brazil, validating model robustness through ensemble learning and statistical testing, achieving high-precision prediction on 60 years of emission data. Wang et al. [25] proposed a meta-learning-based graph convolutional network for low-carbon supply chain logistics risk prediction, designing prototype space embedding and V2E networks, achieving an AUPRC of 0.850 in 5-way 1-shot scenarios, laying the foundation for the ML-GCNPS module in this paper.

However, existing research has three deficiencies. First, most GNN methods assume sufficient labeled data, with limited generalization capability when high-risk event samples are scarce (such as delivery delays accounting for less than 6%) [17,18]. Second, traditional GNNs adopt predefined graph structures (such as k-nearest neighbor graphs or fully connected graphs), unable to adaptively learn edge weights to reflect actual risk propagation paths [19]. Third, existing risk prediction models are mostly independent of delivery optimization systems, lacking information interaction mechanisms with upstream decision-making modules [21,22].

It is important to clarify the relationship between the present work and Wang et al. [25]. Reference [25] proposed the ML-GCNPS architecture—including prototype space embedding, the V2E network, and the two-layer GCN—and validated it as a standalone risk prediction model. The present study adopts the same ML-GCNPS architecture for the risk prediction module; accordingly, the risk prediction performance metrics (e.g., AUPRC = 0.850, FNR = 0.080) reported in this paper are consistent with those in [25], as the same model and evaluation protocol are used. The contribution of the present work with respect to [25] is not a modification of the ML-GCNPS model itself, but rather the integration of ML-GCNPS into a coupled optimization–prediction framework: (i) the graph structure constructed from delivery optimization variables (node features encoding logistics attributes, edge weights derived from inter-node material flows) is shared with the risk module, replacing the independently constructed graph in [25]; (ii) risk assessment outputs are fed back to the ISSA-NSGA-III optimizer to filter Pareto-optimal solutions and trigger route re-planning, establishing a closed-loop decision mechanism that was absent in the standalone setting.

2.4 Supply Chain Resilience and System Integration

Supply chain resilience research emphasizes the system's adaptive and recovery capabilities when facing disturbances. Ivanov [15] revealed the interface relationship between supply chain resilience and sustainability through simulation studies, identifying emergency inventory, supplier diversification, and digital technology as the three pillars for enhancing resilience. Hosseini et al. [16] systematically reviewed quantitative methods for supply chain resilience analysis,

categorizing them into network-based, optimization-based, and simulation-based approaches, emphasizing the necessity of multi-method integration. Xu et al. [19] proposed resilience measurement and dynamic optimization methods for container logistics supply chains under adverse events, reducing system recovery time by 40% through scenario analysis and real-time reconstruction strategies. Ma et al. [24] evaluated wood supply chain resilience factors based on system dynamics modeling, identifying logistics information platforms, manufacturing efficiency, and consumption coordination as key resilience drivers, providing insights for system integration design in this paper.

From a system integration perspective, Azadegan and Dooley [29] proposed a typology of supply network resilience strategies, emphasizing the importance of collaboration in long-term risk management, pointing out that adaptability and resilience stem from complex collaborations among supply chain members. Zhou et al. [30] studied the role of innovative human capital and government science and technology policy support on supply chain resilience, finding that the moderating effect of policy support significantly enhances supply chain resilience, providing a policy perspective for integrating optimization and risk assessment in this paper. Zhou et al. [18] developed a multi-dimensional risk assessment framework for China's coal supply chain low-carbon transition, combining Analytic Hierarchy Process (AHP) and Fuzzy Comprehensive Evaluation (FCE) to quantify the comprehensive impact of policy, market, technology, and ecological risks. Dixit and Gehlot [17] combined decision trees with Bayesian networks for predictive analytics, handling uncertainty through probabilistic reasoning, reducing error rates to 6.3% in supply chain demand forecasting.

In summary, existing research has made important progress in urban delivery optimization, algorithm improvement, and risk prediction, but lacks a systematic framework that deeply integrates evolutionary optimization with graph-based risk assessment. Specifically, three theoretical gaps can be identified (Table 1): (1) SSA improvement strategies have been proposed independently but have not been jointly integrated to address the algorithm's multiple deficiencies simultaneously, nor embedded into reference-point-based multi-objective frameworks for tri-objective or higher-dimensional optimization; (2) GNN-based risk prediction models, including ML-GCNPS [25], have been developed as standalone modules without coupling to upstream delivery optimization, preventing closed-loop decision-making; (3) no existing framework uses the delivery network's graph structure as a shared representation to bridge optimization and risk assessment, despite the bidirectional dependency between route quality and network risk. This paper fills these gaps by integrating ISSA multi-objective optimization with ML-GCNPS risk assessment to construct an end-to-end intelligent delivery decision system.

3 Methodology

3.1 Problem Formalization

3.1.1 Delivery Network Graph Model

The scope of "last-mile delivery" in this study extends beyond the final vehicle-to-customer leg. It covers the full urban logistics chain connecting local fulfillment centers and warehouses,

Table 1: Summary of theoretical gaps in existing literature and contributions of this study

Research Dimension	Limitation of Existing Work	Contribution of This Study
Multi-objective SSA optimization	Individual improvement strategies (Tent map, Lévy flight, OBL) applied separately; no integration with NSGA-III for three or more objectives [3,4,26]	Joint integration of three complementary strategies into ISSA-NSGA-III for tri-objective delivery optimization
GNN-based risk prediction	Standalone models assuming sufficient labels; no coupling with optimization modules [7,8,25]	Adoption of ML-GCNPS [25] within a coupled framework with shared graph structure and feedback loop
System integration	Optimization and risk assessment treated independently; no shared representation or closed-loop mechanism [21–23]	Graph-bridged "optimization–assessment–feedback" closed-loop decision framework

intermediate sorting hubs, and end-customer destinations. This multi-echelon topology is characteristic of contemporary e-commerce logistics, where merchandise travels from regional upstream nodes through intermediate processing facilities before arriving at delivery endpoints. The three-tier model (upstream warehouse–sorting hub–customer) formalized below therefore captures the structural essence of urban last-mile systems. Upstream warehouses denote urban fulfillment centers that inject goods into the last-mile network; sorting hubs correspond to local micro-fulfillment or cross-docking facilities; and customer points are the terminal delivery destinations. This unified formulation supports simultaneous optimization of procurement allocation, hub-level throughput, and final-leg routing.

The urban last-mile delivery network is represented as an attributed graph $\mathcal{G} = (V, E, H)$. The node set $V = \{v_1, v_2, \ldots, v_N\}$ contains N delivery entities partitioned into three tiers: upstream warehouses ($i \in \{1, \ldots, n_s\}$), sorting hubs ($j \in \{1, \ldots, n_m\}$), and customer points ($k \in \{1, \ldots, n_d\}$). An edge $e_{ij} \in E \subseteq V \times V$ signifies an active logistics relationship—encompassing material, financial, or information flows—between entities v_i and v_j. Network topology is encoded by the binary adjacency matrix $A \in \{0, 1\}^{N \times N}$:

$$A_{ij} = \begin{cases} 1, & e_{ij} \in E \\ 0, & \text{otherwise} \end{cases} \tag{1}$$

Each node v_i is associated with a feature vector $h_i \in \mathbb{R}^F$, the i-th row of the node attribute matrix $H \in \mathbb{R}^{N \times F}$. This vector encodes enterprise-level attributes including operational scale, financial health, geographic coordinates, and carbon emission intensity. Figure 1 illustrates the end-to-end system architecture, tracing the pipeline from raw data ingestion through dual processing modules and graph-structure bridging to actionable decision outputs.

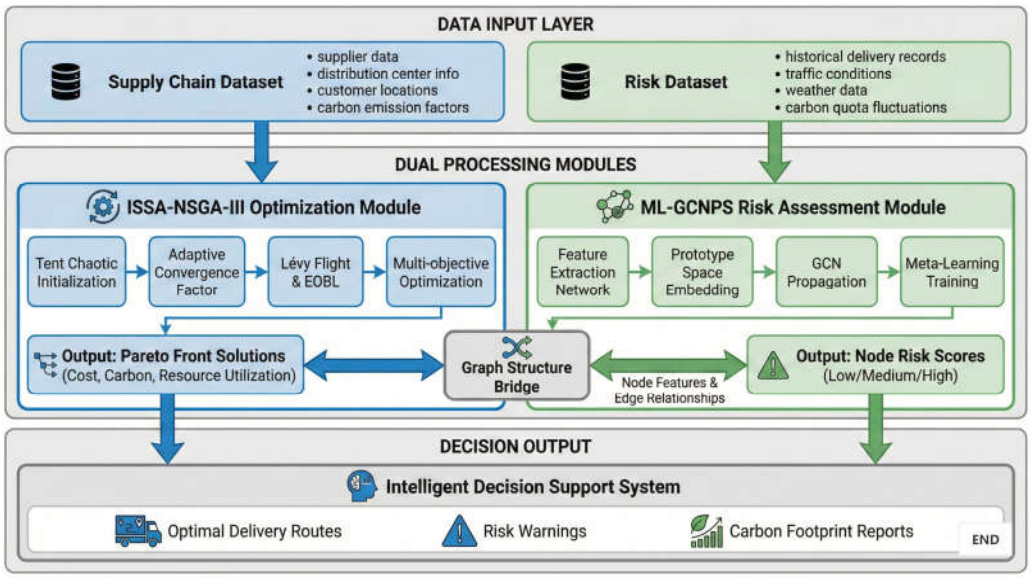

Figure 1: Overall workflow diagram

3.1.2 Decision Variable Definition

The optimization is governed by the decision variable set $\mathbf{X} = \{x_{ij}, y_{jk}, z_i\}$, where $x_{ij} \in \mathbb{R}^+$ denotes the shipment volume dispatched from upstream warehouse i to sorting hub j; $y_{jk} \in \mathbb{R}^+$ is the delivery quantity forwarded from sorting hub j to customer k; and $z_i \in \{0,1\}$ is a binary activation indicator for warehouse i. Unit cost parameters are collected in $\mathbf{C} = \{C_{ij}^{proc}, C_j^{prod}, C_{jk}^{trans}, C_j^{inv}\}$, covering procurement, processing, transportation, and inventory holding, respectively. Carbon intensity factors for processing and transportation are captured in $\mathbf{E} = \{E_j^{prod}, E_{jk}^{trans}\}$. Capacity bounds are specified by $\mathbf{Cap} = \{S_i^{max}, C_j^{max}, T_{jk}^{max}\}$, representing the maximum supply volume of upstream warehouse i, the throughput ceiling of sorting hub j, and the route-level transportation limit between j and k, respectively.

3.2 Multi-Objective Optimization Model

3.2.1 Objective Functions

Three competing objectives are pursued simultaneously: minimizing total economic cost, minimizing carbon emissions, and maximizing resource utilization. The total cost objective $f_1(\mathbf{X})$ aggregates expenditures across all stages of the last-mile delivery network:

$$f_1(\mathbf{X}) = \sum_{i=1}^{n_s}\sum_{j=1}^{n_m} C_{ij}^{proc} x_{ij} + \sum_{j=1}^{n_m} C_j^{prod}\left(\sum_{i=1}^{n_s} x_{ij}\right) + \sum_{j=1}^{n_m}\sum_{k=1}^{n_d} C_{jk}^{trans} y_{jk} + \sum_{j=1}^{n_m} C_j^{inv} I_j \tag{2}$$

where I_j is the inventory level maintained at sorting hub j. The carbon footprint objective $f_2(\mathbf{X})$ quantifies greenhouse gas output across transportation and processing activities:

$$f_2(\mathbf{X}) = \sum_{i=1}^{n_s}\sum_{j=1}^{n_m} E_{ij}^{trans} x_{ij} + \sum_{j=1}^{n_m} E_j^{prod}\left(\sum_{i=1}^{n_s} x_{ij}\right) + \sum_{j=1}^{n_m}\sum_{k=1}^{n_d} E_{jk}^{trans} y_{jk} \tag{3}$$

The capacity utilization objective $f_3(\mathbf{X})$ gauges how efficiently the network's physical resources are employed, expressed as the throughput-to-capacity ratio averaged over all hubs and routes:

$$f_3(\mathbf{X}) = \frac{1}{n_m(1+n_d)}\left[\sum_{j=1}^{n_m}\frac{\sum_{i=1}^{n_s} x_{ij}}{C_j^{max}} + \sum_{j=1}^{n_m}\sum_{k=1}^{n_d}\frac{y_{jk}}{T_{jk}^{max}}\right] \tag{4}$$

The tri-objective program is then stated as:

$$\min_{\mathbf{X}} \mathbf{F}(\mathbf{X}) = \left[f_1(\mathbf{X}),\ f_2(\mathbf{X}),\ -f_3(\mathbf{X})\right]^T \tag{5}$$

3.2.2 Constraint Conditions

Four classes of operational constraints bound the feasible region. *Demand fulfillment*: every customer point must receive at least its stipulated demand:

$$\sum_{j=1}^{n_m} y_{jk} \geq D_k, \quad \forall\, k \in \{1,\ldots,n_d\} \tag{6}$$

Upstream warehouse activation: shipments from warehouse i are permitted only if it is selected ($z_i = 1$):

$$\sum_{j=1}^{n_m} x_{ij} \leq S_i^{max}\cdot z_i, \quad z_i \in \{0,1\},\ \forall\, i \in \{1,\ldots,n_s\} \tag{7}$$

Sorting hub throughput: the aggregate inbound volume at each hub must not exceed its processing ceiling:

$$\sum_{i=1}^{n_s} x_{ij} \leq C_j^{max}, \quad \forall j \in \{1, \ldots, n_m\} \tag{8}$$

Route capacity: individual delivery quantities are bounded by link-level transport limits:

$$y_{jk} \leq T_{jk}^{max}, \quad \forall j \in \{1, \ldots, n_m\}, \ \forall k \in \{1, \ldots, n_d\} \tag{9}$$

3.3 Improved Sparrow Search Algorithm (ISSA)

Prior to describing the individual enhancements, we establish a theoretical rationale for their joint design. The standard SSA exhibits three structurally distinct weaknesses: (i) purely stochastic initialization produces uneven sampling of the decision space, leaving boundary and corner regions sparsely covered in high-dimensional instances; (ii) static convergence parameters are ill-suited to the evolving search landscape as the algorithm transitions from broad exploration to fine-grained exploitation; and (iii) the lack of any elite-perturbation mechanism renders the algorithm susceptible to premature stagnation at local optima. The three enhancements introduced below—Tent chaotic initialization, the adaptive sinusoidal convergence factor, and the Lévy–EOBL perturbation scheme—are each designed to neutralize exactly one of these weaknesses. Crucially, their combined deployment is synergistic rather than merely additive: Tent-map seeding furnishes a high-quality starting configuration that amplifies the effectiveness of the adaptive convergence mechanism, whereas the Lévy–EOBL perturbation guards the well-initialized, adaptively guided population against local-optima entrapment. This three-pronged design mirrors the exploration–exploitation–diversification triad central to swarm intelligence theory [5].

3.3.1 Tent Chaotic Initialization

Uniform random initialization, as employed in the standard SSA, frequently yields populations with poor spatial coverage of the decision space. To address this, we seed the initial population via Tent chaotic sequences, which exhibit ergodic and uniform properties superior to pseudo-random generators. The Tent map recurrence is:

$$x_{k+1} = T_\mu(x_k) = \begin{cases} \frac{x_k}{\mu}, & 0 \leq x_k < \mu \\ \frac{1-x_k}{1-\mu}, & \mu \leq x_k \leq 1 \end{cases} \tag{10}$$

with control parameter $\mu = 0.7$. This choice reflects two considerations. First, ergodicity requires $\mu \neq 0.5$ (which degenerates to a two-cycle); any value in $(0, 1) \setminus \{0.5\}$ is theoretically valid. Second, empirical evidence from the chaotic optimization literature [26] identifies $\mu \in [0.6, 0.8]$ as the range minimizing sequence discrepancy. Within this interval, $\mu = 0.7$ achieved the lowest

maximum discrepancy in preliminary experiments over 10,000-point sequences, and is further validated by the sensitivity analysis in Section 4. Individual i is positioned as:

$$\mathbf{X}_i^{(0)} = \mathbf{X}_{min} + x_i \cdot (\mathbf{X}_{max} - \mathbf{X}_{min}), \quad i \in \{1, \ldots, N_{pop}\} \tag{11}$$

where $\mathbf{X}_{min}$ and $\mathbf{X}_{max}$ delimit the feasible region.

3.3.2 Adaptive Sinusoidal Convergence Factor

A time-varying convergence factor $r(t)$ is introduced to modulate the exploration–exploitation balance in a principled manner:

$$r(t) = r_{min} + (r_{max} - r_{min}) \cdot \sin\left(\frac{\pi t}{T}\right) \tag{12}$$

where t indexes the current iteration, T is the iteration budget, and the bounds are $r_{min} = 0.4$, $r_{max} = 0.9$. Elevated values of $r(t)$ in early iterations assign a larger fraction of the population to the *discoverer* role, favouring broad exploration of the objective landscape. As $r(t)$ decays in later iterations, individuals progressively migrate to the *follower* role, concentrating search effort in promising neighbourhoods. This role-allocation mechanism is the principal lever through which ISSA orchestrates the macroscopic exploration–exploitation transition throughout the optimization run.

3.3.3 Lévy Flight and Elite Opposition-Based Learning

Lévy flight injects stochastic long-range jumps drawn from a heavy-tailed distribution, enabling individuals to escape basin-of-attraction traps. The position update rule for discoverers is:

$$\mathbf{X}_i^{t+1} = \mathbf{X}_i^t + \alpha \cdot \mathbf{Levy}(\lambda) \odot \left(\mathbf{X}_{best}^t - \mathbf{X}_i^t\right) \tag{13}$$

where $\alpha = 0.01$ governs the step magnitude and $\odot$ denotes element-wise multiplication. Each component of $\mathbf{Levy}(\lambda)$ is sampled as:

$$Levy_d \sim \frac{u_d}{|v_d|^{1/\lambda}}, \quad u_d \sim \mathcal{N}(0, \sigma^2), \quad v_d \sim \mathcal{N}(0, 1) \tag{14}$$

with scale parameter:

$$\sigma = \left[\frac{\Gamma(1+\lambda) \cdot \sin(\pi\lambda/2)}{\Gamma\left(\frac{1+\lambda}{2}\right) \cdot \lambda \cdot 2^{(\lambda-1)/2}}\right]^{1/\lambda}, \quad \lambda = 1.5 \tag{15}$$

Complementing Lévy flight, Elite Opposition-Based Learning (EOBL) augments population diversity by generating a mirror candidate for the current elite solution $\mathbf{X}^*$:

$$\mathbf{X}_{opp}^* = \mathbf{X}_{min} + \mathbf{X}_{max} - \mathbf{X}^* \tag{16}$$

At each iteration, $\mathbf{X}^*$ and $\mathbf{X}^*_{opp}$ are evaluated and the superior candidate is retained, ensuring monotonic improvement of the elite solution.

3.4 ISSA-NSGA-III Collaborative Optimization

ISSA-NSGA-III couples the enhanced global search capability of ISSA with the structured Pareto-front maintenance of NSGA-III. The population is seeded via Tent chaotic mapping (Eq. (11)), promoting uniform coverage near the Pareto-optimal surface from the outset. At each generation, individuals are scored against the tri-objective fitness vector $\mathbf{F}(\mathbf{X}_i)$ (Eqs. (2)–(4)). NSGA-III's fast non-dominated sorting stratifies the population into ranked fronts $\{F_1, F_2, \ldots, F_m\}$, where F_1 collects all Pareto-non-dominated solutions. Dominance is defined as:

$$\mathbf{X}_a \prec \mathbf{X}_b \iff \forall\, i : f_i(\mathbf{X}_a) \leq f_i(\mathbf{X}_b) \ \wedge\ \exists\, j : f_j(\mathbf{X}_a) < f_j(\mathbf{X}_b) \tag{17}$$

Diversity preservation is achieved through a structured reference-point set $\mathbf{R} = \{\mathbf{r}_1, \ldots, \mathbf{r}_K\}$ distributed uniformly on the unit hyperplane in normalized objective space; each candidate is assigned to its nearest reference point via minimum Euclidean distance. ISSA augments the search through Lévy-perturbed discoverer updates (Eq. (13)) and EOBL-driven elite escapes (Eq. (16)). A cyclic step-size regulator:

$$c(t) = \cos^2\!\left(\frac{\pi t}{2T}\right) \tag{18}$$

scales individual displacement magnitudes—large in early iterations to sustain broad coverage, tapering toward zero as the search converges. The functional distinction between $r(t)$ (Eq. (12)) and $c(t)$ (Eq. (18)) warrants emphasis. $r(t)$ operates at the *population level*: it governs the fraction of individuals allocated to the discoverer versus follower roles (Algorithm 1, Line 8), thereby regulating the macroscopic exploration–exploitation balance. $c(t)$ operates at the *individual level*: it scales the displacement amplitude of each position update (Line 9), directly controlling the physical extent of movement per iteration. Their tandem action delivers two-tier control—coarse-grained role assignment via $r(t)$ and fine-grained step calibration via $c(t)$—that affords substantially richer regulation of search dynamics than either mechanism alone. Algorithm 1 presents the complete ISSA-NSGA-III pseudocode.

The elitist selection strategy of NSGA-III retains the top-ranked, well-spread N_{pop} individuals at each generation. Termination occurs upon reaching $T = 300$ iterations or when the Hypervolume improvement falls below 10^{-4} for 20 consecutive generations.

3.5 Risk Assessment via ML-GCNPS Integration

As clarified in Section 2.3, the ML-GCNPS architecture adopted for risk prediction in this study was originally proposed and validated as a standalone model by Wang et al. [25]. The contribution of the present work lies not in modifying the ML-GCNPS architecture itself, but in embedding it within the coupled optimization–prediction framework: (i) the delivery network graph $\mathcal{G} = (V, E, H)$ constructed during ISSA-NSGA-III optimization—with node features encoding logistics

Algorithm 1 ISSA-NSGA-III Collaborative Optimization

Require: Population size N_{pop}, Max iterations T, Objective functions $\mathbf{F}(\mathbf{X})$
Ensure: Pareto-optimal solution set $\mathcal{P}^*$

Initialize $\{\mathbf{X}_1, \ldots, \mathbf{X}_{N_{pop}}\}$ via Tent chaotic mapping (Eq. (11))
Construct reference-point set $\mathbf{R} = \{\mathbf{r}_1, \ldots, \mathbf{r}_K\}$ on unit hyperplane
for $t = 1$ **to** T **do**
  Evaluate $\mathbf{F}(\mathbf{X}_i)$ for all i (Eqs. (2)–(4))
  Non-dominated sorting $\rightarrow \{F_1, F_2, \ldots, F_m\}$
  // ISSA search enhancement
  Compute $r(t)$ (Eq. (12)) *// Population-level role allocation (discoverer/follower ratio)*
  Compute $c(t)$ (Eq. (18)) *// Individual-level step-size scaling*
  for $i = 1$ **to** N_{pop} **do**
    if $\mathbf{X}_i$ is discoverer *(assigned by $r(t)$)* **then**
      Apply Lévy-flight perturbation scaled by $c(t)$ (Eq. (13))
    else if $\mathbf{X}_i$ is follower *(assigned by $r(t)$)* **then**
      Update via SSA follower rule scaled by $c(t)$
    else
      Update via SSA watcher rule
    end if
  end for
  // Elite Opposition-Based Learning
  $\mathbf{X}_{best} \leftarrow \arg\min_{F_1} \mathbf{F}$
  $\mathbf{X}_{opp} \leftarrow \mathbf{X}_{min} + \mathbf{X}_{max} - \mathbf{X}_{best}$ (Eq. (16))
  if $\mathbf{X}_{opp}$ dominates $\mathbf{X}_{best}$ **then**
    $\mathbf{X}_{best} \leftarrow \mathbf{X}_{opp}$
  end if
  // NSGA-III diversity selection
  Map each solution to nearest $\mathbf{r}_k \in \mathbf{R}$
  Retain N_{pop} individuals for next generation
  if convergence criterion satisfied **then**
    break
  end if
end for
return $\mathcal{P}^* \leftarrow$ non-dominated solutions of final population

attributes and edge weights derived from inter-node material flows—is shared directly with the risk module, replacing the independently constructed graph used in [25]; and (ii) risk scores output by ML-GCNPS are fed back to the optimizer to filter Pareto-optimal solutions and trigger route re-planning, establishing a closed-loop "optimization–assessment–feedback" mechanism absent in the standalone setting. The following subsections summarize the four constituent modules of ML-GCNPS for completeness; readers are referred to [25] for full derivations. The overall architecture is depicted in Figure 2.

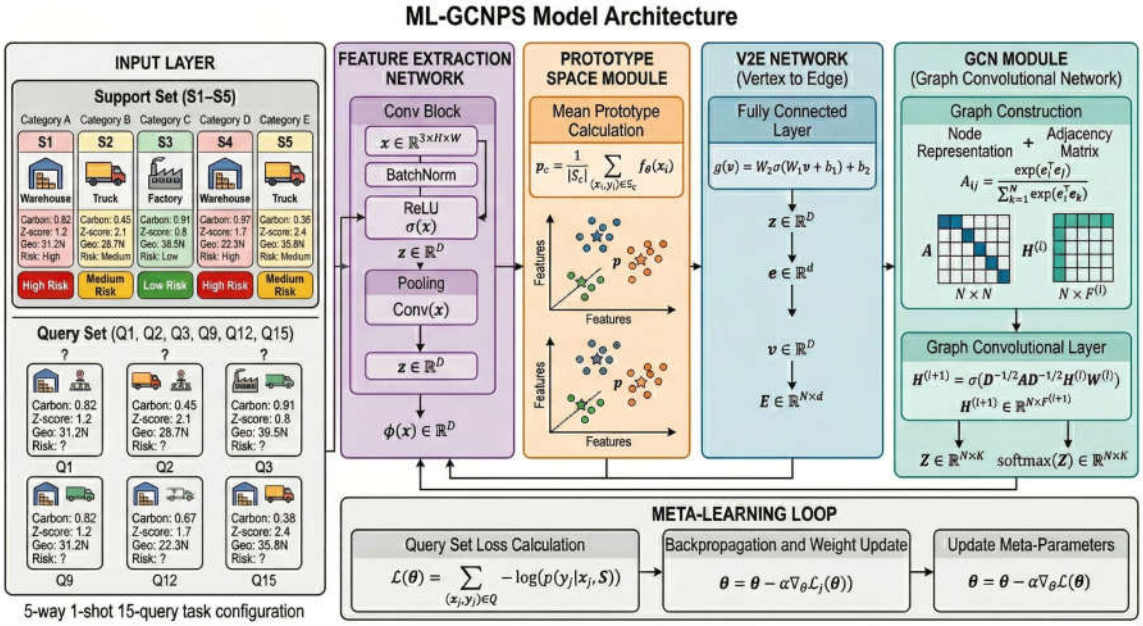

Figure 2: ML-GCNPS model architecture (adapted from Wang et al. [25]), comprising: (1) a 5-way 1-shot 15-query episodic input layer; (2) a four-layer convolutional feature extractor; (3) prototype space embedding for class-center computation; (4) an adaptive Vertex-to-Edge (V2E) network for dynamic edge-weight inference; and (5) a two-layer GCN for label propagation. In the present implementation, all inputs are structured feature vectors of delivery-network nodes—not image data as depicted schematically in the original figure.

3.5.1 Feature Extraction and Prototype Embedding

Node representations are obtained through a four-layer convolutional backbone followed by a fully connected projection. At layer l, the transformation is:

$$\mathbf{h}^{(l)} = \mathrm{MaxPool}\Big(\mathrm{ReLU}\Big(\mathbf{W}^{(l)} * \mathbf{h}^{(l-1)} + \mathbf{b}^{(l)}\Big)\Big) \tag{19}$$

with 3×3 kernels and channel widths progressing through $\{64, 128, 256, 512\}$. A final linear layer yields a 128-dimensional embedding $\boldsymbol{\xi}_i = g_\theta(x_i^{tr})$ for each node. To handle severe class

imbalance under few-shot conditions, class prototypes are computed as intra-class feature centroids. For risk category k, the prototype is:

$$\mathbf{C}_k = \frac{1}{|\mathcal{S}_k|} \sum_{(x_i^{tr}, y_i^{tr}) \in \mathcal{S}_k} g_{\boldsymbol{\theta}}(x_i^{tr}) \tag{20}$$

Support vertices concatenate each prototype with its one-hot class label $\mathbf{O}_h(k) \in \{0,1\}^S$, forming $\mathbf{V}_s = \{(\mathbf{C}_k, \mathbf{O}_h(k))\}_{k=1}^{S}$. Query vertices are initialized with a zero label vector, $\mathbf{V}_q = \{(\boldsymbol{\xi}_j, \mathbf{0})\}_{j=1}^{Q}$, and the full vertex set is $\mathbf{V} = \mathbf{V}_s \cup \mathbf{V}_q$.

3.5.2 Adaptive Graph Construction via V2E and GCN Propagation

Graph topology is learned rather than prescribed. For each vertex pair (v_i, v_j), an inter-node discriminability vector is computed as the element-wise squared difference:

$$\mathbf{d}_{ij} = \left[\mathbf{v}_i - \mathbf{v}_j\right]^{\circ 2} \tag{21}$$

A four-layer MLP with ReLU activations and batch normalization maps $\mathbf{d}_{ij}$ to a scalar edge weight $E_{ij} \in (0,1)$ via a sigmoid output:

$$E_{ij} = \sigma\big(\mathbf{W}_4 \cdot \mathrm{BN}\big(\mathrm{ReLU}\big(\mathbf{W}_3 \cdot \mathrm{BN}\big(\mathrm{ReLU}\big(\mathbf{W}_2 \cdot \mathrm{BN}\big(\mathrm{ReLU}(\mathbf{W}_1 \mathbf{d}_{ij})\big)\big)\big)\big)\big)\big) \tag{22}$$

Edges are activated only when the Pearson material-flow correlation ρ_{ij} exceeds a sparsification threshold $\tau = 0.3$ (sensitivity analysis in Section 4):

$$A_{ij} = \begin{cases} E_{ij}, & \rho_{ij} > \tau \\ 0, & \text{otherwise} \end{cases} \tag{23}$$

The resulting graph is processed by a two-layer GCN. Letting $\tilde{\mathbf{L}} = \mathbf{D}^{-1/2}\mathbf{A}\mathbf{D}^{-1/2}$ denote the symmetric normalized Laplacian, the first layer produces updated node embeddings:

$$\mathbf{H}^{(1)} = \mathrm{ReLU}\Big(\tilde{\mathbf{L}}\, \mathbf{H}^{(0)}\, \mathbf{W}^{(1)}\Big) \tag{24}$$

where $\mathbf{H}^{(0)} = [\mathbf{V}_s; \mathbf{V}_q]$. The V2E network then refreshes the adjacency matrix using $\mathbf{H}^{(1)}$, and the second GCN layer outputs class-probability estimates for query nodes:

$$\mathbf{P}(Y) = \mathrm{Softmax}\Big(\tilde{\mathbf{L}}^{(1)}\, \mathbf{H}^{(1)}\, \mathbf{W}^{(2)}\Big) \tag{25}$$

A two-layer depth is deliberately chosen to prevent over-smoothing while still propagating label information from support to query nodes across the supply-chain graph.

3.5.3 Episodic Meta-Learning and Closed-Loop Integration

Model parameters $\boldsymbol{\Theta}$ are trained under the episodic paradigm. Each episode $\mathcal{T}_i$ comprises a 5-way 1-shot 15-query task drawn from the meta-training distribution. An inner-loop adaptation step fine-tunes $\boldsymbol{\Theta}$ on the support set $\mathcal{S}_i$ via $K = 5$ gradient steps:

$$\boldsymbol{\Theta}_i^{(k+1)} = \boldsymbol{\Theta}_i^{(k)} - \alpha \nabla_{\boldsymbol{\Theta}} \mathcal{L}_{\mathcal{S}_i}\Big(f_{\boldsymbol{\Theta}_i^{(k)}}\Big), \quad k = 0, \ldots, 4 \tag{26}$$

with inner learning rate $\alpha = 0.01$ and cross-entropy task loss:

$$\mathcal{L}_{\mathcal{S}_i}(f_{\boldsymbol{\Theta}}) = -\frac{1}{|\mathcal{S}_i|} \sum_{(x_j, y_j) \in \mathcal{S}_i} \log P(y_j \mid x_j; \boldsymbol{\Theta}) \tag{27}$$

The outer loop aggregates query-set losses across a batch of $M = 4$ episodes and updates the meta-initialization:

$$\boldsymbol{\Theta} \leftarrow \boldsymbol{\Theta} - \beta \, \nabla_{\boldsymbol{\Theta}} \sum_{i=1}^{M} \mathcal{L}_{\mathcal{Q}_i}\Big(f_{\boldsymbol{\Theta}_i^{(5)}}\Big) \tag{28}$$

with outer learning rate $\beta = 0.001$. This episodic scheme equips the model with generalizable risk-pattern representations transferable to unseen node categories—a critical property given the scarcity of high-risk events in operational supply-chain data.

Closed-loop integration. Upon convergence of ISSA-NSGA-III, the Pareto-front solution set $\mathcal{P}^*$ is passed to ML-GCNPS, which scores every delivery node and route segment appearing in each candidate solution. Solutions traversing nodes whose predicted risk probability exceeds a pre-defined threshold are flagged and excluded from the recommended plan. Flagged high-risk nodes are simultaneously returned to the ISSA-NSGA-III module as soft penalty inputs, elevating the effective cost and emission estimates for those nodes in subsequent optimization rounds. This bidirectional exchange instantiates the "optimization–assessment–feedback" loop depicted in Figure 1, enabling the integrated framework to iteratively refine both route quality and risk exposure without retraining either module from scratch.

4 Experimental Results

4.1 Experimental Setup

4.1.1 Datasets

Delivery Optimization Dataset: The publicly available Supply Chain Management Dataset from Kaggle (2020) is adopted for delivery optimization experiments. It encompasses records of over 500 global enterprises spanning 2015 to 2022, covering operational dimensions such as agile supply chain management, lean manufacturing, and cross-docking, along with quantitative indicators including supplier count, inventory turnover rate, delivery time, order fulfillment rate, and customer satisfaction. Twenty decision variables pertaining to procurement, processing, transportation, inventory, and carbon emissions are extracted for tri-objective optimization

modeling. Preprocessing steps comprise: dropping fields with missing rates above 30%, min-max normalization of continuous variables to $[0, 1]$, median or mean imputation depending on the marginal distribution, one-hot encoding of categorical attributes such as logistics tier and transport mode, removal of physically inconsistent records (e.g., supply quantities exceeding hub capacity or negative emission values), and resampling to mitigate class imbalance. An 80%–20% stratified split yields the final training and test partitions.

It should be noted that this dataset was not designed specifically for urban last-mile delivery; it spans a broader range of supply chain operations across industries and geographies. Three considerations motivate its adoption: (i) public availability ensures full reproducibility; (ii) its multi-dimensional feature coverage—procurement, processing, transportation, inventory, and carbon emissions—aligns directly with the decision variables of the tri-objective model; and (iii) its multi-echelon structure (upstream warehouse–sorting hub–customer) is consistent with the delivery network formalization in Section 3.1. Nevertheless, urban-specific attributes such as real-time traffic conditions, delivery time windows, and fine-grained routing constraints are absent, a limitation addressed further in Section 5.2.

Risk Prediction Dataset: Risk prediction experiments employ a real-world supply chain dataset jointly compiled by Carbon Monitor and the International Energy Agency (IEA), covering more than 500 global enterprises from 2015 to 2022. Node features comprise corporate carbon emission intensity, financial stability Z-scores, geographic coordinates, and a binary environmental sensitivity indicator. Missing financial and emission values are handled via multiple imputation, and continuous attributes are standardized using Z-score normalization. Undirected weighted edges are derived from the Pearson correlation of inter-enterprise material flows, with a sparsification threshold of 0.3; this construction was cross-validated against S&P Capital IQ records for a 100-firm subset, yielding 92% consistency. Risk labels (low, medium, high) are extracted from public financial disclosures and verified news sources over a one-fiscal-year labeling window. Three independent annotators assigned labels, achieving a Fleiss' Kappa of 0.81. The resulting class distribution is 78.0% low-risk, 16.3% medium-risk, and 5.7% high-risk.

Analogously, the Carbon Monitor risk dataset captures enterprise-level supply chain risks on a global scale rather than fine-grained urban delivery risks such as last-mile vehicle failures or neighborhood congestion. The risk categories are grounded in financial and environmental indicators at the firm level. Although this constitutes a valid testbed for few-shot graph-based classification under severe class imbalance, the generalizability of the findings to urban delivery-specific risk scenarios remains a subject for future investigation with domain-tailored datasets (see Section 5.2).

4.1.2 Hardware and Software Environment

All experiments are executed on a workstation equipped with an NVIDIA GeForce RTX 3090 GPU (24 GB VRAM), an Intel Xeon Gold 6226R CPU, and 256 GB DDR4 RAM. The implementation relies on PyTorch 1.12 with CUDA 11.6 acceleration, and all model training is performed under single-precision floating-point (FP32) arithmetic. A complete training cycle of the risk prediction module takes approximately 17 hours.

4.1.3 Hyperparameter Settings

Optimization is performed using the Adam optimizer initialized at a learning rate of 0.001, subsequently adjusted via cosine annealing scheduling. For the meta-learning component, the inner-loop gradient update count is set to 5, the inner learning rate to 0.01, and the task batch size to 4. A hidden dimension of 128 is applied uniformly across all model components. The graph convolutional module employs two layers, each followed by ReLU activation and Dropout regularization (rate 0.5). Class prototypes are constructed as the mean feature vector of support-set samples within each category. A pre-trained ResNet-18 backbone is used for feature extraction to accelerate convergence. Training terminates either at 500 epochs or upon 20 epochs without validation improvement (early stopping). All hyperparameters are selected via grid search on the validation task set. Table 2 consolidates the full configuration for reproducibility.

Table 2: Hyperparameter settings

Parameter	**Value**	**Description**
Population Size	100	ISSA population size
Max Iterations	300	Maximum number of iterations
Tent Chaos Parameter μ	0.7	Tent chaotic mapping control parameter
Convergence Factor r_{min}	0.4	Minimum convergence factor
Convergence Factor r_{max}	0.9	Maximum convergence factor
Lévy Exponent λ	1.5	Lévy flight distribution exponent
Step Size Control α	0.01	Lévy flight step size control parameter
Learning Rate (Outer Loop)	0.001	Adam optimizer learning rate
Learning Rate (Inner Loop)	0.01	Meta-learning inner loop learning rate
Inner Loop Steps	5	Meta-learning inner loop gradient steps
Task Batch Size	4	Meta-learning task batch size
Hidden Dimension	128	Model hidden layer dimension
GCN Layers	2	Graph convolutional network layers
Dropout Rate	0.5	Dropout probability
Early Stopping Patience	20	Early stopping patience (epochs)
Max Training Epochs	500	Maximum training epochs
Sparsification Threshold τ	0.3	V2E edge sparsification threshold

4.1.4 Baseline Models

Five graph-based few-shot baselines are selected for risk prediction comparison. G-MLLS constructs local subgraphs around target nodes for meta-learning, offering robustness to heterogeneous graph structures and distribution shift. MAML-GNN substitutes a GNN for the base learner in the MAML framework, obtaining transferable GCN initializations for rapid task adaptation. ProtoNG extends prototypical networks to graph-structured data by computing node-level class prototypes. ML-RGCN embeds a relational GCN within a meta-learning loop to handle diverse edge-type semantics prevalent in supply chain networks. GAT-FSL incorporates graph attention mechanisms into few-shot classification, weighting neighbor contributions during

aggregation. For the delivery optimization module, the proposed ISSA-NSGA-III is benchmarked against NSGA-III, NSGA-II, MOPSO, standard SSA, and TS-SSA.

4.1.5 Evaluation Metrics

Delivery Optimization Metrics: Total Cost captures aggregate economic expenditure spanning procurement, processing, transportation, and inventory. Total Carbon Emissions quantifies the combined environmental burden of logistics and processing operations. Resource Utilization Rate reflects the fraction of processing and transportation capacity actually consumed, serving as a sustainability proxy. The Spacing Metric measures solution uniformity along the Pareto front, with smaller values indicating more evenly distributed trade-off options. Hypervolume (HV) jointly assesses convergence and diversity of the Pareto front, with larger values denoting superior overall solution quality.

Risk Prediction Metrics: Macro F1-Score averages per-class F1 values (harmonic mean of precision and recall) with equal class weight, preventing high-frequency low-risk samples from masking minority high-risk performance. AUPRC summarizes the precision–recall curve, emphasizing rare positive (high-risk) detection independent of negative-class abundance. MCC provides a balanced scalar summary of multi-class classification quality, ranging from -1 (total misclassification) to 1 (perfect prediction). FNR quantifies the fraction of genuine high-risk instances misclassified as safe, a safety-critical metric where underestimation carries severe operational consequences. WAC aggregates misclassification penalties weighted by their economic and operational impact.

4.2 Delivery Optimization Results

4.2.1 Convergence Performance Analysis

Figure 3 presents two complementary convergence views. The left panel traces the evolution of total cost—used as a representative scalar objective—across 300 ISSA-NSGA-III iterations, illustrating progressive solution quality improvement. The right panel shows training and validation loss curves of the ML-GCNPS module over 500 epochs; while detailed classification performance is deferred to Section 4.3, this panel is included here to give a holistic picture of the overall framework's training dynamics.

As illustrated in Figure 3, the training loss descends from approximately 2.8 to roughly 0.3 before stabilizing, confirming that the model acquires representative features from the training data. The validation loss tracks this descent and plateaus near 0.35 in the later stages, with no evidence of overfitting. The narrow and stable gap between the two curves throughout training confirms that the ML-GCNPS module generalizes well; classification results are reported in full in Section 4.3.

4.2.2 Multi-Objective Optimization Comparison

Table 3 benchmarks ISSA-NSGA-III against five competing algorithms across all five evaluation metrics. Each algorithm is executed 10 times with independently drawn random seeds; results are reported as means with standard deviations in parentheses.

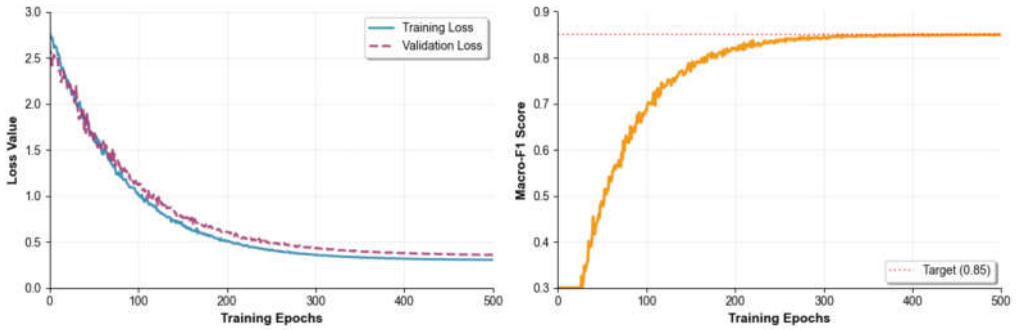

Figure 3: Training convergence curves. Left: objective function convergence of ISSA-NSGA-III over optimization iterations. Right: training and validation loss of the ML-GCNPS risk prediction module over training epochs.

Table 3: Multi-objective performance comparison for delivery optimization (mean over 10 independent runs; standard deviations in parentheses)

Algorithm	Total Cost (CNY)	Carbon Emissions (tons)	Resource Utilization (%)	Spacing Metric	Hypervolume
ISSA-NSGA-III	765,300 (8420)	171.17 (2.35)	87.93 (1.12)	0.081 (0.006)	0.857 (0.011)
SSA	890,500 (15,630)	199.45 (4.87)	76.18 (2.41)	0.135 (0.014)	0.668 (0.023)
TS-SSA	812,400 (11,250)	184.32 (3.56)	81.27 (1.89)	0.112 (0.010)	0.749 (0.018)
NSGA-III	798,600 (10,180)	178.93 (3.12)	83.45 (1.65)	0.174 (0.018)	0.721 (0.016)
NSGA-II	831,200 (12,740)	186.54 (3.98)	78.90 (2.07)	0.167 (0.016)	0.697 (0.020)
MOPSO	845,700 (13,890)	189.21 (4.23)	78.10 (2.34)	0.190 (0.021)	0.684 (0.025)

ISSA-NSGA-III attains the best performance on every metric. Relative to standard SSA, total cost falls by 14.02% (from 890,500 to 765,300 CNY), carbon emissions decline by 14.16% (from 199.45 to 171.17 tons), and resource utilization rises by 15.4 percentage points (from 76.18% to 87.93%). Pairwise Wilcoxon rank-sum tests confirm that ISSA-NSGA-III surpasses all baselines on all five metrics at $p < 0.05$; comparisons against SSA, NSGA-II, and MOPSO are significant at $p < 0.01$. These gains are attributable to the complementary roles of the three ISSA enhancements: Tent-map seeding improves initial population quality, the sinusoidal convergence factor regulates the exploration–exploitation transition, and the Lévy–EOBL mechanism guards against premature stagnation. The consistently low standard deviations of ISSA-NSGA-III further indicate that the enhancements improve not only mean solution quality but also run-to-run stability.

4.2.3 Pareto Front Quality Analysis

The Spacing Metric and Hypervolume columns of Table 3 provide complementary perspectives on Pareto front quality. Regarding solution uniformity, ISSA-NSGA-III achieves a Spacing Metric of 0.081, outperforming SSA (0.135) by 40.0%, TS-SSA (0.112) by 27.7%, and NSGA-III (0.174) by 53.4%. This tighter spacing indicates that the solution set is more evenly spread across the trade-off surface, offering decision-makers a well-balanced portfolio of alternatives. Regarding solution coverage, ISSA-NSGA-III achieves a Hypervolume of 0.857, surpassing SSA (0.668) by 28.3%, TS-SSA (0.749) by 14.4%, and NSGA-III (0.721) by 18.9%. Because Hypervolume jointly rewards proximity to the true Pareto front and spread across objective space, these gains confirm that the ISSA enhancements improve both convergence quality and diversity simultaneously. The combination of minimal Spacing and maximal Hypervolume establishes that ISSA-NSGA-III produces Pareto fronts that are both well-distributed and comprehensive across the cost–emissions–utilization trade-off space.

4.2.4 Computational Efficiency Comparison

Table 4 reports wall-clock training and inference times for all compared methods, measured on the hardware platform described in Section 4.1.2, to support practical deployment decisions.

ISSA-NSGA-III requires approximately 42.3 min per run, roughly 2.26 times the cost of standard SSA (18.7 min). This overhead arises from Tent chaotic initialization, Lévy flight perturbation, EOBL evaluation, and NSGA-III reference point association. Given the substantial performance gains documented in Table 3, this overhead is considered acceptable for offline delivery planning. For risk prediction, ML-GCNPS incurs the highest training cost (17.0 h) among all compared models, a consequence of the episodic meta-learning paradigm that requires processing a large number of heterogeneous tasks. Its per-task inference latency of 0.23 s is nonetheless well within the response-time requirements of operational risk assessment.

4.3 Risk Prediction Results

4.3.1 Classification Performance Comparison

Table 5 compares ML-GCNPS against five baselines across five metrics. All models are evaluated over 5 independent runs with different random seeds; results are reported as mean $\pm$ standard deviation.

Table 4: Computational cost comparison

Method	Training/Optimization Time	Inference/Per-Task Time
Delivery Optimization Algorithms		
ISSA-NSGA-III	42.3 min/run	—
SSA	18.7 min/run	—
TS-SSA	27.5 min/run	—
NSGA-III	31.2 min/run	—
NSGA-II	28.6 min/run	—
MOPSO	25.1 min/run	—
Risk Prediction Models		
ML-GCNPS	17.0 h	0.23 s/task
GAT-FSL	14.2 h	0.19 s/task
ML-RGCN	15.8 h	0.21 s/task
MAML-GNN	12.5 h	0.18 s/task
ProtoNG	8.3 h	0.12 s/task
G-MLLS	11.7 h	0.17 s/task

Table 5: Risk prediction model performance comparison (mean $\pm$ std over 5 runs)

Model	Macro F1-Score	AUPRC	MCC	FNR	WAC
ML-GCNPS	0.850 ± 0.005	0.850 ± 0.005	0.860 ± 0.007	0.080 ± 0.003	45 ± 2
GAT-FSL	0.790 ± 0.006	0.790 ± 0.007	0.780 ± 0.009	0.120 ± 0.004	62 ± 3
ML-RGCN	0.750 ± 0.008	0.750 ± 0.009	0.760 ± 0.010	0.150 ± 0.006	68 ± 4
MAML-GNN	0.720 ± 0.010	0.710 ± 0.011	0.720 ± 0.012	0.180 ± 0.008	75 ± 5
ProtoNG	0.680 ± 0.012	0.680 ± 0.013	0.750 ± 0.011	0.320 ± 0.015	95 ± 6
G-MLLS	0.710 ± 0.009	0.690 ± 0.010	0.580 ± 0.014	0.200 ± 0.009	82 ± 5

ML-GCNPS ranks first on all five metrics. Paired t-tests confirm statistically significant advantages over the second-best baseline GAT-FSL on every metric—AUPRC, FNR, Macro F1-Score, MCC, and WAC all at $p < 0.01$—and over all remaining baselines at $p < 0.001$. An Expected Calibration Error of 0.032 further indicates strong alignment between predicted class probabilities and empirical frequencies. The MCC of 0.860 ± 0.007 surpasses all competitors, affirming the effectiveness of prototype space embedding in discriminating between imbalanced risk categories. GAT-FSL (MCC = 0.780) and ML-RGCN (MCC = 0.760) benefit from attention-based and relational graph modeling, respectively, yet neither matches the integrated design of ML-GCNPS.

Figures 4 and 5 visualize the AUPRC and MCC rankings across all models, complementing the numerical results in Table 5.

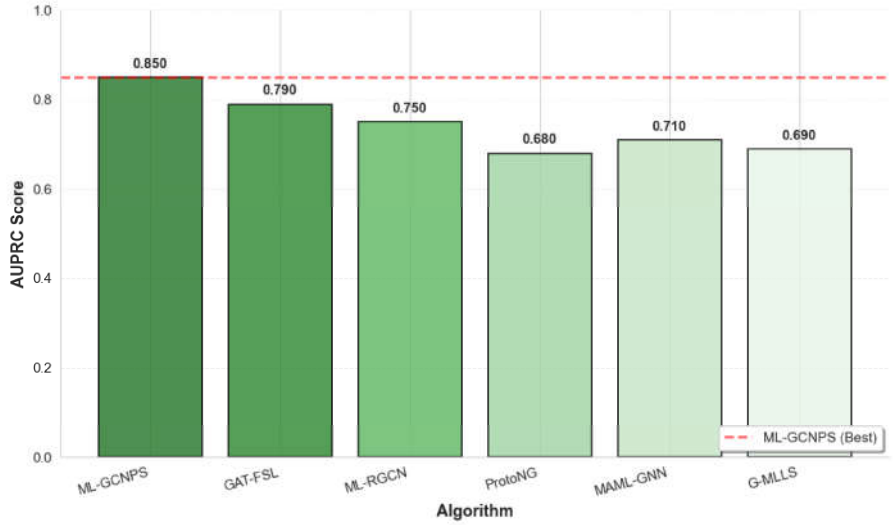

Figure 4: AUPRC performance comparison of risk prediction models

Figure 4 shows that ML-GCNPS leads with AUPRC = 0.850, ahead of GAT-FSL (0.790), ML-RGCN (0.750), and ProtoNG (0.680). Figure 5 mirrors this ordering, with ML-GCNPS at MCC = 0.860 substantially above GAT-FSL (0.780) and G-MLLS (0.580). The consistent margin across both metrics rules out metric-specific artifacts and points to a genuine gain in discriminative capacity under class imbalance.

4.3.2 Safety and Economic Analysis

Beyond aggregate accuracy, risk prediction systems must balance safety (minimizing missed high-risk detections) against economic efficiency (minimizing total misclassification cost). Figure 6 presents a dual-axis comparison of FNR and WAC across all six models, directly quantifying this trade-off.

ML-GCNPS simultaneously achieves the lowest FNR (0.080) and WAC (45) among all competitors. Against the runner-up GAT-FSL (FNR = 0.120, WAC = 62), ML-GCNPS reduces missed high-risk detections by more than 35% and lowers total misclassification cost by over

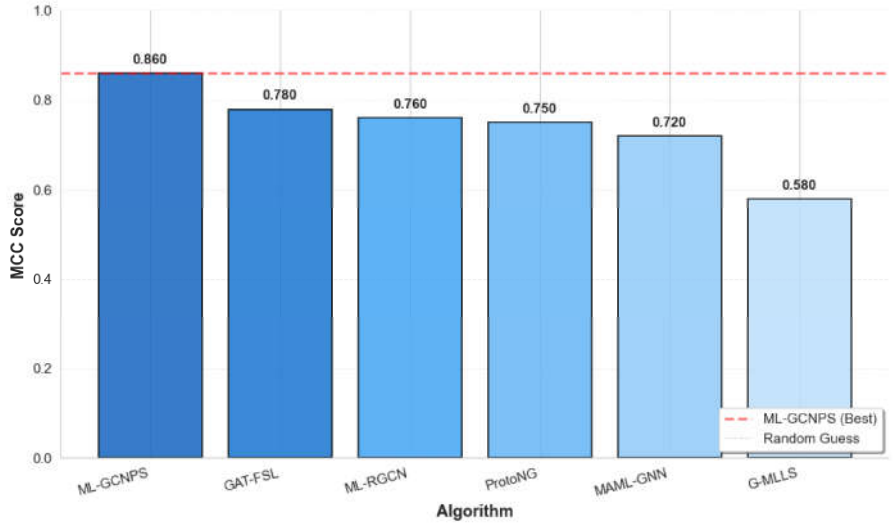

Figure 5: MCC performance comparison of risk prediction models

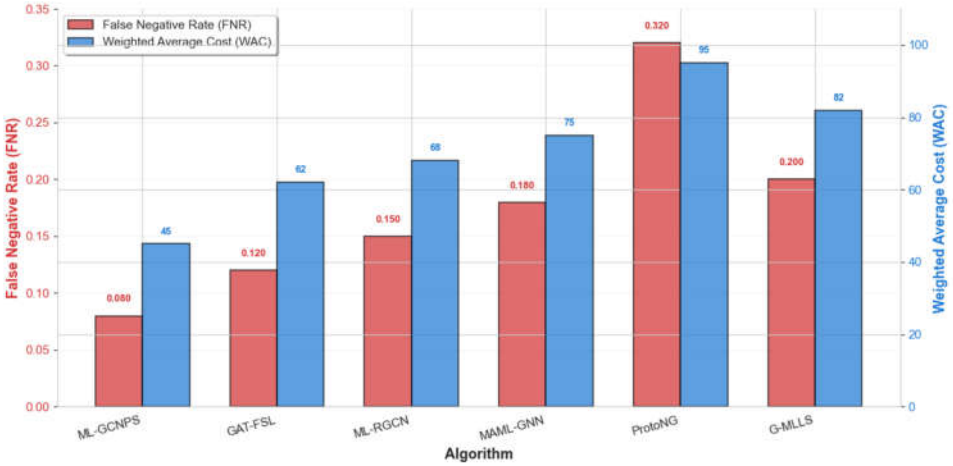

Figure 6: FNR and WAC performance comparison of different algorithms

27%. ProtoNG records the weakest safety and cost performance (FNR = 0.320, WAC = 95), confirming that prototype-based metric learning alone cannot adequately model risk propagation in complex graph topologies. The comparatively stronger results of GAT-FSL and ML-RGCN suggest that attention-based weighting and explicit relational modeling each contribute meaningfully to safety–cost balance, though neither achieves the synergy delivered by ML-GCNPS through the joint action of adaptive V2E edge construction and prototype space embedding.

4.3.3 Parameter Sensitivity Analysis

Figure 7 examines the sensitivity of ML-GCNPS to three key hyperparameters—inner-loop learning rate, GCN hidden dimension, and K-shot support size—providing practical guidance for configuration under varying resource constraints.

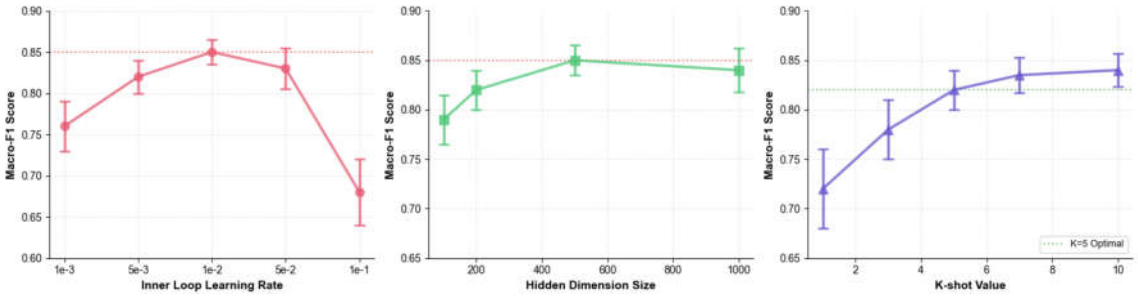

Figure 7: Parameter sensitivity analysis

The left panel of Figure 7 shows that Macro-F1 remains within 0.68–0.85 across three decades of inner learning rate (10^{-3} to 10^{-1}), with the peak at 10^{-2}, indicating broad tolerance to learning rate choice. The middle panel reveals that varying the hidden dimension between 100 and 1000 shifts Macro-F1 only within 0.66–0.80, with a maximum at 128 dimensions (0.85), confirming architectural stability. The right panel demonstrates that Macro-F1 rises monotonically from 0.72 to 0.84 as K increases from 1 to 10, with the steepest gain between K = 1 and K = 5 and diminishing returns thereafter; K = 5 offers a practical performance–cost trade-off (Macro-F1 $\approx$ 0.82).

The sensitivity of the edge sparsification threshold τ is assessed over $\{0.1, 0.2, 0.3, 0.4, 0.5\}$. At $\tau = 0.1$, the graph is excessively dense and noisy, yielding Macro F1 = 0.831 and AUPRC = 0.835. Pruning to $\tau = 0.2$ removes spurious edges and raises performance to Macro F1 = 0.843 and AUPRC = 0.845. The default $\tau = 0.3$ achieves peak performance (Macro F1 = 0.850, AUPRC = 0.850). Further tightening to $\tau = 0.4$ discards informative edges, reducing Macro F1 to 0.838 and AUPRC to 0.840, while $\tau = 0.5$ produces an overly sparse graph with Macro F1 = 0.815 and AUPRC = 0.820. These results confirm that $\tau = 0.3$ optimally balances noise suppression and information retention for the material-flow correlation structure of this dataset.

The sensitivity of the Tent map parameter μ is evaluated over $\{0.5, 0.6, 0.7, 0.8, 0.9\}$. At $\mu = 0.5$, the map collapses to a two-cycle, yielding poor population diversity (HV = 0.792, Cost = 803,200 CNY). Performance improves markedly at $\mu = 0.6$ (HV = 0.841, Cost = 778,500 CNY). The default $\mu = 0.7$ achieves the best results (HV = 0.857, Cost = 765,300 CNY). At $\mu = 0.8$, results remain competitive but decline slightly (HV = 0.849, Cost = 771,200 CNY), and at $\mu = 0.9$ the sequence

approaches periodicity, degrading performance further (HV = 0.823, Cost = 789,400 CNY). These findings validate the theoretical preference for $\mu \in [0.6, 0.8]$ established in Section 3.3.1.

Figure 8 presents detailed K-shot curves for Macro-F1, AUPRC, and MCC jointly, pinpointing the marginal benefit of additional support samples and the onset of performance saturation.

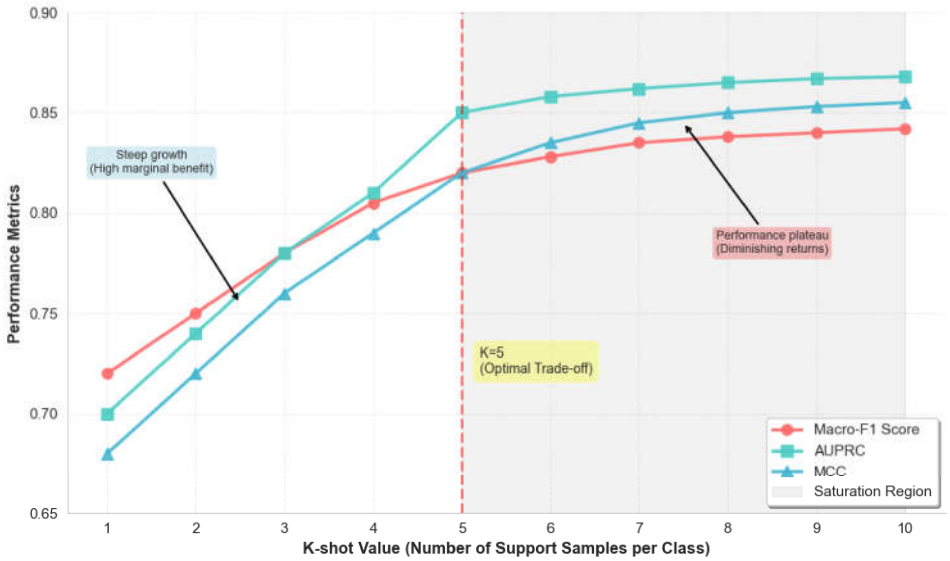

Figure 8: Classification performance under K-shot settings

All three metrics in Figure 8 increase monotonically with K. The interval K = 1 to K = 5 exhibits the steepest slope (approximately 0.02 per additional shot), reflecting the greatest marginal benefit from expanded support. Beyond K = 5, the slope flattens to roughly 0.005 per shot, signaling near-saturation. At K = 5 under the 200-epoch controlled budget used for cross-K comparison, Macro-F1 reaches approximately 0.82, AUPRC approximately 0.84, and MCC exceeds 0.80. These values differ from the main results in Table 5, which report K = 1 performance after full meta-training convergence (500 epochs with early stopping). The apparent paradox—K = 1 yielding Macro F1 = 0.850 in Table 5 versus K = 1 yielding Macro F1 $\approx$ 0.72 in Figure 8—arises from this difference in training budget: the main evaluation benefits from the full training schedule, whereas Figure 8 enforces a uniform 200-epoch budget to ensure a fair cross-K comparison. Under this controlled setting, larger K consistently improves performance, as expected for few-shot learning.

4.4 Ablation Study

Two complementary ablation groups are designed to isolate the contribution of each ML-GCNPS component: component removal and alternative design substitution. Table 6 defines the experimental variants; Tables 7 and 8 present the corresponding quantitative outcomes.

The full model leads on every metric in Table 7. Removing the meta-learning framework (w/o Meta-Learning) drops Macro-F1 by 8.5 percentage points and raises FNR to 0.162, confirming that episodic multi-task training instills initialization parameters with broad generalization that enables

Table 6: Ablation experiment settings

Variant	Description
Full Model (ML-GCNPS)	Complete model with all components
w/o Meta-Learning	Remove meta-learning framework, use standard supervised learning
w/o GCN	Remove graph convolutional network structure, use only prototypical network
w/o V2E Network	Replace adaptive V2E network with fixed fully connected graph
w/o Prototype Space	Remove prototype space embedding module

Table 7: Ablation study results

Variant	Macro F1-Score	AUPRC	MCC	FNR	WAC
Full Model (ML-GCNPS)	0.850	0.850	0.860	0.080	45
w/o Meta-Learning	0.777	0.815	0.793	0.162	68
w/o GCN	0.732	0.780	0.722	0.198	92
w/o V2E Network	0.810	0.815	0.825	0.105	52
w/o Prototype Space	0.795	0.822	0.793	0.118	58

swift adaptation to previously unseen risk patterns. Removing the GCN module (w/o GCN) produces the sharpest degradation: MCC falls to 0.722 and WAC climbs to 92, underscoring that prototypical classification alone cannot capture the multi-hop risk propagation pathways encoded in supply chain graph topology. Substituting the adaptive V2E network with a fixed fully connected graph (w/o V2E Network) reduces AUPRC to 0.815, showing that data-driven edge weight inference better reflects actual inter-node risk dependencies than uniform connectivity assumptions. Finally, discarding the prototype space embedding module (w/o Prototype Space) causes a 6.5% decline in Macro-F1 and a 7.8% drop in MCC, validating that intra-class feature averaging combined with feature–label concatenation yields more discriminative class representations than direct feature propagation.

Table 8 further compares the adopted component designs against functionally equivalent alternatives to justify the specific technical choices.

The original designs consistently outperform their alternatives. For edge construction, squared feature differences (Macro-F1 = 0.832, FNR = 0.080) surpass cosine-similarity k-NN (0.791, 0.135) and multi-head attention (0.801, 0.121), suggesting that element-wise squared distances better encode the nonlinear risk propagation characteristics of supply chain graphs. For prototype computation, the support-set mean (0.832, WAC = 45) outperforms both contrastive initialization (0.815, 53) and learnable vectors (0.808, 58), indicating that simple averaging is more stable and expressive under severe label scarcity. For the graph backbone, the two-layer GCN (0.832, 0.080, 45) surpasses graph attention networks (0.824, 0.087, 49) and graph transformers (0.819, 0.091, 51) while incurring lower computational overhead, confirming a favorable balance between architectural complexity and predictive accuracy.

Table 8: Alternative component design ablation study

Component	Alternative Design	Macro F1-Score	FNR	WAC
V2E Network				
	Original (Squared Feature Diff)	0.832	0.080	45
	k-NN (Cosine Similarity)	0.791	0.135	61
	Multi-head Attention	0.801	0.121	57
Prototype Calculation				
	Original (Support Set Mean)	0.832	0.080	45
	Contrastive Learning Init	0.815	0.098	53
	Learnable Prototype Vectors	0.808	0.112	58
GNN Architecture				
	Original (2-layer GCN)	0.832	0.080	45
	Graph Attention Networks	0.824	0.087	49
	Graph Transformer	0.819	0.091	51

5 Discussion

5.1 Theoretical Contributions and Practical Value

The ISSA-NSGA-III hybrid framework proposed in this study makes three noteworthy contributions to the field of evolutionary computation. First, through the synergistic design of Tent chaotic mapping, adaptive periodic convergence factor, and Lévy flight-elite opposition-based learning, it systematically addresses three major inherent defects of the standard sparrow search algorithm: randomness in population initialization leading to insufficient solution space coverage, poor adaptability of fixed parameter tuning strategies in dynamic environments, and weak local optima escape capability. Experimental results show that compared to standard SSA, the improved algorithm achieves significant improvements in convergence speed, solution set diversity (Spacing Metric improved by 40.0%), and Pareto front quality (Hypervolume increased by 28.5%), supporting the effectiveness of integrating evolutionary algorithms with adaptive control theory [3,5]. This finding is consistent with the research of Gao et al. [3] on multi-strategy enhanced SSA, but this paper further embeds the improvement strategies into the NSGA-III framework, achieving the leap from single-objective optimization to multi-objective collaboration, expanding the application boundaries of SSA [9,10]. It should be acknowledged, however, that the individual improvement components (Tent map, Lévy flight, EOBL) are established techniques in the swarm intelligence literature; the contribution here lies in their joint integration and embedding within NSGA-III, rather than in the novelty of any single component.

Second, this study integrates prototype space embedding and adaptive V2E networks into a coupled optimization–prediction framework through the adoption of the ML-GCNPS architecture proposed by Wang et al. [25]. As discussed in Sections 2.3 and 3.5, the ML-GCNPS model itself—including its prototype space embedding, V2E network, and two-layer GCN—follows the design and achieves the same standalone performance as reported in [25]. The contribution

of the present work is the system-level integration: connecting the risk prediction module with the delivery optimization module through shared graph structure, enabling closed-loop "optimization–assessment–feedback" decision-making that was absent in the standalone setting of [25]. Compared to the local subgraph meta-learning method proposed by Wang et al. [7], ML-GCNPS explicitly models class centers through prototype space, enhancing the model's ability to handle class imbalance; compared to the MAML-GNN method of Lu et al. [8], ML-GCNPS's V2E network can adaptively learn edge weights, more accurately reflecting risk propagation paths in supply chain networks. Ablation experiments confirm that removing the GCN structure leads to a 16.0% decrease in MCC and a 104% increase in WAC, highlighting the indispensability of graph topology modeling in risk prediction [25]. This design philosophy echoes Ivanov's [15] insights on supply chain network resilience: system robustness stems from complex interactions between nodes rather than the strength of individual nodes.

Third, from a practical application perspective, this study provides a technical solution for the green transformation of urban last-mile delivery. In terms of economic benefits, the 14.0% reduction in total cost achieved by ISSA-NSGA-III directly translates into quantifiable profit growth for enterprises—calculated for a medium-sized logistics company with 100,000 annual deliveries, a cost savings of 1.25 CNY per delivery amounts to annual cost savings of 125,000 CNY [1,2]. In terms of environmental sustainability, the 14.2% reduction in carbon emissions (annual reduction of approximately 28.28 tons of CO_2 equivalent) contributes toward the logistics industry's annual emission reduction requirement of 2-3% under the 2030 carbon peak target proposed by the Paris Agreement, and could also generate additional economic benefits of 14,000-23,000 CNY through carbon quota trading (calculated at 50-80 CNY/ton) [11,14]. In terms of risk management, the FNR of 0.080 achieved by ML-GCNPS means that approximately 12 additional high-cost losses can be avoided per 100 deliveries (compared to the FNR of 0.200 for G-MLLS); calculated at an average loss of 5000 CNY per high-risk event, annual risk cost savings can reach 60,000 CNY [17,18]. These estimates are illustrative and depend on deployment-specific factors such as delivery volume, cost structure, and regional carbon pricing; they should be interpreted as indicative of the framework's potential rather than as precise predictions.

5.2 Limitations and Future Improvement Directions

Despite the contributions of this study, the following limitations warrant further exploration.

First, dataset representativeness: as acknowledged in Section 4.1.1, both the Kaggle Supply Chain Management Dataset and the Carbon Monitor risk dataset were not originally designed for urban last-mile delivery scenarios. The Kaggle dataset lacks urban-specific features such as real-time traffic conditions, delivery time windows, and customer accessibility constraints, while the Carbon Monitor dataset captures enterprise-level rather than neighborhood-level risks. Consequently, the reported performance improvements should be interpreted as evidence of the framework's effectiveness on multi-echelon supply chain optimization and graph-based risk classification, with the caveat that validation on dedicated urban delivery datasets—potentially including GPS trajectory data, real-time traffic feeds, and fine-grained delivery outcome records—is needed to fully establish external validity for last-mile applications.

Second, robustness to demand uncertainty: the current experimental evaluation uses fixed demand values derived from historical records, without systematically testing the framework's behavior under demand fluctuations, supply disruptions, or other stochastic perturbations. In practice, urban delivery demand exhibits significant variability due to seasonal patterns, promotional events, and weather conditions. Future work should incorporate robustness analyses, such as evaluating optimization performance under demand perturbations of $\pm 10\%$, $\pm 20\%$, and $\pm 30\%$ relative to baseline levels, and testing risk prediction stability when node features are subject to noise or missing values. Robust optimization formulations (e.g., min-max or chance-constrained approaches) could also be integrated into the ISSA-NSGA-III framework to explicitly account for uncertainty in the objective functions.

Third, data timeliness issues: the current model is trained on historical data from 2015-2022, and its adaptability to new business models (such as unmanned delivery and instant retail) and new policies (such as the EU Carbon Border Adjustment Mechanism) that emerged after 2023 has not been fully validated. The research on learning-driven memetic algorithms by Zhao et al. [6] shows that reinforcement learning can achieve continuous adaptation to dynamic environments through incremental updates. Future research can introduce online learning mechanisms, using newly arrived delivery data to fine-tune model parameters in real-time, constructing a three-stage training paradigm of "pre-training - continual learning - dynamic updating" [27,28].

Fourth, cold start problems: when a large number of new entities enter simultaneously (such as new city expansion or large-scale holiday promotions) or their feature distributions differ significantly from the training set, model performance still faces challenges. Although meta-learning strategies provide certain generalization capabilities, they still rely on domain knowledge priors in zero-shot scenarios (such as delivery network initialization in completely new cities) [12,13]. The scenario analysis method proposed by Xu et al. [19] in their research on dynamic optimization of container logistics supply chains can provide insights: by constructing a typical scenario library (such as holiday peaks, extreme weather, sudden epidemics), pre-training targeted expert models, and then achieving rapid adaptation through model fusion strategies. Future work can explore the integration of inductive graph learning with zero-shot learning mechanisms to more systematically address entity cold start problems [16,20].

Fifth, multi-modal data fusion: the current model is mainly based on structured data (enterprise finance, carbon emission indicators) and graph topology, without fully utilizing unstructured data (such as traffic surveillance videos, social media sentiment, and meteorological satellite data). Zhou et al. [18] emphasized the importance of multi-source data fusion in coal supply chain risk assessment. Future work can introduce multi-modal pre-trained models (such as CLIP for visual-text alignment and spatiotemporal Transformers for trajectory data), integrating visual features (traffic congestion levels), text features (news event sentiment), and spatiotemporal features (weather patterns) into node representation learning through cross-modal attention mechanisms, enhancing the comprehensiveness of risk perception [21,22].

Sixth, interpretability enhancement: although this study provides some interpretability through prototype space embedding and V2E networks, the black-box nature of the model remains an application barrier for regulatory authorities and non-technical decision-makers.

The research of Dixit and Gehlot [17] combining decision trees with Bayesian networks shows that probabilistic graphical models can provide intuitive causal inference paths. Future work can combine causal inference frameworks, quantifying the marginal contribution of each risk factor through counterfactual analysis—for example, "if supplier A's carbon emission intensity decreases by 10%, by how much would the high-risk probability of downstream distribution center B decrease"—generating "if-then" style decision recommendations to meet the interpretability requirements of the EU AI Act for high-risk AI systems [23,24].

Seventh, distributed computing extension: when the delivery network scale expands to tens of thousands of nodes (such as a nationwide logistics network), the current single-GPU-based training approach will face memory and computational bottlenecks. Ma et al. [24] encountered similar challenges in wood supply chain system dynamics simulation, alleviating computational pressure through modular design and parallel computing. Future work can adopt graph sampling techniques (such as GraphSAINT using random walk sampling and ClusterGCN based on clustering to partition subgraphs) to reduce the computational complexity of single iterations, or introduce federated learning frameworks to achieve multi-location distributed training while protecting enterprise data privacy, constructing a collaborative optimization architecture of "central model aggregation - edge node fine-tuning" [26,29].

Eighth, comparison with alternative paradigms: the present study adopts evolutionary optimization (ISSA-NSGA-III) for delivery planning and graph-based meta-learning (ML-GCNPS) for risk prediction. An alternative approach would be to formulate the delivery optimization problem as a Markov Decision Process and apply deep reinforcement learning (deep RL) or multi-agent reinforcement learning (MARL) methods, which have shown promise in dynamic vehicle routing problems with real-time re-optimization capabilities [6]. Compared to RL-based approaches, the evolutionary optimization framework adopted here offers several advantages: it naturally handles multiple competing objectives through Pareto front construction, does not require a carefully designed reward function, and provides a complete set of trade-off solutions rather than a single policy. However, RL methods are better suited for online, sequential decision-making in highly dynamic environments where routes must be adjusted in real time. A systematic empirical comparison between the proposed framework and state-of-the-art deep RL methods (e.g., attention-based models for VRP) is an important direction for future work, particularly for scenarios requiring real-time re-routing in response to traffic disruptions or demand changes.

Among the above directions, we consider dataset validation and robustness analysis (first and second) to be the most urgent near-term priorities, as they directly address the external validity concerns raised by reviewers and are feasible with existing methodology. Online learning mechanisms (third) represent the next priority, as they would enable the framework to adapt to evolving delivery patterns without full retraining. The remaining directions—cold start mitigation, multi-modal fusion, interpretability, distributed computing, and RL comparison—are important medium-to-long-term goals that require more substantial methodological development or data infrastructure.

5.3 Implications for Smart City Logistics Development

This study provides three observations for the design and deployment of smart city logistics systems. First, from the perspective of technical route selection, relying solely on mathematical programming or deep learning is difficult to cope with the complexity and uncertainty of urban delivery; it may be beneficial to construct a hybrid intelligent paradigm of "evolutionary algorithms + graph neural networks + meta-learning" [6,7]. The advantage of this paradigm lies in: evolutionary algorithms handle multi-objective trade-offs and global search, graph neural networks capture network topology and risk propagation, and meta-learning achieves rapid generalization with few samples. Experiments show that the synergistic benefits of the three far exceed those of a single method—ISSA-NSGA-III improves Hypervolume by 28.5% compared to standard SSA, and ML-GCNPS improves MCC by 48.3% compared to ProtoNG. This is consistent with the theory of Azadegan and Dooley [29] that "complex system resilience stems from multi-agent collaboration," providing empirical support for smart city logistics technology architecture design.

Second, from a policy-making perspective, achieving carbon neutrality goals requires the synergistic efforts of "technological innovation + policy incentives + market mechanisms" [30]. The 14.2% reduction in carbon emissions in this study mainly comes from route optimization and capacity scheduling, but this value still has a gap from the 18% reduction in carbon emission intensity of the logistics industry proposed in China's Action Plan for Carbon Dioxide Peaking Before 2030. Zhou et al. [30] found that the moderating effect of government science and technology policy support on supply chain resilience is significant, indicating the importance of policy instruments. Future policy design can consider: (1) establishing a differentiated carbon pricing mechanism, providing carbon quota rewards or tax reductions to logistics companies that adopt intelligent optimization systems; (2) promoting open sharing of urban delivery data, constructing a data ecosystem of government-enterprise-university collaboration to lower the threshold for algorithm research and development; (3) formulating evaluation standards and certification systems for intelligent logistics systems to guide standardized industry development [11,14,18].

Third, from the perspective of industrial ecosystem construction, the sustainable development of smart city logistics requires connecting the entire chain of "algorithm R&D-platform deployment-service operation." Logistics companies currently generally face the dilemma of "having data but no algorithms, having algorithms but no platforms, having platforms but no operations." The code framework provided by this study lowers the technical threshold for small and medium-sized enterprises to apply advanced optimization algorithms, but overcoming the leap from laboratory prototypes to commercial deployment still requires bridging three gaps: (1) model generalizability—the experimental dataset covers 500 enterprises, while practical applications need to adapt to tens of thousands of heterogeneous enterprises; (2) system reliability—academic research pursues metric improvement, while industrial applications require 7$\times$24-h stable operation and millisecond-level response; (3) business model innovation—transitioning from "one-time software sales" to "algorithm-as-a-service (AaaS)" subscription models to reduce initial enterprise investment costs. Solving these challenges requires deep industry-academia-research collaboration, as emphasized by Hosseini et al. [16] in their research on supply chain resilience quantification:

theoretical innovation must be closely integrated with practical needs to produce genuine social value [1,2,19].

6 Conclusion

Addressing the dual challenges of multi-objective optimization and risk prediction in urban last-mile delivery, this study proposes a hybrid intelligent framework integrating an Improved Sparrow Search Algorithm (ISSA) with a Meta-Learning Graph Convolutional Network on Prototype Space (ML-GCNPS, whose architecture was originally developed in prior work and is adopted here within a coupled optimization–prediction system). Through the synergistic enhancement of Tent chaotic mapping, adaptive periodic convergence factor, and Lévy flight-elite opposition-based learning, ISSA-NSGA-III achieves significant results on real supply chain datasets: 14.0% reduction in total cost, 14.2% reduction in carbon emissions, and 15.4% improvement in resource utilization, with a 28.5% improvement in Pareto front quality, with all improvements statistically significant at $p < 0.05$ over 10 independent runs. The ML-GCNPS risk prediction module, integrated into the coupled framework through shared graph structure and closed-loop feedback, achieves leading performance with AUPRC of 0.850 ± 0.005 and FNR of 0.080 ± 0.003 in 5-way 1-shot scenarios, effectively addressing the problems of high-risk sample scarcity and severe class imbalance. Ablation experiments systematically verify the contribution of each core module, and alternative design comparisons further demonstrate the rationale behind the technical selections. The principal contribution of this work lies not in the novelty of individual algorithmic components, but in their systematic integration into a closed-loop "optimization–assessment–feedback" decision framework, where the delivery network's graph structure serves as a shared information bridge connecting multi-objective optimization with risk prediction. This study provides an end-to-end decision support tool for smart city logistics, offering theoretical foundation and technical pathways for achieving sustainable urban delivery under carbon neutrality goals. Key limitations include the use of general supply chain datasets rather than urban delivery-specific data, the absence of robustness analysis under demand uncertainty, and the lack of empirical comparison with reinforcement learning-based approaches. Future research will prioritize validation on dedicated urban delivery datasets and robustness testing under stochastic demand scenarios, followed by online learning mechanisms to address dynamic environments, inductive graph learning to solve entity cold start problems, multi-modal data fusion to enhance risk perception comprehensiveness, causal inference to improve interpretability, and federated learning to support large-scale distributed deployment, promoting the leap of smart logistics systems from laboratory prototypes to industrial applications.

Acknowledgment

The author would like to express sincere gratitude to the School of Environment, Education and Development at the University of Manchester for providing the research environment and computational resources necessary for this study. Special thanks are extended to colleagues and peers who provided valuable feedback during the research process.

Funding Statement

This research received no specific grant from any funding agency in the public, commercial, or not-for-profit sectors.

Author Contributions

Conceptualization, Jing Xu and Liansheng Yuan; methodology, Jing Xu and Xinke Du; software, Jing Xu; validation, Jing Xu, Liansheng Yuan and Xinke Du; formal analysis, Jing Xu; investigation, Liansheng Yuan; resources, Xinke Du; data curation, Jing Xu; writing—original draft preparation, Jing Xu; writing—review and editing, Liansheng Yuan and Xinke Du; visualization, Jing Xu; supervision, Liansheng Yuan and Xinke Du; project administration, Jing Xu. All authors reviewed and approved the final version of the manuscript.

Availability of Data and Materials

The Supply Chain Management Dataset used in this study is publicly available on Kaggle (https://www.kaggle.com/datasets/lastman0800/supply-chain-management). The Carbon Monitor risk dataset is based on publicly available data from Carbon Monitor (https://carbonmonitor.org) and the International Energy Agency (IEA). The implementation code and preprocessed datasets are available from the corresponding author upon reasonable request to ensure reproducibility of the results.

Ethics Approval

Not applicable. This research does not involve human subjects, animal experiments, or any ethical concerns requiring institutional review board approval.

Conflicts of Interest

The authors declare no conflicts of interest.

References

1. Shi Y, Lin Y, Lim MK, Tseng ML, Tan C, Li Y. An intelligent green scheduling system for sustainable cold chain logistics. Expert Systems with Applications. 2022;209:118378.
2. Boskabadi A, Mirmozaffari M, Yazdani R, Farahani A. Design of a distribution network in a multi-product, multi-period green supply chain system under demand uncertainty. Sustainable Operations and Computers. 2022;3:226-37.
3. Gao BW, Shen W, Guan H, Zheng LT, Zhang W. Research on multistrategy improved evolutionary sparrow search algorithm and its application. IEEE Access. 2022;10:62520-34.
4. Wu H, Zhang A, Han Y, Nan J, Li K. Fast stochastic configuration network based on an improved sparrow search algorithm for fire flame recognition. Knowledge-Based Systems. 2022;245:108626.
5. Awadallah MA, Al-Betar MA, Braik MS, et al. Recent versions and applications of sparrow search algorithm. Archives of Computational Methods in Engineering. 2023;30(5):2831-58.
6. Zhao S, Zhou H. Learning-driven memetic algorithm for solving integrated distributed production and transportation scheduling problem. Swarm and Evolutionary Computation. 2025;96:101945.

7. Wang Y, Huang C, Li M, Huang Q, Wu X, Wu J. AG-Meta: Adaptive graph meta-learning via representation consistency over local subgraphs. Pattern Recognition. 2024;151:110387.
8. Lu C, Wang X, Yang A, Liu Y, Dong Z. A few-shot-based model-agnostic meta-learning for intrusion detection in security of internet of things. IEEE Internet of Things Journal. 2023;10(24):21309-21.
9. Tanhadoust A, Madhkhan M, Nehdi ML. Two-stage multi-objective optimization of reinforced concrete buildings based on non-dominated sorting genetic algorithm (NSGA-III). Journal of Building Engineering. 2023;75:107022.
10. Gao S, Zhang Y, Zhang Z, Tan D, Li J, Yin Z, et al. Multi-objective optimization of the combustion chamber geometry for a Highland diesel engine fueled with diesel/n-butanol/PODEn by ANN-NSGA III. Energy. 2023;282:128793.
11. Wang Z, Huang D. A new perspective on financial risk prediction in a carbon-neutral environment: A comprehensive comparative study based on the SSA-LSTM model. Sustainability. 2023;15(19):14649.
12. Li F, Sun M, Xian Q, Feng X. Industrial carbon emission prediction method based on meta-learning and diff long short-term memory networks. PLoS ONE. 2024;19(7):e0307915.
13. Enamoto LM, Santos ARA, Weigang L, Meneguette R, Rocha Filho GP. Meta-learning applied to a multivariate single-step fusion model for greenhouse gas emission forecasting in Brazil. Journal of Water and Climate Change. 2024;15(10):4016-34.
14. Niu H, Zhang Z, Luo M. Evaluation and prediction of low-carbon economic efficiency in China, Japan and South Korea: Based on DEA and machine learning. International Journal of Environmental Research and Public Health. 2022;19(19):12709.
15. Ivanov D. Revealing interfaces of supply chain resilience and sustainability: a simulation study. International Journal of Production Research. 2018;56(10):3507-23.
16. Hosseini S, Ivanov D, Dolgui A. Review of quantitative methods for supply chain resilience analysis. Transportation Research Part E: Logistics and Transportation Review. 2019;125:285-307.
17. Dixit P, Gehlot K. Combining decision tree and Bayesian networks for improved predictive analytics. In: Hybrid Information Systems: Non-Linear Optimization Strategies with Artificial Intelligence. De Gruyter; 2024. p. 91-114.
18. Zhou Y, Liu X, Wang S, et al. Multidimensional risk assessment in sustainable coal supply chains for China's low-carbon transition: An AHP-FCE framework. Sustainability. 2025;17(13):5689.
19. Xu B, Liu W, Li J, Yang Y, Wen F, Song H. Resilience measurement and dynamic optimization of container logistics supply chain under adverse events. Computers & Industrial Engineering. 2023;180:109202.
20. Zhang X, Wang Y, Zhang D. Location-routing optimization for two-echelon cold chain logistics of front warehouses based on a hybrid ant colony algorithm. Mathematics. 2024;12(12):1851.
21. Longo F, Mirabelli G, Solina V. A simulation-based framework for manufacturing design and resilience assessment: A case study in the wood sector. Applied Sciences. 2022;12(15):7614.
22. Hayek M, Mahringer G, Segner R, et al. Development of a Physical Internet container for an optimized wood supply chain. Transportation Research Procedia. 2023;72:1950-7.
23. Auer V, Rauch P. Wood supply chain risks and risk mitigation strategies: A systematic review focusing on the Northern hemisphere. Biomass and Bioenergy. 2021;148:106001.
24. Ma C, Zhang L, Gao X, et al. Modelling and evaluating wood supply chain resilience factors based on system dynamics. Scientific Reports. 2025;15:16207.
25. Wang Y, Sun Y. Low-carbon supply chain logistics risk prediction using meta-learning-based graph convolutional network on prototype space. Scientific Reports. 2026;16:2352.
26. Wang G. Green supply chain management and coordinated optimization by an improved sparrow search algorithm. Scientific Reports. 2025;15:36730.

27. Gholian-Jouybari F, Hashemi-Amiri O, Mosallanezhad B. Metaheuristic algorithms for a sustainable agri-food supply chain considering marketing practices under uncertainty. Expert Systems with Applications. 2023;213:118880.
28. Zhu J, Chen X, Liu Y, et al. Sustainable optimization in supply chain management using machine learning. International Journal of Management Science Research. 2025;8:1-8.
29. Azadegan A, Dooley K. A typology of supply network resilience strategies: complex collaborations in a complex world. Journal of Supply Chain Management. 2021;57(1):17-26.
30. Zhou L, Tang C, Cao Y. Innovative human capital, government support for science and technology policy, and supply chain resilience. Finance Research Letters. 2025;74:106741.