

Hybrid optimization framework for closed-loop supply chains: embedding neural networks in two-stage stochastic programming

IMAN SEYEDI[†] AND ENZA MESSINA

Department of Computer Science Systems and Communication, University of Milano-Bicocca, 20126 Milan, Italy

[†]Corresponding author. Email: Seyediman.seyedi@unimib.it

[Received on 9 April 2025; accepted on 9 September 2026]

Accepted by: M. Zied Babai

This paper develops an integrated surrogate model for optimizing a sustainable closed-loop supply chain (CLSC) in an additive manufacturing context. The model combines a neural network with two-stage stochastic programming (2SP) to handle demand uncertainty. This model significantly improves strategic and operational decision-making by integrating critical elements such as location planning and material flows, leveraging machine learning (ML) to manage uncertainties and enhance adaptability to dynamic demand behaviours. The efficacy of our approach is supported by computational experiments showing that the integration of 2SP and ML can provide solutions that are more scalable and computationally efficient than sample average approximation for the problem instances considered. The results suggest that this model may offer a useful tool for practitioners working on sustainable CLSC design under demand uncertainty, though further validation on larger real-world instances is needed. From a managerial perspective, the proposed framework can support sustainable investment planning and risk management in circular economy strategies.

Keywords: neural network surrogate model; two-stage stochastic programming; additive manufacturing; stochastic optimization; sustainable closed loop supply chain.

1. Introduction

Optimizing global supply chains under uncertainty requires advanced computational approaches to enable real-time decision-making. Despite the growing interest in stochastic programming for supply chain management, research gaps persist in adequately modelling uncertainties in closed-loop systems. The integration of machine learning (ML) within the optimization framework can offer a way to address the complexity of the resulting problem. In this paper, we integrate mixed-integer linear optimization with deep neural networks to efficiently address demand uncertainty. In this paper, we develop a two-stage stochastic programming (2SP) model for the 3D printing closed-loop supply chain with demand uncertainty. A sustainable manufacturing approach incorporating Polyethylene Terephthalate (PET) bottle waste into closed-loop 3D printing is essential for addressing global plastic pollution, conserving resources and reducing dependence on virgin materials. By promoting reuse and recycling, this approach aligns with circular economy principles and fosters green manufacturing innovation. [Figure 1](#) presents the collected rate of PET bottles in some European countries in the year 2023. The success of PET recycling across Europe highlights the crucial role of well-established deposit-return schemes in ensuring a steady supply of high-quality recycles for sustainable manufacturing.

Closed-loop supply chains present unique optimization challenges that are substantially amplified under demand uncertainty. Unlike forward-only supply chains, closed-loop supply chain (CLSCs) must

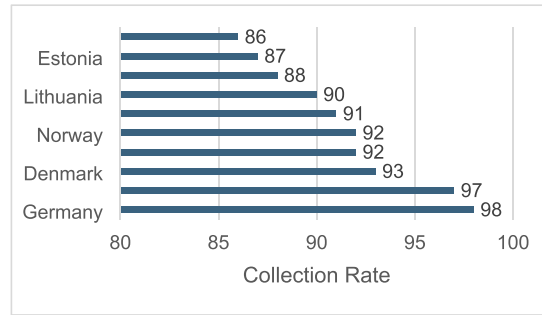


FIG. 1 PET Collection Rates in Europe, 2023.

simultaneously coordinate facility location, material flows, production scheduling and reverse logistics, all of which are sensitive to fluctuations in both customer demand and return flows. When demand is uncertain, these interdependencies can propagate inefficiencies across the entire network, leading to stockouts, excess inventory, underutilized recycling capacity and suboptimal facility configurations (Gao *et al.*, 2024; Akbari-Aghghaleh *et al.*, 2025). In the specific context of 3D-printing CLSCs, demand variability is particularly pronounced: the additive manufacturing (AM) market exhibits a compound annual growth rate of approximately 21% (Protolabs, 2024), with significant fluctuations driven by technological change, shifting customer preferences and seasonal patterns. This makes deterministic planning approaches fundamentally inadequate.

2SP is the standard framework for optimizing systems under uncertainty, distinguishing between first-stage decisions made before uncertainty is revealed such as facility opening choices, and second-stage recourse decisions made after demand is realized, such as material flows and production quantities. While 2SP provides robust solutions, its computational cost grows rapidly with the number of scenarios: the deterministic equivalent formulation requires solving a single large-scale mixed-integer program whose size scales linearly with the number of scenarios in both variables and constraints. In practice, even moderate scenario sets (e.g., 50–100 scenarios) render this approach computationally intractable for realistic CLSC instances. Alternative methods such as sample average approximation (SAA) (Kleywegt *et al.*, 2002) and decomposition techniques like the L-shaped method (Van Slyke & Wets, 1969) mitigate this to some extent, but still require repeated second-stage solves and exhibit limited scalability as the scenario set grows. In contrast, the proposed neural network surrogate generalizes more effectively across unseen scenarios, particularly when the number of training scenarios is small.

The modeling of demand uncertainty in CLSC contexts has received growing attention. Studies such as Badri *et al.* (2017), Liu *et al.* (2023), and Khalili-Fard *et al.* (2024) have applied 2SP to general CLSC network design under demand uncertainty, demonstrating the value of stochastic over deterministic solutions. In the AM context specifically, Chowdhury *et al.* (2019) introduced a 2SP model for AM supply chain network design, incorporating demand variability and wastage rates. However, existing work on 3D-printing CLSCs either relies on small, fixed scenario sets that limit solution flexibility, or does not address the scalability of the solution method when the scenario space grows. To our knowledge, no prior work combines a neural network surrogate approach with 2SP specifically for a 3D-printing closed-loop supply chain that integrates both forward and reverse logistics with recycled PET materials. This gap motivates the present study.

To enhance computational efficiency, while maintaining solution accuracy, we embed within the optimization framework a neural network that approximates the expected second-stage costs. This approach supports strategic decision-making, enabling flexible and sustainable solutions.

The main contributions of the paper are twofold:

1. We extend the deterministic CLSC model of [Rajabi-Kafshgar et al. \(2025\)](#) by explicitly incorporating uncertainty in both market and customer demand. Here, market (recycling companies) demand refers to the demand for recycled materials by reclaimers and processing centers (PC), while customer demand corresponds to the demand for 3D-printed end products. This allows the model to more realistically capture variability across both production and consumption levels, improving strategic decision-making flexibility and ensuring greater service reliability in closed-loop supply chains.
2. We employ a neural network surrogate approach within the 2SP formulation to approximate the second-stage recourse function, which scales with the number of scenarios in the model. This enables a more flexible and generalizable solution. Compared to [Patel et al. \(2022\)](#), our approach is specifically designed to efficiently evaluate the recourse cost function across a large number of scenarios. Indeed, [Patel et al. \(2022\)](#) implemented a two-neural-network architecture where one is used for embedding scenarios and the other for approximating the recourse function value. This requires the generation of training samples that span the domain of the first stage solutions, augmented with the space of possible scenarios. This significantly increases the training burden and hinders scalability to real case studies. For this reason, we approximate the recourse value function by using a single network model, which approximates the recourse function obtained over a fixed number of scenarios, thus not depending on the scenario embedding models. By embedding this network in the optimization model, we exploit the neural network generalization capability, enabling solutions that perform well even on previously unseen scenarios.

The paper is organized as follows: [Section 2](#) reviews research in this field; [Section 3](#) outlines the proposed model; [Section 4](#) discusses the value of flexibility; [Section 5](#) introduces the solution methodology; [Section 6](#) discusses computational experiments and data analysis; and [Section 7](#) presents conclusions and future research directions.

2. Literature review

CLSCs integrate forward and reverse logistics to create a circular economy based on refurbishing, reusing, remanufacturing and recycling. It helps to achieve maximum resource utilization with a minimum waste generation rate for sustainable manufacturing ([Ade Irawan et al., 2022](#); [Lozano-Oviedo et al., 2024](#)). This approach offers economic, social and environmental benefits across industries like electronics, automotive and AM ([Simonetto et al., 2022](#); [Shaharudin et al., 2023](#); [Xing et al., 2024](#)). In this paper, the design of CLSC networks has been developed in strategic and operational decisions. The location of facilities and capacity planning decisions are included at the strategic level, while production quantities and material flows are covered at the operational level. Therefore, effective CLSC design would be able to consider these factors in detail with models enabling optimum performance across the entire supply chain.

The performance of CLSCs is significantly affected by uncertainties in demand and return flows, leading to costly inefficiencies such as stockouts and excess inventory ([Seyedi et al., 2016](#); [Chen et al., 2024](#); [Gao et al., 2024](#)). Demand uncertainty impacts inventory management and production planning, while return flow quality influences refurbishment and remanufacturing feasibility. Effective

management strategies, including flexible production systems, dynamic inventory management and robust network design, are essential for ensuring the reliability and efficiency of CLSCs (Hosseini *et al.*, 2014; Biçe & Batun, 2021). 2SP is a widely adopted methodology for optimizing CLSC network design under uncertainty, offering robustness by integrating strategic first-stage decisions with operational second-stage decisions once uncertainties are realized (Khalili-Fard *et al.*, 2024; Shafiee Roudbari *et al.*, 2024). Studies have shown that addressing these uncertainties makes CLSCs more resilient and adaptive (Tavana *et al.*, 2022; Gaula & Jha, 2024; Han *et al.*, 2024).

Amirteimoori *et al.* (2022) in their paper, evaluated the relative efficiency of SCs in the presence of reverse flows and stochastic data, underlying the importance of considering uncertainties for increasing the reliability of the efficiency and sustainability of operations in SCs.

AM, particularly 3D printing, has seen increased adoption due to its ability to reduce waste, use recycled materials and improve resource efficiency in manufacturing. Studies highlight its energy and material advantages over traditional methods (Gibson *et al.*, 2015). The urgency to address plastic pollution has prompted AM's exploration for sustainability, particularly in incorporating recycled materials like plastics, supporting a circular economy by reducing waste and enabling localized production (Seyedi & Maleki-Daronkolaie, 2013; Sanchez *et al.*, 2020; Cañado *et al.*, 2022; Priavoulou *et al.*, 2022).

Chowdhury *et al.* (2019) introduced an integrated AM supply chain network model using 2SP, incorporating facility location and capacity decisions under demand uncertainty. This work emphasized process-level factors like demand variability and wastage rates, which are essential for sustaining additive manufacturing. Fleischmann *et al.* (1997) laid the theoretical foundation for CLSCs, highlighting reverse logistics and circular economy principles. 2SP has been extensively applied to optimize inventory, transportation and resource allocation under uncertainty (Liu *et al.*, 2023). Liu & Papageorgiou (2013) further advanced these concepts with sophisticated models featuring multiple uncertainty sources, revealing the limitations of deterministic optimization methods. Badri *et al.* (2017) developed a 2SP model for value-based supply chain network design, considering customer demand and core availability uncertainties, emphasizing the importance of balancing tactical and strategic decisions. Addressing the limitations of risk-neutral models, Liu *et al.* (2023) proposed a risk-averse 2SP framework using parallel simplified swarm optimization, balancing expected profits and risk, thus filling a critical research gap in handling uncertainty in complex supply chains.

Machine learning (ML) has become a vital tool in supply chain optimization, offering advanced techniques for demand prediction, production scheduling and decision-making processes. Neural networks, a subset of ML, have significantly improved prediction accuracy by capturing complex nonlinear relationships in data (Carbonneau *et al.*, 2008; Goodfellow *et al.*, 2016; Seyedi *et al.*, 2025). Nassehi *et al.* (2022) explored the integration of ML algorithms with traditional optimization models, enhancing the forecasting capabilities of supply chain frameworks, particularly in uncertain demand scenarios (Alabi *et al.*, 2022). Applying deep learning in complex supply chains, such as 3D printing recycling, has shown great potential for improving decision-making by enabling faster and more accurate predictions.

Integrating ML with 2SP offers a promising solution for complex optimization challenges under uncertainty. While 2SP effectively handles decision-making under unpredictable conditions, neural networks excel at uncovering intricate relationships and improving computational efficiency (Weichert *et al.*, 2019). Recent advancements have demonstrated that using neural networks as surrogate models for predicting second-stage costs can reduce computational burdens in large-scale stochastic programming (Patel *et al.*, 2022). Fischetti & Jo (2018) highlighted the potential of neural networks to predict solution quality across various uncertainty scenarios, providing an effective alternative to traditional scenario-based approximation techniques. This synergy enhances the scalability and efficiency of supply chain optimization models, enabling better resource management and strategic decision-making.

Several recent works have explored the integration of ML with 2SP. [Patel et al. \(2022\)](#) proposed Neur2SP, considering two neural network architectures.

The proposed approach in [Patel et al. \(2022\)](#) integrates scenario information directly into the learning process by training neural networks that map first-stage decisions and scenario embeddings to recourse function values. This design allows the surrogate to approximate the conditional response of the system under varying stochastic environments. In particular, in [Patel et al. \(2022\)](#) two distinct neural network architectures are considered:

The single-cut architecture learns a direct mapping $(x, \{\xi^s\}_{s=1}^{|S|}) \mapsto \bar{Q}(x, \{\xi^s\}_{s=1}^{|S|})$. The entire scenario set is first encoded and aggregated into a joint representation ζ , which is then concatenated with the first-stage decision vector x as input to a single neural network. Alternatively, the multi-cut architecture learns the mapping $(x, \xi^s) \mapsto Q(x, \xi^s)$ separately for each scenario ξ^s . Both frameworks have proven effective in various classical two-stage SP problems. A commonality in these successful applications is that the first-stage decision space is typically low-dimensional and primarily composed of binary variables (only one of their computational case studies has a two-dimensional first-stage decision whose variables are restricted in the interval $[0,5]$). Dealing with continuous first stage decisions makes it harder for the neural network to generalize from a finite training set.

Moreover, the scenario embedding aggregation tends to produce a scenario set embedding that is mediated among scenarios and may lose information about the distribution of the scenario set. Significantly high, especially when dealing with large-scale problems or complex scenario spaces that require extensive data and multiple training iterations.

Differently, our approach focuses on learning the expected performance of continuous first-stage decisions on an out-of-sample scenario set. Neural networks are trained using recourse values computed over a sufficiently large scenario set, enabling the model to implicitly capture stochastic variability without explicit scenario conditioning, allowing the network to generalize to unseen out-of-sample scenarios. [Fischetti & Jo \(2018\)](#) demonstrated that trained neural networks can be embedded into Mixed-Integer Linear Programming (MILP) formulations via big-M constraints on ReLU activations; this constitutes the standard baseline for NN-in-MIP embedding, and we adopt it here through its implementation in the Gurobi solver. Several refinements to this reformulation exist in the literature but fall outside the scope of this paper. [Table 1](#) provides a structured comparison of these approaches. As summarized in [Table 1](#), the proposed approach reduces training complexity by eliminating scenario embedding, while retaining the MILP-compatibility of the [Fischetti & Jo \(2018\)](#) framework and extending it to a realistic CLSC application.

This study contributes to sustainable supply chain management, AM and operational research by integrating deep neural networks with 2SP for 3D printing CLSCs. Building upon the work of [Rajabi-Kafshgar et al. \(2025\)](#), the proposed model uniquely combines advanced 3D printing technologies with rPET materials, addressing demand uncertainty in sustainable manufacturing. The novel modeling approach incorporates demand fluctuations, offering a flexible and practical framework adaptable to diverse business environments.

Integrating ML to address computational challenges inherent in traditional scenario-based stochastic optimization is a key innovation. Second-stage costs are efficiently approximated by embedding a neural network surrogate model within the 2SP framework, reducing computational complexity without sacrificing solution quality. This methodology advances existing literature by moving beyond purely deterministic or simplified stochastic models, ensuring robust and scalable decision-making for large-scale CLSC instances. The proposed framework offers decision-makers a powerful tool for adopting environmentally and economically sustainable practices in various industrial applications, bridging key research gaps and pushing the boundaries of operational research methodologies.

TABLE 1 *Comparative analysis of related neural network-based optimization approaches with respect to the proposed method*

Aspect	Patel <i>et al.</i> (2022)	Fischetti & Jo (2018)	Proposed method
Network architecture	Two networks (scenario embedding + recourse approximation)	Single network embedded in MILP	Single network embedded in MILP
Training input	First-stage decisions and scenario vectors	First-stage decisions	First-stage decisions
Training target	Per-scenario recourse function value $Q(x, \xi)$	Solution quality metric	Expected recourse cost $E[Q(x, \xi)]$ over a fixed scenario set
Scenario embedding required	Yes	No	No
Training data size	Large (joint decision–scenario space)	Moderate	Moderate
MILP embedding via big-M	Yes	Yes	Yes
Applied to CLSC	No	No	Yes
Scalability to large instances	Limited	Moderate	Demonstrated up to large instances

3. Problem definition

The paper proposes a 2SP mixed-integer linear programming formulation for the CLSC network of 3D printing filament operations in the circular economy. In the first stage, location decisions are made; material flow calculations between facilities are performed in the second stage. We consider the closed-loop supply chain shown in Fig. 2, which consists of two parallel networks, one forward and one backward. In the forward network, recycling centers (RCs) provide filaments to 3D printing centers (DPCs), which send manufactured 3D-printed objects to customers, while the wastes from printing processes are sent to treatment centers (TCs). Also, customers can send their waste directly to TCs.

In the backward network, the PET bottles collected by end-users are transported to collection centers (CCs), where they are sorted and prepared for the PCs. In PCs PET bottles are compressed and formed into bales before being routed to either reclaimers (Markets) or TCs. The TCs node connects the two sub-networks. Indeed, it receives materials from FCs, DPCs, and PCs that are shipped back to RCs after being ground into flakes, washed, and dried.

Finally, in the RCs, the flakes blended with raw materials to improve filament quality will be subsequently processed into 3D printing filaments, and the loop restarts.

Uncertainty arises from fluctuations in market and customer demand, called (DM) and (DC), respectively, due to costumer behavior, market trends, and quality variations. To model this uncertainty, we generated a set of scenarios. Specifically, we employed a Latin hypercube sampling method to create scenarios within a range that captures the spectrum of potential demand fluctuations.

In this paper, the term hybrid optimization refers to the integration of mathematical programming (2SP) with ML (a neural network surrogate) within a single decision framework. This combination leverages the predictive power of neural networks while retaining the rigorous structure of optimization models.

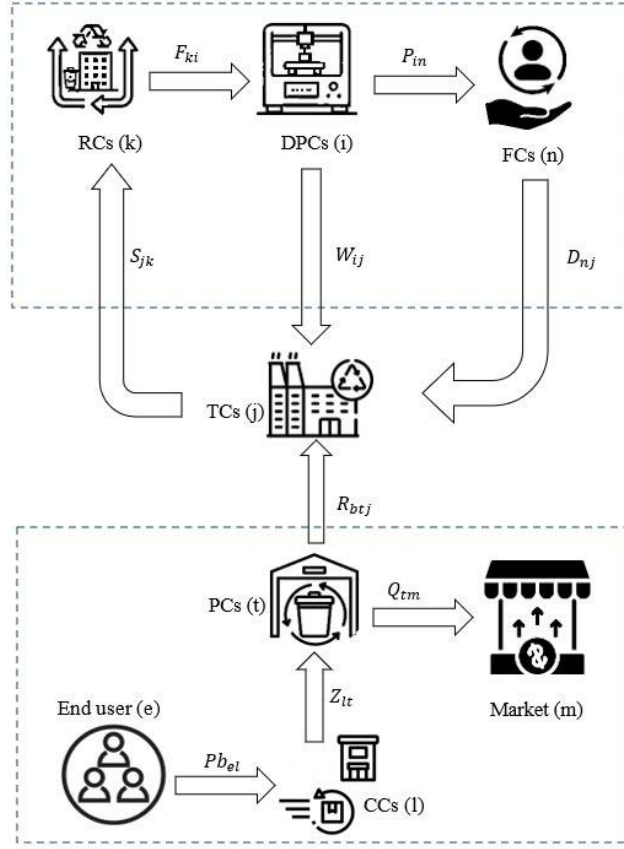


FIG. 2 Proposed 3D printing CLSC network.

Throughout this paper, demand uncertainty is modelled by perturbing base-case demand values within a $\pm 20\%$ range, reflecting realistic variability in the AM market (Protolabs, 2024). Scenarios are generated using Latin hypercube sampling (LHS) to ensure good coverage of the uncertainty space. The number of scenarios used varies by purpose: a coarse three-scenario set (high: $+20\%$, average: 0% , low: -20%) is used in Section 4 for the expected value of perfect information (EVPI) and value of stochastic solution (VSS) analysis, following standard practice in the stochastic programming literature (Birge, 1982); a set of 100 LHS-generated scenarios is used in Section 5 for neural network training; and a set of 10 scenarios is used for SAA in the comparative experiments of Section 6, representing the maximum scenario budget under which SAA remains computationally tractable for the problem sizes considered.

3.1. Formulation of the proposed 2SP

The mathematical model proposed in this section is a stochastic version of the Mixed Integer Programming model presented in Rajabi-Kafshgar *et al.* (2025) where demands are considered as uncertain parameters.

In particular, we used the two-stage stochastic programming framework, given by:

$$\min_{x \in X} \left\{ c^\top x + E_\xi [Q(x, \xi)] \right\}, \quad (1)$$

where:

$$Q(x, \xi) = \min_{y^\xi} \left\{ q^\top y^\xi : Ay^\xi \geq b - Bx, y^\xi \geq 0 \right\}, \quad (2)$$

x , is a set of first-stage decisions, which in this case will represent facility setups, while, y^ξ is the second-stage decision capturing operational responses after demand ξ are realized.

note that $q^\top y^\xi$ represent the cost associated with second-stage decisions under demand realization ξ (scenario).

To solve a 2SP problem, we can use its extended deterministic version (also called Deterministic Equivalent), which explicitly enumerates all scenarios in a single large-scale mixed-integer program. In practice, due to the exponential growth in problem size with the number of scenarios, several alternative approaches are commonly used, including the sample average approximation (SAA) (Kleywegt *et al.*, 2002) and decomposition techniques such as Benders decomposition and the L-shaped method (Van Slyke & Wets, 1969).

In the next subsection, we present the formulation of our problem under the following assumptions:

- I. Material characteristics: All PET bottles collected from end-users are assumed to be of the same color. Therefore, color sorting does not influence the demand, processing or routing of PET bales,
- II. Network structure: The number of facilities and their potential locations are predetermined. Each facility belongs to a specific echelon (e.g., CC, PC, TC, RC, filament customers, DPC, markets and end-users),
- III. Facility location: The possible sites for locating facilities at each echelon are predefined in advance

3.2. Mathematical modeling

This section presents the mathematical model developed to address the problem. The key components, including indices, variables and parameters, are systematically defined and summarized in Table 2. Subsections 3.2.1 and 3.2.2 describe the objective functions and the constraints that govern the model.

3.2.1. Objective functions The objective is to minimize the strategic cost related to the facilities' opening cost and the operational cost included in the strategic decision under different demand scenarios.

And operationally, we have:

1. The facility opening cost

$$f_1 = \sum_i FU_i u_i + \sum_j FY_j y_j + \sum_k FV_k v_k + \sum_t FX_t x_t + \sum_l FG_l g_l \quad (3)$$

Where u_i , y_j , v_k , x_t and g_l are binary variable and FU_i , FY_j , FV_k , FX_t and FG_l are fixed opening costs.

TABLE 2 Essential notations for the suggested optimization model

Indices	
$b = 1, \dots, B$	Bale Types (TB) at processing facilities
$e = 1, \dots, E$	End-user site locations
$i = 1, \dots, I$	Possible locations for 3D printing centers (DPC)
$j = 1, \dots, J$	Possible locations for treatment facilities (TC)
$k = 1, \dots, K$	Possible locations for recycling facilities (RC)
$l = 1, \dots, L$	Possible locations for collection centers (CC)
$m = 1, \dots, M$	Market site locations
$n = 1, \dots, N$	Filament customer (FC) site locations
$\xi = 1, \dots, \Xi$	Index of scenarios
$t = 1, \dots, T$	Possible locations for processing facilities (PC)
Parameters	
FU_i	The fixed cost of the i -th DPC
FY_j	The fixed cost of the j -th TC
FV_k	The fixed cost of the k -th RC
FX_t	The fixed cost of the t -th PC
FG_l	The fixed cost of the l -th CC
CV_k	Recycling cost per filament item at recycling facility (RC_k),
CF_{ki}	Shipping cost per filament item from recycling facility (RC_k) to 3D printing center (DPC_i),
CU_i	Printing cost per product item at 3D printing center (DPC_i),
CW_{ij}	Shipping cost per waste item from 3D printing center (DPC_i) to treatment center (TC_j),
CY_j	Treatment cost per product item at treatment center (TC_j)
CS_{jk}	Shipping cost per PET or waste flake item from treatment center (TC_j) to recycling facility (RC_k)
CX_t	Sorting and processing cost per product item at processing center (PC_t)
CR_{btj}	Shipping cost per Bale (TB_b) item from processing center (PC_t) to treatment center (TC_j)
CG_l	Collection cost per product item at collection center (CC_l)
CZ_{lt}	Shipping cost per separated PET bottle waste item from collection center (CC_l) to processing center (PC_t)
SC_k	Cost savings per recycled filament item shipped from recycling facility (RC_k) to 3D printing center (DPC_i)
CP_{in}	Shipping cost per printed object from 3D printing center (DPC_i) to filament customer (FC_n)
CB_{el}	Shipping cost per unit of PET bottle waste from end-user e to collection center (CC_l)
DM_m^ξ	Demand for Bale Type (TB_b) at market m in scenario ξ
CD_{nj}	Shipping cost per waste item from filament customer (FC_n) to treatment center (TC_j)
DC_n^ξ	Demand for 3D-printed objects at filament customer (FC_n) in scenario ξ
CQ_{tm}	Shipping cost per Bale Type (TB_b) from processing center (PC_t) to market m
θ_b	Percentage of Bale Type (TB_b) sorted at processing center (PC_t)
γ_l	Proportion of waste in PET bottles collected after the separation process at CC_l ,
α_n	Proportion of waste generated at filament customer (FC_n),

(Continued)

TABLE 2 *Continued.*

Parameters	
λ_i	Waste PROPortion at DPC_i ,
β_t	Treatment capacity percentage of processing center (PC_t) for PET bottles
ρ_k	Filament production rate at recycling facility (RC_k)
$CapU_i$	Capacity of DPC_i ,
$CapY_j$	Capacity of TC_j ,
$CapV_k$	Capacity of RC_k ,
$CapX_t$	Capacity of PC_t ,
$CapG_l$	Capacity of CC_l ,
TP_e	Amount of PET bottles discarded by end-user e .
P_ξ	Probability occurrence of scenario ξ
$penalty_demand$	Penalty cost for unmet demands for each product
Decision variable	
First-stage decision variables	
u_i	1 If the 3D DPC_i is opened at the location; 0 otherwise,
y_j	1 If TC_j is opened at the location; 0 otherwise,
v_k	1 If RC_k is opened at the location; 0 otherwise,
x_t	1 If PC_t is opened at the location; 0 otherwise,
g_l	1 If CC_l is opened at the location; 0 otherwise.
Second-stage decision variables	
f_{ki}^ξ	Quantity of filaments moved from RC_k to DPC_i in scenario ξ [tons],
p_{in}^ξ	Quantity of printed objects moved from DPC_i to FC_n in scenario ξ [tons],
d_{nj}^ξ	Amount of waste moved from FC_n to TC_j in scenario ξ [tons],
w_{ij}^ξ	Amount of waste moved from DPC_i to TC_j in scenario ξ [tons],
s_{jk}^ξ	Quantity of PET flakes moved from TC_j to RC_k in scenario ξ [tons],
r_{btj}^ξ	Amount of Bales moved from PC_t to TC_j in scenario ξ [tons],
q_{tm}^ξ	Amount of Bales moved from PC_t to market m [tons],
z_{lt}^ξ	Amount of separated PET bottle waste moved from CC_l to PC_t [tons],
a_{el}^ξ	Amount of PET bottles moved from end-user e to CC_l [tons],
dm_m^ξ	Amount of unmet demands for market [tons]
dc_n^ξ	Amount of unmet demands for customers [tons]

2. Operational cost, which depends on the scenario ξ .

2.a. Transportation cost among facilities.

$$\begin{aligned}
 f_{2\xi} = & \sum_k \sum_i CF_{ki} f_{ki}^\xi + \sum_i \sum_n CP_{in} p_{in}^\xi + \sum_n \sum_j CD_{nj} d_{nj}^\xi + \sum_i \sum_j CW_{ij} w_{ij}^\xi + \sum_j \sum_k CS_{jk} s_{jk}^\xi \\
 & + \sum_b \sum_t \sum_j CR_{btj} r_{btj}^\xi + \sum_t \sum_m CQ_{tm} q_{tm}^\xi + \sum_l \sum_t CZ_{lt} z_{lt}^\xi + \sum_e \sum_l CB_{el} a_{el}^\xi
 \end{aligned} \quad (4)$$

2.b. Production Costs

$$\begin{aligned}
f_{3\xi} = & \sum_i CU_i \left(\sum_n p_{in}^\xi + \sum_j w_{ij}^\xi \right) + \sum_k \sum_j CY_j s_{jk}^\xi + \sum_i \sum_k CV_k f_{ki}^\xi \\
& + \sum_t \sum_l CG_l z_{lt}^\xi + \sum_t CX_t \left(\sum_j r_{btj}^\xi + \sum_m q_{tm}^\xi \right)
\end{aligned} \quad (5)$$

2.c. penalty costs for unsatisfied demand

$$f_{4\xi} = \sum_m \text{penalty}_{demand} \times dm_m^\xi + \sum_n \text{penalty}_{demand} \times dc_n^\xi \quad (6)$$

The complete objective function is then given by:

$$Z = f_1 + \sum_\xi P_\xi (f_{2\xi} + f_{3\xi} + f_{4\xi}) \quad (7)$$

The recourse function value $\sum_\xi P_\xi (f_{2\xi} + f_{3\xi} + f_{4\xi})$ represents the average cost needed to make the first-stage decision to adapt to uncertainty.

3.2.2. *Model constraints* Below are the key constraints of the model:

$$\sum_k f_{ki}^\xi \leq CapU_i \times u_i \quad \forall i \in I, \forall \xi \in \Xi, \quad (8)$$

$$\sum_b \sum_t r_{btj}^\xi \leq \beta_j \times CapY_j \times y_j \quad \forall j \in J, \forall \xi \in \Xi, \quad (9)$$

$$\sum_n d_{nj}^\xi + \sum_i w_{ij}^\xi \leq (1 - \beta_j) \times CapY_j \times y_j \quad \forall j \in J, \forall \xi \in \Xi, \quad (10)$$

$$\sum_j s_{jk}^\xi \leq \frac{1}{\rho_k} \times CapV_k \times v_k \quad \forall k \in K, \forall \xi \in \Xi, \quad (11)$$

$$\sum_l z_{lt}^\xi \leq CapX_t \times x_t \quad \forall t \in T, \forall \xi \in \Xi, \quad (12)$$

$$\sum_e a_{el}^\xi \leq CapG_l \times g_l \quad \forall l \in L, \forall \xi \in \Xi, \quad (13)$$

$$\sum_l a_{el}^\xi \leq TP_e \quad \forall e \in E, \forall \xi \in \Xi. \quad (14)$$

$$\sum_i f_{ki}^\xi \leq \rho_k \times \sum_j s_{jk}^\xi \quad \forall k \in K, \forall \xi \in \Xi, \quad (15)$$

$$\sum_n p_{in}^\xi \times \frac{1}{(1 - \lambda_i)} \leq \sum_i f_{ki}^\xi \quad \forall i \in I, \forall \xi \in \Xi, \quad (16)$$

$$\sum_k s_{jk}^\xi \leq \sum_n d_{nj}^\xi + \sum_i w_{ij}^\xi + \sum_t r_{tj}^\xi \quad \forall j \in J, \forall \xi \in \Xi, \quad (17)$$

$$\sum_j r_{tj}^\xi + \sum_m q_{tm}^\xi \leq \sum_l z_{lt}^\xi \quad \forall t \in T, \forall \xi \in \Xi, \quad (18)$$

$$\sum_t z_{lt}^\xi \leq (1 - \gamma_l) \times \sum_e a_{el}^\xi \quad \forall l \in L, \forall \xi \in \Xi, \quad (19)$$

$$\sum_j d_{nj}^\xi \leq \alpha_n \times \sum_i p_{in}^\xi \quad \forall n \in N, \forall \xi \in \Xi, \quad (20)$$

$$\sum_j w_{ij}^\xi \leq \frac{\lambda_i}{(1 - \lambda_i)} \times \sum_n p_{in}^\xi \quad \forall i \in I, \forall \xi \in \Xi. \quad (21)$$

$$\sum_t q_{tm}^\xi \geq DM_m^\xi \quad \forall b \in B, \forall m \in M, \forall \xi \in \Xi, \quad (22)$$

$$\sum_i p_{in}^\xi \geq DC_n^\xi \quad \forall n \in \bar{N}, \forall \xi \in \Xi. \quad (23)$$

$$u_i, y_j, v_k, x_t, g_l \in \{0, 1\} \quad \forall i, j, k, t, l, \quad (24)$$

$$f_{ki}^\xi, p_{in}^\xi, d_{nj}^\xi, w_{ij}^\xi, s_{jk}^\xi, r_{tj}^\xi, q_{bm}^\xi, z_{lt}^\xi, a_{el}^\xi \geq 0 \quad \forall i, j, k, t, l, n, b, t, e, \xi. \quad (25)$$

Constraints (8)–(13) account for the limited capacity of each center, ensuring that the quantities of products delivered to DPC_i , TC_j , RC_k , PC_t and CC_l do not exceed the respective capacity limits. Additionally, Constraint (14) ensures that the amount of PET bottles delivered by the end-users to CC_l remains within the capacity limits of end-user e . The quantity of products sent to a center must be less than or equal to the quantity received. Constraints (15)–(21) implement flow-balance constraints, ensuring that the quantity of products and materials entering each facility equals the quantity leaving it. Accordingly, Constraint (15) ensures that the amount of filament shipped from RC_k does not exceed the material quantity received from TC_j . Similarly, Constraints (16)–(19) regulate the flow of materials in DPC_i , TC_j , RC_k , PC_t and CC_l . Additionally, Constraints (20)–(21) specify the amount of waste transported from DPC_i and FC_n to TC_j . Constraints (22)–(23) meet the demand for both 3D-printed customers and markets. Constraints (24)–(25) present positive and binary variables.

4. The value of flexibility

In order to show the value of including uncertainty in CLSC, we considered the test instance derived from the realistic scenarios of a CLSC network, as presented in [Rajabi-Kafshgar et al. \(2025\)](#), focusing on 3D printing and recycling operations. The test problem has four 3D printing centers, four TC, eight RC, six PC and four CC for a total $\{2^{26}\}$ possible configurations.

For the stochastic CLSC problem, customer demand is assumed to follow a uniform distribution, fluctuating within $\pm 20\%$ to reflect market variability.

One important parameter impacting the need for a flexible configuration to hedge against uncertainty is the penalty cost for unmet demand. For simplicity, we considered a reference penalty cost equal to 100 for all the products, and we then evaluated the effect of increasing penalty costs up to 50%. The higher the

TABLE 3 *EVPI analysis based on the percentage increase in the amount of penalty cost*

Penalty increase	0%	20%	30%	40	50%
EVPI	17.4%	21.6%	23.7%	25.7%	27.8%

TABLE 4 *VSS analysis based on the percentage increase in the amount of penalty cost*

Penalty increase	0%	20%	30%	40%	50%
VSS	2.5%	3.4%	3.9%	4.3%	4.7%

penalty cost, the more conservative solutions are likely to incur high costs when demand increases. This means that to deal with uncertainty, we should employ solutions to hedge against an uncertain demand, balancing the risk of investments and operating costs with the risk of unmet demand.

The EVPI and VSS can be used to evaluate this trade-off and assess the value of flexibility. In decision theory, EVPI is the price one would pay to access perfect information about the future. VSS quantifies the benefit of using a stochastic model instead of an expected value (deterministic) approximation in decision-making problems under uncertainty (Birge, 1982). For this illustrative analysis, we use a three-scenario discretization (low: -20% , nominal: 0% , high: $+20\%$) of the $\pm 20\%$ uncertainty range. While this coarse discretization does not fully recover the underlying continuous distribution, it provides a tractable and interpretable basis for computing EVPI and VSS, and is sufficient to illustrate the qualitative value of incorporating uncertainty into the model. The full 100-scenario LHS distribution is used in the training and optimization experiments of Sections 5 and 6.

Tables 3 and 4 show the trends of EVPI and VSS under different penalty cost values. We can observe that EVPI and VSS are always greater than zero, and increase with the penalty value. These results emphasize the potential benefit of incorporating uncertainty into decision-making through stochastic programming.

5. Solution method

Incorporating uncertainties into optimization models captures real-world complexities but increases problem-solving challenges (Hosseini Baboli *et al.*, 2023). The flexibility of solutions improves with the number of scenarios, though at the cost of higher computational demands. Therefore, we need to balance computational feasibility and solution quality. To avoid the explosion of scenarios and to enhance the generalization flexibility, this paper proposes a surrogate model incorporating a neural network within the 2SP framework. The proposed neural network approximates the expected second-stage costs, which implicitly avoids the explicit computation of the recourse problem over all scenarios. This approach can improve computational efficiency and guarantee the flexibility of solutions to solve large-scale CLSC problems under uncertainty. Furthermore, the neural network allows for the prediction of complex relationships, such as the influence of demand variability on cost factors usually neglected by traditional models. This methodology offers a robust framework for the design of sustainable and efficient supply chains.

5.1. Scenario generation

The performance of the proposed surrogate-based 2SP framework depends critically on the quality and diversity of the training data. To ensure reproducibility and robustness, we detail here the procedures used to generate first-stage solutions (x) and demand scenarios (ξ).

5.1.1. Training data generation First-stage decision vectors x represent binary facility-opening choices across the five echelon types of the CLSC network: four DPCs, four TCs, eight RCs, six PCs and four CCs, resulting in an input dimensionality of 26 binary variables. To generate a diverse and representative training dataset, we applied LHS with the constraint that at least one facility of each type remains open to maintain network feasibility. A total of $N_{tr} = 1,000$ first-stage decision vectors were generated this way. We further enriched the sample space with 17 extreme configurations, both sparse (minimum open facilities) and dense (all facilities open), to ensure coverage of boundary cases, resulting in a final dataset of 1017 samples.

Each sampled first-stage vector x_i was then evaluated under 100 demand scenarios generated by LHS within the $\pm 20\%$ range described in Section 3. For every configuration, the second-stage recourse problem was solved independently for each scenario, and the expected cost $E_\xi [Q(x_i, \xi)]$ was estimated as the sample average:

$$E_\xi [Q(x_i, \xi)] \approx \frac{1}{100} \sum_{\xi=1}^{100} Q(x_i, \xi) \quad (26)$$

The resulting input–output pairs $(x_i, E_\xi [Q(x_i, \xi)])$, $i = 1, \dots, 1017$, form the surrogate model's training dataset. The dataset was split into 80% training (814 samples) and 20% validation (203 samples). Sensitivity analysis confirmed that this dataset size offers the best balance between model accuracy, coverage of the decision space, and computational effort.

5.1.2. Demand scenario generation Demand uncertainty was modeled by perturbing base-case demand levels by $\pm 20\%$, reflecting variability in AM demand as supported by industry reports (Protolabs, 2024). This approach balances computational tractability with the need to explore a range of demand outcomes in the optimization model. The same LHS procedure is used to generate scenarios for both the Neural Network-based Surrogate (NNS) training dataset and the SAA formulation, ensuring that both methods are exposed to scenarios drawn from the same underlying demand distribution and that the comparison in Section 6.1 is not biased by differences in scenario quality.

5.2. Neural network for second-stage cost approximation

The neural network takes as input the 26-dimensional binary first-stage decision vector $x \in \{0, 1\}^{26}$, representing the facility opening decisions across all echelon types. The output is a single scalar value $f_\theta(x) \in \mathbb{R}^+$ representing the approximated expected recourse cost $E_\xi [Q(x_i, \xi)]$. The network therefore implements a mapping $f_\theta : \{0, 1\}^{26} \rightarrow \mathbb{R}^+$. To approximate the expected recourse function $E_\xi [Q(x, \xi)]$, we employ a feedforward neural network regression model $f_\theta(x)$ with rectified linear unit (ReLU) activation function. The network is trained so that $f_\theta(x)$ closely matches the second-stage cost by minimizing the mean squared error loss function:

$$L(\theta) = \frac{1}{N_{tr}} \sum_{i=1}^{N_{tr}} (y_i - f_\theta(x_i))^2, \quad (27)$$

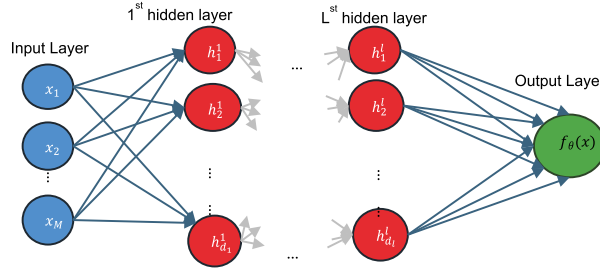


FIG. 3 Example of a 1-layer fully-connected neural network architecture diagram.

The training samples $\{x_i, \sum_{\xi} Q(x_i, \xi)\}$, $i = 1 \dots N_{tr}$, consist of randomly generated first-stage decisions x_i paired with their corresponding second-stage costs, computed over in-sample demand scenarios generated using the procedure described in Section 5.1. The vector θ denotes the neural network parameters. Once trained, the neural network serves as a surrogate for the expected recourse function and is ready to be embedded directly into the optimization model by translating $f_{\theta}(x)$ into a set of multilayer perceptron (MLP)-compatible linear constraints as detailed in Section 5.3. This integration enables the optimization algorithm to evaluate $E_{\xi} [Q(x, \xi)]$ rapidly, avoiding the need to repeatedly solve second-stage problems and thereby achieving substantial computational savings.

Fig. 3 shows an example of the neural network architecture where the learned mapping $f_{\theta}(x)$ is represented in a form compatible with Mixed-Integer Programming (MIP) constraints. It visualizes the layers and the connections between them. A neural network configuration is defined by its architecture (e, n_e) where e denotes the number of hidden layers and n_e denotes the number of neurons in each hidden layer. Such networks are widely used to approximate nonlinear mappings between input features and target responses, where the hidden layers apply nonlinear activation functions to capture complex relationships.

- **Input Layer ($l = 0$):** Contains m input nodes representing x_j . $j = 1, \dots, M$.
- **Hidden Layer l :** Contains d_k nodes fully connected to all nodes of Hidden Layer $l-1$, for $l = 1, \dots, L$.
- **Output Layer:** A single node representing $f_{\theta}(x)$, fully connected to all d_L nodes of Hidden Layer L . Mathematically, a L -layer fully-connected neural network can be expressed as:

In a fully connected neural network with M inputs, L hidden layers ($k = 1, \dots, L$) and d_k units in each layer, the unit j in the k -th layer can be expressed as follows:

$$h_j^1 = \sigma \left(\sum_{i=1}^{d_0} w_{ij}^0 x_i + b_j^0 \right) \quad i = 1, \dots, M, \quad j = 1, \dots, d_1; \quad (28)$$

$$h_j^l = \sigma \left(\sum_{i=1}^{d_{l-1}} w_{ij}^{l-1} h_i^{l-1} + b_j^{l-1} \right); \quad i = 1, \dots, d_{l-1}, \quad j = 1, \dots, d_l; \quad (29)$$

$$f(x) = \sum_{j=1}^{d_L} w_j^L h_j^L + b^L. \quad j = 1, \dots, d_L \quad (30)$$

Here, $x \in \mathbb{R}^m$ is the input, $f(x) \in \mathbb{R}$ is the scalar prediction, $h^l \in \mathbb{R}^{d_l}$ is the output of l -th hidden layer, $W^l \in \mathbb{R}^{d_{l+1} \times d_l}$ is the matrix of weights from layer l to $l+1$, $b^l \in \mathbb{R}^{d_{l+1}}$ is the bias at the l -th layer, and σ is a non-linear activation function (here, $\text{ReLU}_{(h)} = \max\{0, h\}$).

5.3. Neural network embedding in MILP

After training the neural network to approximate the expected recourse function $E_\xi [Q(x, \xi)]$, the next step is to integrate this surrogate model into the optimization framework. To embed the neural network into the optimization model, the learned mapping $f_\theta(x)$ is represented using MIP-compatible constraints. Each ReLU activation is modeled using the “big-M” formulation to ensure the neural network’s outputs align with its predictions. The resulting MILP incorporates the neural network as a surrogate for the second-stage cost:

$$\min_{x, \hat{y}} \left\{ c^\top x + \hat{y} \right\}, \quad (31)$$

where $\hat{y} = f_\theta(x)$ is implemented using linear constraints that approximate ReLU activations. Each hidden layer in the neural network is expressed as:

$$h_j^k = \text{ReLU} \left(\sum_j^{d_l} W_{ij}^{k-1} h_i^{k-1} + b_j^{k-1} \right), \quad (32)$$

The ReLU function is modeled through mixed-integer constraints (Fischetti & Jo, 2018).

$$\sum_i^{d_{l-1}} W_{ij}^{l-1} \hat{h}_i^{l-1} + b_j^{l-1} = \hat{h}_j^l - \check{h}_j^l \quad (33)$$

$$\hat{h}_j^l \leq M z_j^l, \quad (34)$$

$$\check{h}_j^l \leq M (1 - z_j^l), \quad (35)$$

$$\hat{h}_j^l, \check{h}_j^l \geq 0, \quad (36)$$

$$z_j^l \in \{0, 1\}, \quad (37)$$

where z_j^l is a binary variable indicating whether the ReLU is active. The performance of the MILP embedding depends on the calibration of the big-M constants used in the ReLU formulation. Excessively large M values lead to weak linear relaxations, while overly small values may exclude feasible solutions. In this study, we relied on Gurobi’s built-in ML embedding framework, which automatically computes big-M bounds from neuron weights and input ranges. Conceptually, the bounds on each neuron’s pre-activation value can be inferred from the feasible ranges of its inputs. This ensures that M is sufficiently tight to preserve solver efficiency, consistent with optimization-based bound tightening approaches (Fischetti & Jo, 2018). Figure 4 is a Framework overview diagram illustrating the steps of the proposed methodology.

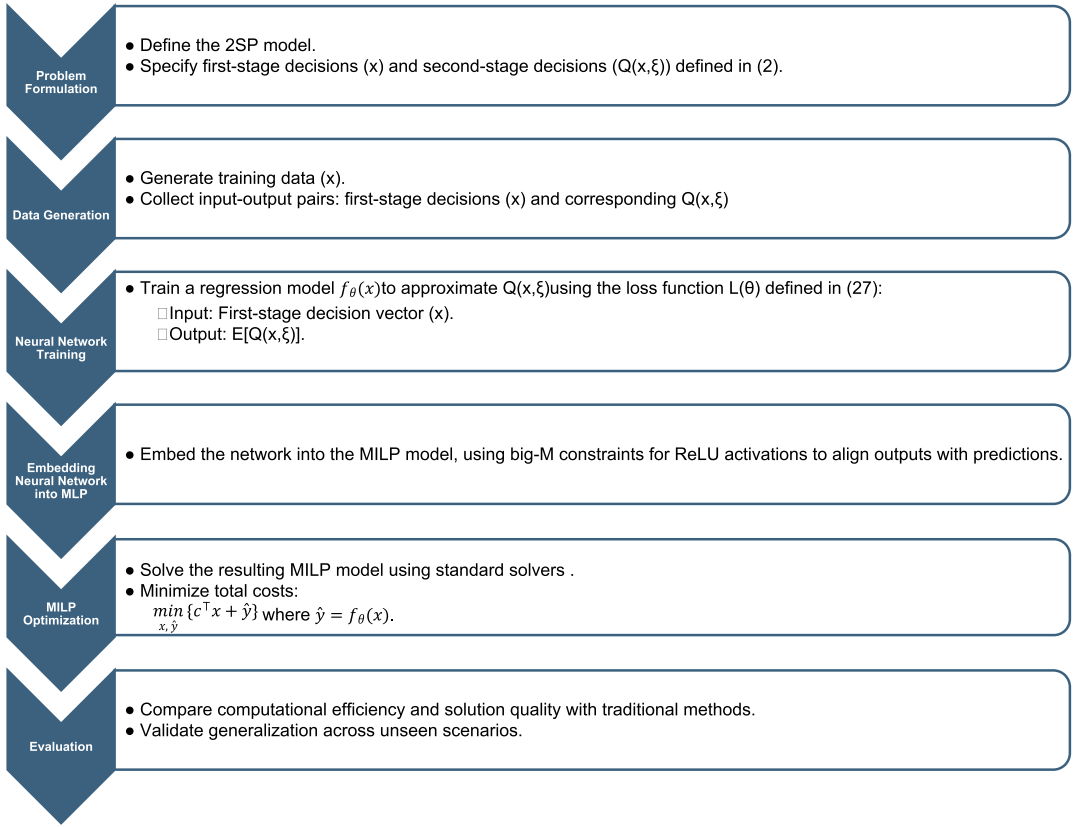


FIG. 4 Embedded model framework overview diagram.

5.4. Neural network architecture selection and sensitivity analysis

In this section, we evaluate the chosen neural network to ensure it provides an accurate and computationally efficient model for integration into the optimization framework. We tested several network architectures, and experimentally, the 7×18 configuration achieved the best overall performance. Here, the notation 7×18 refers to an MLP with seven hidden layers, each composed of 18 neurons, followed by a single output node that estimates the expected recourse cost. This deep yet compact architecture provided an effective balance between predictive accuracy and computational tractability. As summarized in Table 5, this structure achieved r^2 value of 0.921, indicating excellent predictive accuracy, while maintaining a relatively low Mean Absolute Percentage Error (MAPE) of 10.13%. These metrics confirm the model's reliability in effectively approximating target values. The 7×18 structure also balances accuracy and computational efficiency, generating 657 decision variables (C.Var) and 132 constraints when integrated into the optimization model. In this paper, we use a 18×7 model that has been depicted through good search.

Table 5 summarizes the performance of eight neural network configurations, varying the number of hidden layers (3, 5, 7, 9) and neurons per layer (10, 14, 18, 22). Several observations can be made. First, predictive accuracy (r^2) improves consistently as network depth and width increase, from 0.81 for the

TABLE 5 *Performance of neural networks*

NN structure	r^2	MAPE (%)	C.Var	Constr.	Opt. Time (s)
3×10	~ 0.81	~ 18.5	171	42	~ 45
3×18	~ 0.85	~ 15.2	297	66	~ 62
5×10	~ 0.87	~ 13.8	285	70	~ 75
5×14	~ 0.89	~ 12.1	399	98	~ 88
5×18	~ 0.90	~ 11.4	513	114	~ 95
7×18	0.921	10.13	657	132	~ 110
7×22	~ 0.923	~ 9.98	801	160	~ 185
9×18	~ 0.924	~ 9.85	855	174	~ 310

Bold values indicate the selected neural-network structure and its corresponding performance results.

smallest configuration (3×10) to 0.924 for the largest (9×18). However, this improvement becomes marginal beyond the 7×18 configuration, which achieves $r^2 = 0.921$ and $\text{MAPE} = 10.13\%$. Second, the number of decision variables and constraints introduced into the MILP grows linearly with network size, directly impacting the optimization time. The 9×18 configuration, while marginally more accurate, requires approximately 310 seconds to solve—nearly three times the optimization time of the 7×18 configuration (110 seconds)—without meaningful accuracy gains. Third, shallower networks (3×10 , 3×18) are fast to optimize but suffer from insufficient predictive accuracy, with MAPE values exceeding 15%. Based on this analysis, the 7×18 configuration represents the optimal trade-off between predictive accuracy, MILP tractability and computational efficiency, and is therefore selected for all subsequent experiments.

Detailed diagnostic plots including the learning curve, predicted vs. actual scatter plot, residual distribution, Q-Q plot and kernel density estimation are provided in [Appendix A](#). These plots confirm that the selected 7×18 architecture achieves unbiased predictions with normally distributed residuals and strong alignment between predicted and actual values.

5.5. Neural network surrogate performance

We embedded the NN model in the 2SP problem, and we solved it as shown in [Table 6](#). The table demonstrates its effectiveness as a surrogate model for estimating the expected recourse function $E_{\xi} [Q(x, \xi)]$ in Equation (3) for the small case problem with (4 DPCs, 4 TCs, 8 RCs, 6 PCs, 4 CCs). The model achieves a Best Objective and Best Bound of 29,729.58 in terms of the recourse function with a 0.00% optimality gap, indicating high precision. The total solving time of 141.15 seconds is highly efficient, and the approximation error Δ which is measured as the percentage difference between the surrogate solution and actual optimal recourse function costs over 100 out-of-sample scenarios, is only 2%.

The total online time of 141.15 seconds comprises approximately 30 seconds of neural network training and 110 seconds of MILP optimization. The scenario generation phase represents a one-time offline investment of approximately 5085 seconds, which is amortized across all subsequent uses of the trained surrogate. In contrast, SAA requires 409.60 seconds for the small instance with only 10 scenarios, and would require scenario re-generation for every new problem instance.

TABLE 6 *NNS performance in optimization*

Component	Time (s)
Scenario generation	~5085 (one-time offline)
Neural network training	~30
MILP optimization	~110
Best Objective ($E_{\xi} [Q(x, \xi)]$)	29,729.58
Best Bound	29,729.58
Gap (%)	0.00
Δ (%)	2.0
Total online time (training + optimization)	141.15

6. Computational results

Throughout this section, computational efficiency is measured in terms of wall-clock time, decomposed into three components: (i) scenario generation time—the one-time offline cost of solving all second-stage recourse problems needed to build the NNS training dataset (1017 first-stage vectors $\times 100$ scenarios = $101,700$ second-stage solves, requiring approximately 5085 seconds in total); (ii) neural network training time which is the time to train the surrogate model on the generated dataset, approximately 30 seconds for the 7×18 architecture; and (iii) optimization time which is the time to solve the MILP with the embedded neural network, which is the relevant online decision-making cost. For SAA, the reported time covers scenario generation and problem solving only, as no training phase is involved. Importantly, the scenario generation cost for NNS is a one-time investment: once the surrogate is trained, it can be reused for any new first-stage configuration without resolving any second-stage problems, making the optimization time the primary metric for online comparison.

To validate the effectiveness of our NNS approach, we compare it against the SAA, which estimates the expected recourse function $E_{\xi} [Q(x, \xi)]$ using a finite set of N scenarios. This comparison evaluates the trade-off between solution quality and computational time. All experiments were implemented in Python 3.9, using TensorFlow 2.13 for the neural network (MLP) and Gurobi 10.0.3 as the MILP solver. Experiments were conducted on a machine with an Intel Core i7-1355U CPU @ 1.70 GHz and 16 GB RAM. Early stopping was applied during neural network training to prevent overfitting. The overall performance of the MILP model depends heavily on the quality of the trained neural network, emphasizing the need to include rare and edge-case scenarios in the training data for stability and reliability.

6.1. Comparative analysis

Building on the single-instance benchmark of Section 5.5, we now extend the comparison to three problem instances of increasing size to evaluate scalability. We evaluated the performance and scalability of the NNS approach against SAA across three problem instances of increasing size: Small, Medium and Large (Table 7). Each instance differs in the number of facilities and demand variability, with SAA. SAA is implemented with $N = 10$ scenarios, generated via the same LHS procedure described in Section 3. This value of N represents the practical limit beyond which SAA becomes computationally intractable for the medium and large instances within a two-hour time limit, and thus constitutes a fair baseline for evaluating the computational advantage of the NNS approach.

TABLE 7 *Performance Comparison of NNS and SAA ($N = 10$, LHS scenarios) across Problem Sizes. NNS times include training and optimization; SAA times include scenario generation and solving. SAA failed to converge for the large instance within the 2-h time limit*

Test problem	Method	Objective value (\$)	Time (s)
Small	NNS	30093.59	110.40
Small	SAA	33,038.10	409.60
Medium	NNS	45,300.45	177.3
Medium	SAA	49,000.80	2056.36
Large	NNS	69,482.27	231.10
Large	SAA	—	—

To ensure a fair and transparent comparison, the SAA implementation follows the same scenario generation procedure described in Section 3. Demand scenarios are generated using LHS within the $\pm 20\%$ range around base-case demand values. For each problem instance, SAA constructs a deterministic equivalent formulation by explicitly enumerating all $N = 10$ scenarios, resulting in a single large-scale MILP whose size scales with N . The value $N = 10$ was selected as the maximum scenario budget under which SAA remains computationally tractable for all three problem instances within the two-hour time limit: preliminary experiments showed that $N = 20$ caused SAA to exceed this limit even for the small instance. The total number of variables in the SAA formulation is therefore N times the number of second-stage variables plus the number of first-stage variables, which for the small instance amounts to $10 \times [\text{second-stage variables}] + 26$ binary variables. It is important to note that both NNS and SAA are evaluated on the same set of 100 out-of-sample scenarios generated independently from the training and SAA scenario sets, ensuring a fair comparison of solution quality.

- Small: 4 DPCs, 4 TCs, 8 RCs, 6 PCs, 4 CCs; demand $\pm 20\%$
- Medium: 6 DPCs, 5 TCs, 8 RCs, 10 PCs, 6 CCs; demand $\pm 20\%$
- Large: 12 DPCs, 10 TCs, 18 RCs, 13 PCs, 10 CCs; demand $\pm 20\%$

For NNS, reported times include neural network training (~ 30 s) and MILP optimization (~ 110 s for small, ~ 177 s for medium, ~ 231 s for large), but exclude the one-time offline scenario generation cost (~ 5085 s). For SAA, reported times include scenario generation and problem solving with 10 scenarios. The results indicate that the NNS approach achieves lower objective values than SAA across all tested problem sizes, with computation times substantially lower than SAA. For the large instance, SAA failed to converge within the two-hour time limit, while NNS produced a solution in 231 s. These results should be interpreted in the context of the specific problem sizes tested, and generalization to larger instances remains a direction for future work.

6.2. Network configuration analysis

In this section, we analyze the network configuration of the small-scale instance to highlight key trends in facility utilization and cost allocation, since the medium and large instances exhibited similar patterns. The optimization model reveals significant insights into how resources are deployed across the network. The network configuration results presented here are derived from the optimal first-stage solution x^* produced by the NNS model for the small instance. Specifically, Fig. 5 shows the number

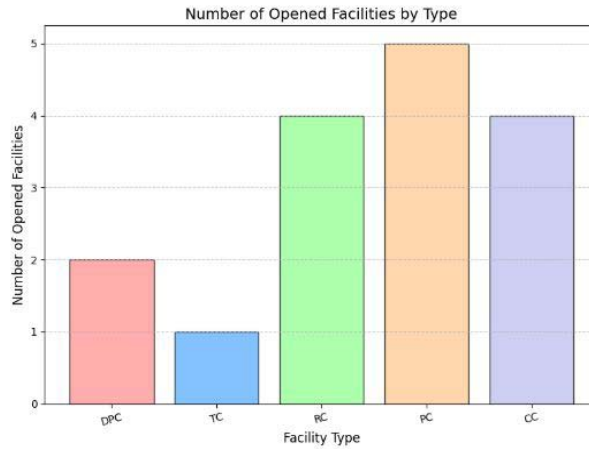


FIG. 5 Optimal number of open facilities by type.

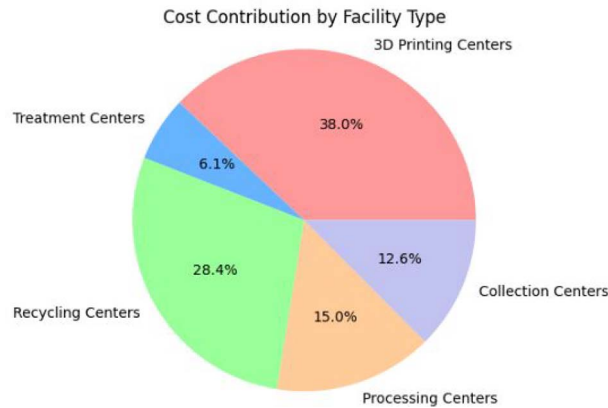


FIG. 6 Proportional cost contributions by facility type.

of facilities opened across each echelon type in x^* , while Fig. 6 shows the cost breakdown obtained by evaluating the second-stage recourse problem under the optimal solution x^* across the 100 out-of-sample scenarios described in Section 3. The expected costs are then decomposed by facility type to produce the proportional cost contributions. This analysis is intended to provide managerial insights into the structure of the optimal solution rather than to compare methods.

As shown in Fig. 5, PC lead in the number of facilities opened, highlighting their crucial role within the supply chain network. In contrast, fewer 3D printing centers (DPC) and TC indicate more strategic resource allocation. Figure 6 shows that DPC are the major cost drivers, accounting for 38% of total expenses. Recycling, processing and CC contribute moderately, while TCs have the smallest share at 6.1%. These insights emphasize the financial impact of DPC and the importance of targeted cost-reduction strategies.

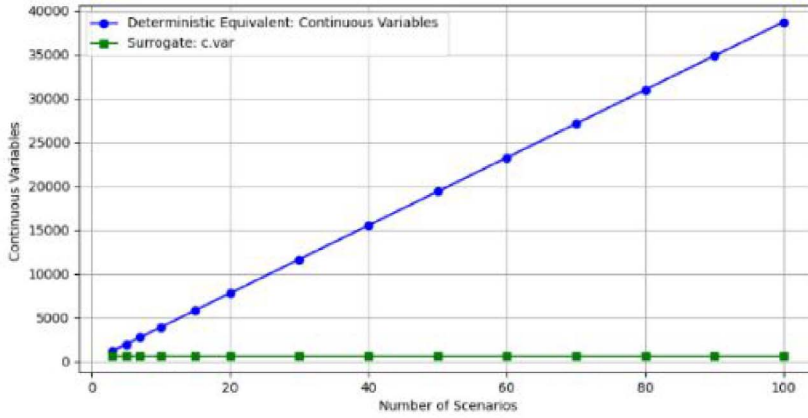


Fig. 7 Variables comparison: deterministic equivalent vs. surrogate.

6.3. Comparison of deterministic equivalent and NNS solution

We now compare the deterministic equivalent model and the neural network surrogate (NNS) in terms of scalability, computational efficiency and solution robustness. Figures 7 and 8 illustrate how the number of continuous variables and constraints grows with the number of scenarios. While both dimensions expand rapidly in the deterministic model, the surrogate method maintains a minimal, fixed size, demonstrating its superior scalability for large scenario sets. Figures 7 and 8 compare the dimensions of the deterministic equivalent and surrogate methods when the number of scenarios increases. Figure 7 shows the rapid growth of continuous variables in the deterministic equivalent model as scenarios increase, while the surrogate method remains constant. Figure 8 highlights a similar trend for constraints, with the surrogate method maintaining a minimal and fixed count. These results suggest that the surrogate method maintains near-constant problem dimensions as the number of scenarios increases, supporting its potential for scalability in larger scenario-based problems.

As shown in Fig. 9, the deterministic equivalent model exhibits exponential growth in computational time as the number of scenarios increases. For instance, the time starts below 100 s for a problem with three scenarios but exceeds 2000 s for a medium problem with 10 scenarios, demonstrating that the deterministic model becomes impractical for larger-scale applications. In contrast, the NNS model offers a significant gain in computational efficiency. Its total online time consists of two components: neural network training, which takes approximately 30 s for the 7×18 architecture and increases slightly as the number of scenarios grows due to the larger dataset, and MILP optimization, which remains constant at approximately 110 s regardless of the number of scenarios, since the surrogate decouples scenario evaluation from the optimization process. The one-time scenario generation cost of approximately 5085 seconds is excluded from Fig. 9 as it is an offline investment shared across all optimization queries using the same surrogate model.”

To compare solution quality fairly, we adopt the following out-of-sample evaluation procedure. First, both models are solved independently to obtain their respective optimal first-stage solutions: x_{DET}^* for the deterministic equivalent and x_{NNS}^* for the NNS model. The deterministic equivalent is solved using the same $N = 10$ scenarios used by SAA, while the NNS model uses the trained surrogate. Second, both solutions are evaluated on the same set of 100 out-of-sample scenarios, generated independently from any

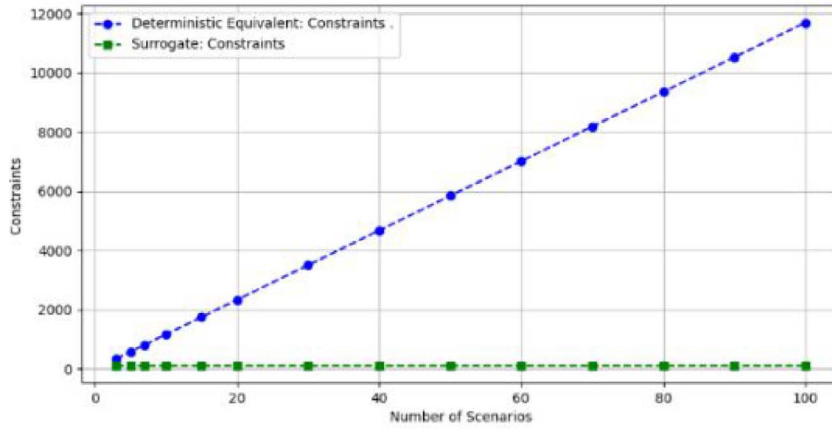


Fig. 8 Constraints comparison: deterministic equivalent vs. surrogate.

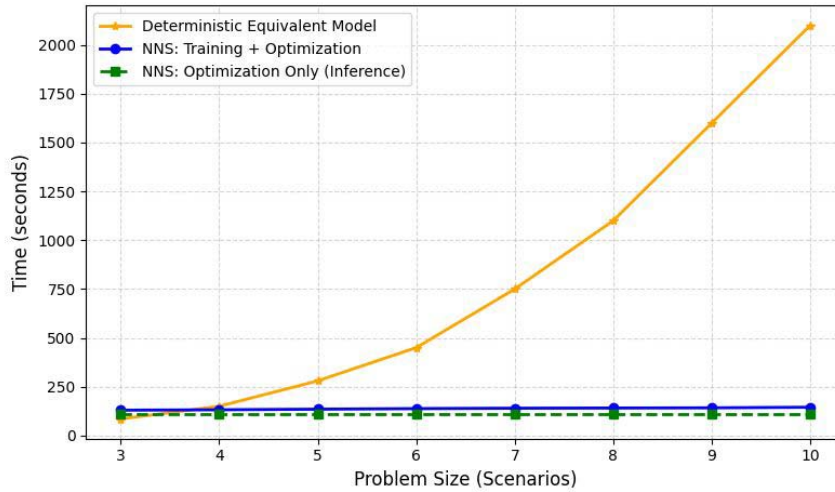


Fig. 9 Comparison of computational time vs. problem size.

training or solving data using the LHS procedure described in [Section 3](#). For each out-of-sample scenario ξ , the second-stage recourse problem is solved for both x_{DET}^* and x_{NNS}^* , yielding scenario-specific total costs. Third, the resulting cost distributions, each consisting of 100 cost observations, are compared using boxplots ([Fig. 10](#)), density plots ([Fig. 11](#)), and histograms ([Fig. 12](#)). This procedure ensures that both models are evaluated under identical conditions and that the comparison reflects true out-of-sample performance.

While the deterministic equivalent model evaluates a small number of scenarios, the surrogate model efficiently integrates a large number, capturing broader uncertainties and improving solution robustness. In contrast, the deterministic model may underestimate the impact of uncertainty due to its limited scenario space and becomes less reliable as the number of scenarios grows.

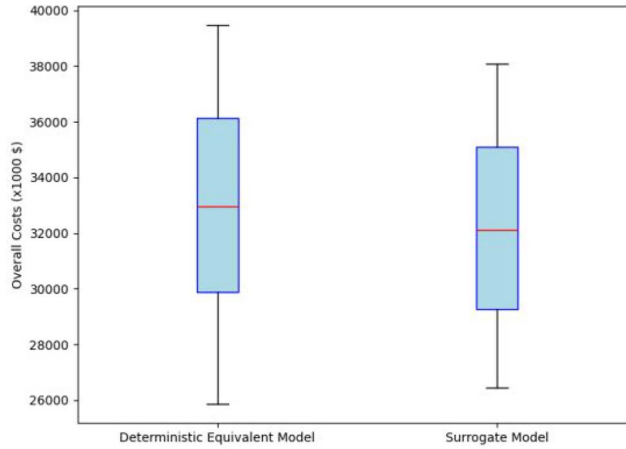


Fig. 10 Boxplot of overall costs: deterministic equivalent vs surrogate models.

Figure 10 presents a boxplot of overall cost variability across the 100 out-of-sample scenarios. While the median costs of the two models are similar which is reflecting the fact that both models solve the same underlying optimization problem, the NNS solution exhibits a narrower interquartile range and fewer high-cost outliers than the deterministic equivalent. This difference, while visually modest, is consistent with the VSS values reported in Tables 3 and 4 (2.5%–4.7%), which indicate that the benefit of stochastic modeling is real but moderate for this instance size. The primary advantage of the NNS model over the deterministic equivalent in this context is therefore not a dramatic reduction in expected cost, but rather improved robustness, reflected in lower cost variance, and substantially better computational scalability, as demonstrated in Fig. 9. Figure 11 compares the full cost distributions, emphasizing that the surrogate model achieves lower overall costs across scenarios. Figure 12 combines these distributions into a histogram, visually reinforcing the surrogate model’s efficiency and reduced cost variability relative to the deterministic approach. These figures indicate that the surrogate model achieves lower cost variability than the deterministic equivalent across the 100 out-of-sample scenarios considered, suggesting more consistent performance under uncertainty.

The flexibility of decision-making under uncertainty is a key advantage of the surrogate model. Figure 13 illustrates stark differences in resource allocation strategies. By leveraging neural networks to evaluate cost scenarios, the surrogate approach captures complex trade-offs between competing objectives, balancing fixed investment costs with operational flexibility.

The surrogate model combines neural network approximation with MILP optimization, offering a computationally tractable alternative to the deterministic equivalent for the CLSC instances examined in this study.

As summarized in Table 8, the comparison underscores the surrogate model’s superiority in addressing uncertainty within CLSCs. While the deterministic equivalent model offers cost-optimized solutions for limited scenarios, its inability to scale or adapt to extensive uncertainties reduces its practical utility. In contrast, the surrogate model integrates advanced ML to achieve scalable, robust and computationally efficient solutions, making it a compelling alternative for large-scale, uncertain CLSC problems.

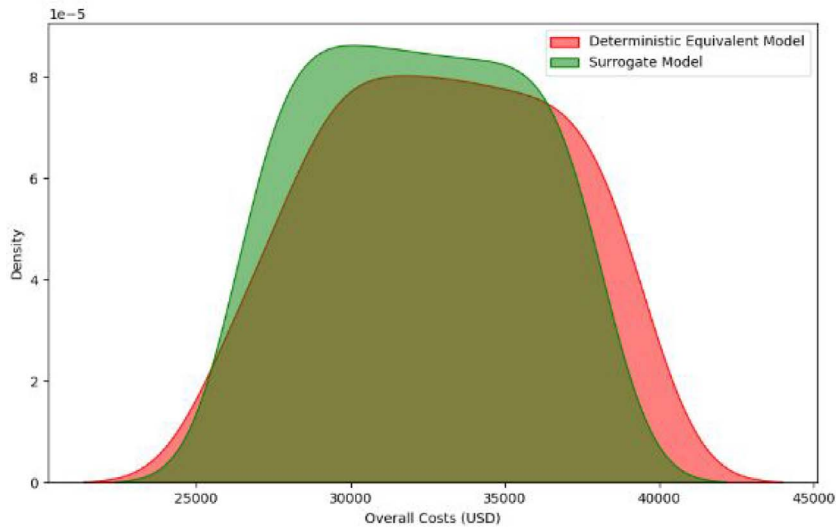


FIG. 11 Comparison of cost distributions: deterministic equivalent vs surrogate models.

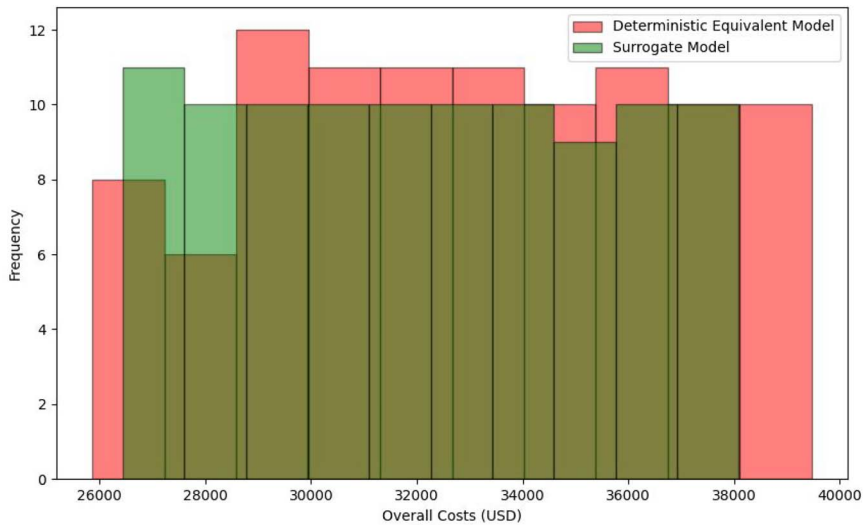


FIG. 12 Combined histogram: deterministic equivalent vs surrogate models.

6.4. Managerial implications

The results of the proposed model provide concrete policy and managerial insights that extend beyond methodological contributions:

6.4.1. Strategic Investment in Processing Centers (PCs) Figure 5 shows that PCs dominate the set of facilities opened across optimal solutions. This indicates that PCs act as central hubs connecting Collection Centers (CCs), TCs and the Market. From a managerial perspective, underinvestment in PCs

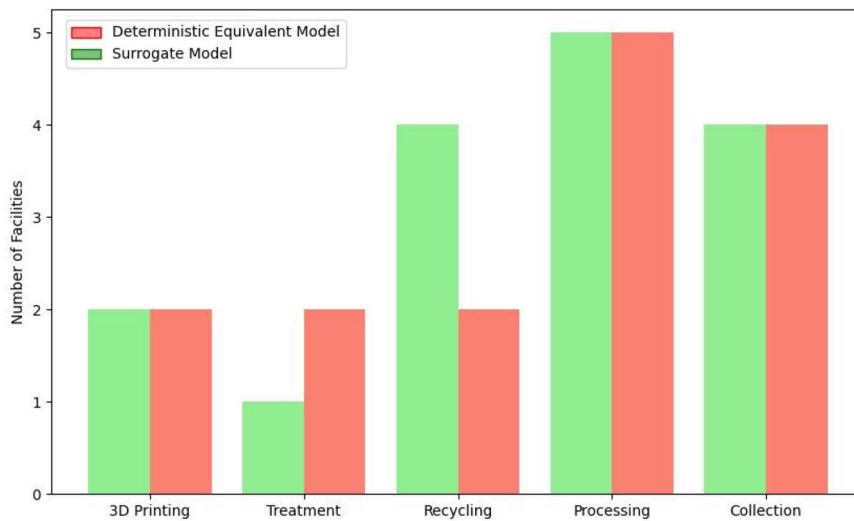


FIG. 13 Facility allocation comparison.

TABLE 8 Key comparisons: deterministic equivalent vs. surrogate models

Metric	Deterministic equivalent model	Surrogate model
Computational complexity	Exponential with scenarios	Near linear; reduced formulation
Solution robustness	Limited to evaluated scenarios	Robust across extensive scenarios
Cost	Cost-optimized limited to small scenarios	higher robustness cost
Adaptability	Limited flexibility; suboptimal trade-offs	High adaptability; nuanced solutions
Scalability	Poor; impractical for large problems	Excellent; scalable for real-world CLSC

could create bottlenecks, limiting overall network efficiency. Quantitative analysis suggests that ensuring sufficient capacity in PCs is critical to maintain system throughput and operational robustness under demand variability.

6.4.2. High-leverage cost management in 3D printing centers (DPCs) As shown in Fig. 6, DPCs contribute 38% of total system costs, representing the largest single cost component. This identifies DPCs as high-leverage points for cost reduction. Managers can prioritize interventions such as capacity sharing, selective outsourcing, process improvements or adoption of energy-efficient printing technologies. Targeting DPCs for cost reduction is expected to yield the largest marginal decrease in total system cost, reflecting a quantitative prioritization for managerial action.

6.4.3. Leveraging the Surrogate Model for Scenario Planning Figures 7–9 demonstrate that the surrogate model maintains near-constant computational effort as the number of scenarios increases, in

contrast to the exponential growth observed for the deterministic equivalent. This enables managers to perform extensive “what-if” analyses under different demand profiles, return rates or policy interventions (e.g., deposit-return schemes) in real time. Such capability allows for stress-testing of the network and evaluation of investment trade-offs, which would be computationally impractical with classical deterministic models.

6.4.4. Balancing Cost and Risk under Uncertainty Figures 10–12 show that the surrogate model consistently delivers lower variability and more stable cost outcomes compared to the deterministic equivalent. Managers can leverage this stability to allocate budgets more reliably, hedge against demand uncertainty and make investment decisions that are robust to stochastic fluctuations. These results underscore the value of integrating predictive analytics with optimization to support risk-informed decision-making.

6.4.5. Enabling Sustainability through Robust Planning The model highlights that maintaining adequate capacity in recycling and treatment facilities (RCs and TCs, Figs. 5 and 13) strengthens the circular economy performance of the network. Managers can use this insight to align investment decisions with sustainability targets, ensuring that reverse logistics flows are not capacity-constrained. By quantitatively linking facility design to environmental objectives, the model supports informed decisions that balance operational efficiency, cost and sustainability performance.

7. Conclusions

This study presents a neural network-based surrogate model for optimizing closed-loop supply chains in 3D printing and recycling. By combining ML with 2SP, the approach efficiently approximates second-stage costs, capturing non-linear cost relationships while significantly reducing computational complexity.

Compared to deterministic equivalent models, the surrogate demonstrates superior performance by accommodating a broader range of scenarios, improving decision-making under uncertainty and supporting resource allocation and facility configurations. Importantly, the model generalizes effectively to out-of-sample scenarios, providing consistent and robust solutions even under previously unseen demand variations. Empirical results show reductions in computation times relative to SAA across the tested instances, with NNS producing solutions within the time limit for all cases including the large instance where SAA failed to converge.

Key advantages of the proposed model include flexibility, scalability, adaptability and generalization capability, making it suitable for real-world applications characterized by uncertain demand and operational variability. By integrating neural networks with MILP, the approach addresses computational bottlenecks in stochastic optimization, offering a practical framework for data-driven, flexible supply chain design.

Funding

In the last stage of publishing the paper, Open access funding provided by Università degli Studi di Milano - Bicocca within the CRUI CARE Agreement.

Declaration of interests

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability statements

The data underlying this article will be shared on reasonable request to the corresponding author.

REFERENCES

- ADE IRAWAN, C., DAN-ASABE ABDULRAHMAN, M., SALHI, S. & LUIS, M. (2022) An efficient matheuristic algorithm for bi-objective sustainable closed-loop supply chain networks. *IMA J. Manag. Math.*, **33**, 603–636.
- AKBARI-AGHGHAEH, Z., MOZDGHIR, A., SEYEDI, I. & MESSINA, E. (2025) Designing a perishable closed-loop poultry supply chain: Metaheuristic approaches and model evaluation. *Environ. Dev. Sustain.*, 1–43. <https://doi.org/10.1007/s10668-025-06675-6>.
- ALABI, T. M., AGHIMIEN, E. I., AGBAJOR, F. D., YANG, Z., LU, L., ADEOYE, A. R. & GOPALUNI, B. (2022) A review on the integrated optimization techniques and machine learning approaches for modeling, prediction, and decision making on integrated energy systems. *Renew. Energy*, **194**, 822–849.
- AMIRTEIMOORI, A., KHOSHANDAM, L., KORDROSTAMI, S., JAHANI SAYYAD NOVEIRI, M. & MATIN, R. K. (2022) Performance analysis in a stochastic supply chain with reverse flows: A DEA-based approach. *IMA J. Manag. Math.*, **33**, 433–456.
- BADRI, H., GHOMI, S. F. & HEJAZI, T. H. (2017) A two-stage stochastic programming approach for value-based closed-loop supply chain network design. *Transp. Res. E: Logist. Transp. Rev.*, **105**, 1–17.
- BIÇE, K. & BATUN, S. (2021) Closed-loop supply chain network design under demand, return and quality uncertainty. *Comput. Ind. Eng.*, **155**, 107081.
- BIRGE, J. R. (1982) The value of the stochastic solution in stochastic linear programs with fixed recourse. *Math. Programming*, **24**, 314–325.
- CAÑADO, N., LIZUNDIA, E., AKIZU-GARDOKI, O., MINGUEZ, R., LEKUBE, B., ARRILLAGA, A. & ITURRONDOBEITIA, M. (2022) 3D printing to enable the reuse of marine plastic waste with reduced environmental impacts. *J. Ind. Ecol.*, **26**, 2092–2107.
- CARBONNEAU, R., LAFRAMBOISE, K. & VAHIDOV, R. (2008) Application of machine learning techniques for supply chain demand forecasting. *Eur. J. Oper. Res.*, **184**, 1140–1154.
- CHEN, P. B., LIN, G. H., XU, W. & ZHU, X. (2024) Supply chain network equilibrium with outsourcing for fresh agricultural products under stochastic demands. *IMA J. Manag. Math.*, **36**, dpae024.
- CHOWDHURY, S., SHAHVARI, O., MARUFUZZAMAN, M. & BIAN, L. (2019) Sustainable design of on-demand supply chain network for additive manufacturing. *IIE Trans.*, **51**, 744–765.
- FISCHETTI, M. & JO, J. (2018) Deep neural networks and mixed integer linear optimization. *Constraints*, **23**, 296–309.
- FLEISCHMANN, M., BLOEMHOF-RUWAARD, J. M., DEKKER, R., VAN DER LAAN, E., VAN NUNEN, J. A. & VAN WASSENHOVE, L. N. (1997) Quantitative models for reverse logistics: A review. *Eur. J. Oper. Res.*, **103**, 1–17.
- GAO, Y., LU, S., ZHAN, S., HU, C. & LIU, X. (2024) Closed-loop supply chain network design with price-greenness-sensitive demand: A distributionally robust chance-constrained optimization approach. *Comput. Oper. Res.*, **172**, 106803.
- GAULA, A. K. & JHA, J. K. (2024) Pricing strategy with quality improvement in a dual collection channel closed-loop supply chain under return uncertainty. *Oper. Res.*, **24**, 27.
- GIBSON, I., ROSEN, D. & STUCKER, B. (2015) ‘Direct digital manufacturing’. In: *Additive Manufacturing Technologies*. 2nd edn. Cham: Springer. New York, NY: Springer New York, pp. 375–397.
- GOODFELLOW, I., BENGIO, Y. & COURVILLE, A. (2016) *Deep Learning*. Cambridge, MA: MIT Press.
- HAN, B., SHI, S., PARK, Y. & XU, Y. (2024) A responsive closed-loop supply chain network design under demand uncertainty. *Comput. Ind. Eng.*, **192**, 110233.
- HOSSEINI BABOLI, S. A., ARABKOOHSAR, A. & SEYEDI, I. (2023) Numerical modeling and optimization of pressure drop and heat transfer rate in a polymer fuel cell parallel cooling channel. *J. Braz. Soc. Mech. Sci. Eng.*, **45**, 201.
- HOSSEINI, S., FARAHANI, R. Z., DULLAERT, W., RAA, B., RAJABI, M. & BOLHARI, A. (2014) A robust optimization model for a supply chain under uncertainty. *IMA J. Manag. Math.*, **25**, 387–402.

- KHALILI-FARD, A., PARSAEE, S., BAKHSHI, A., YAZDANI, M., AGHSAMI, A. & RABBANI, M. (2024) Multi-objective optimization of closed-loop supply chains to achieve sustainable development goals in uncertain environments. *Eng. Appl. Artif. Intell.*, **133**, 108052.
- KLEYWEGT, A. J., SHAPIRO, A. & HOMEM-DE-MELLO, T. (2002) The sample average approximation method for stochastic discrete optimization. *SIAM J. Optim.*, **12**, 479–502.
- LIU, S. & PAPAGEORGIOU, L. G. (2013) Multiobjective optimisation of production, distribution and capacity planning of global supply chains in the process industry. *Omega*, **41**, 369–382.
- LIU, Z., HU, L. M. & YEH, W. C. (2023) Risk-averse two-stage stochastic programming-based closed-loop supply chain network design under uncertain demand. *Appl. Soft Comput.*, **147**, 110743.
- LOZANO-OVIEDO, J., CORTÉS, C. E. & REY, P. A. (2024) Sustainable closed-loop supply chains and their optimization models: A review of the literature. *Clean Technol. Environ. Policy*, **26**, 999–1023.
- NASSEHI, A., ZHONG, R. Y., LI, X. & EPUREANU, B. I. (2022) Review of machine learning technologies and artificial intelligence in modern manufacturing systems. *Design and Operation of Production Networks for Mass Personalization in the Era of Cloud Technology* (D. MOURTZIS ed.). Amsterdam: Elsevier, pp. 317–348.
- PATEL, R. M., DUMOUCHELLE, J., KHALIL, E. & BODUR, M. (2022) Neur2SP: Neural two-stage stochastic programming. *Adv. Neural Inf. Process. Syst.*, **35**, 23992–24005.
- PRIAVOULOU, C., TROULLAKI, K., TSIOURIS, N., GIOTTSAS, C. & KOSTAKIS, V. (2022) Tracing sustainable production from a degrowth and localisation perspective: A case of 3D printers. *J. Clean. Prod.*, **376**, 134291.
- Protolabs. (2024) *3D Printing Trend Report*. Maple Plain, MN: Proto Labs, Inc. URL <https://www.protolabs.com/resources/guides-and-trend-reports/3d-printing-trend-report/>.
- RAJABI-KAFSHGAR, A., SEYEDI, I. & TIRKOLAEI, E. B. (2025) Circular closed-loop supply chain network design considering 3D printing and PET bottle waste. *Environ. Dev. Sustain.*, **27**, 20345–20381. <https://doi.org/10.1007/s10668-024-04767-3>.
- SANCHEZ, F. A. C., BOUDAUD, H., CAMARGO, M. & PEARCE, J. M. (2020) Plastic recycling in additive manufacturing: A systematic literature review and opportunities for the circular economy. *J. Clean. Prod.*, **264**, 121602.
- SEYEDI, I. & MALEKI-DARONKOLAEI, A. (2013) Solving a two-stage assembly flowshop scheduling problem to minimize the mean tardiness and earliness penalties by three meta-heuristics. *Casp. J. Appl. Sci. Res.*, **2**.
- SEYEDI, I., MIRZAZADEH, S., MALEKI-DARONKOLAEI, A., MUKHTAR, M. & SAHRAN, S. (2016) An inventory model with reworking and setup time to consider effect of inflation and time value of money. *J. Eng. Sci. Technol.*, **11**, 416–430.
- SEYEDI, I., CANDELIERI, A., MESSINA, E. & ARCHETTI, F. (2025) Wasserstein Distributionally robust optimization for chance constrained facility location under uncertain demand. *Mathematics.*, **13**, 2144. <https://doi.org/10.3390/math13132144>.
- SHAFIEE ROODBARI, E., FATEMI GHOMI, S. M. T. & EICKER, U. (2024) Designing a multi-objective closed-loop supply chain: A two-stage stochastic programming method applied to the garment industry in Montréal, Canada. *Environ. Dev. Sustain.*, **26**, 6131–6162.
- SHAHARUDIN, M. R., ZAILANI, S., TAN, K. C., CROSS, J. & HOTRAWAISAYA, C. (2023) Fostering closed-loop supply chain orientation by leveraging strategic green capabilities for circular economy performance: Empirical evidence from Malaysian electrical and electronics manufacturing firms. *Environ. Dev. Sustain.*, **27**, 24397–24434.
- SIMONETTO, M., SGARBOSSA, F., BATTINI, D. & GOVINDAN, K. (2022) Closed loop supply chains 4.0: From risks to benefits through advanced technologies. A literature review and research agenda. *Int. J. Prod. Econ.*, **253**, 108582.
- SUPPIPAT, S. & HU, A. H. (2022) A scoping review of design for circularity in the electrical and electronics industry. *Resour. Conserv. Recycl. Adv.*, **13**, 200064.
- TAVANA, M., KIAN, H., NASR, A. K., GOVINDAN, K. & MINA, H. (2022) A comprehensive framework for sustainable closed-loop supply chain network design. *J. Clean. Prod.*, **332**, 129777.
- VAN SLYKE, R. M. & WETS, R. (1969) L-shaped linear programs with applications to optimal control and stochastic programming. *SIAM J. Appl. Math.*, **17**, 638–663.
- WEICHERT, D., LINK, P., STOLL, A., RÜPING, S., IHLENFELDT, S. & WROBEL, S. (2019) A review of machine learning for the optimization of production processes. *Int. J. Adv. Manuf. Technol.*, **104**, 1889–1902.

XING, R., ZHUANG, M. & LI, B. (2024) For-profit or for-CSR: Decisions of power batteries recovery in the supply chain. *IMA J. Manag. Math.*, **37**, 411–444. <https://doi.org/doi.org/10.1093/imaman/dpae011>.

Appendix A Neural Network Diagnostic Plots

Figures A1–A6 present a comprehensive evaluation of the selected 7×18 neural network configuration. Figure A1 shows the learning curve, where training loss decreases rapidly and levels off, confirming convergence. Figure A2 shows a scatter plot of predicted vs. actual values, with points clustering closely around the ideal fit line, indicating strong predictive accuracy. Figure A3 presents the residual distribution, centered around zero with a bell-shaped curve, confirming that prediction errors are random and unbiased. Figure A4 shows the Q-Q plot of residuals, confirming approximate normality with minor deviations at the extremes. Figures A5 and A6 compare the distributions of actual and predicted values through overlapping histograms and kernel density estimation, respectively, both showing strong alignment between predicted and actual distributions.

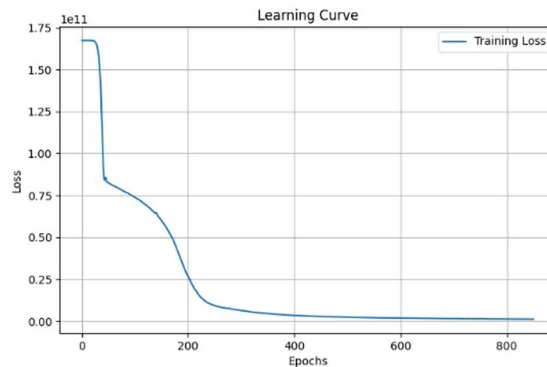


FIG. A1 Learning curve showing training loss over epochs.

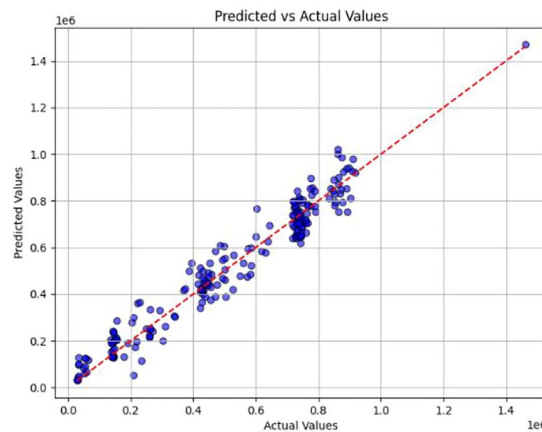


FIG. A2 Predicted vs actual values.

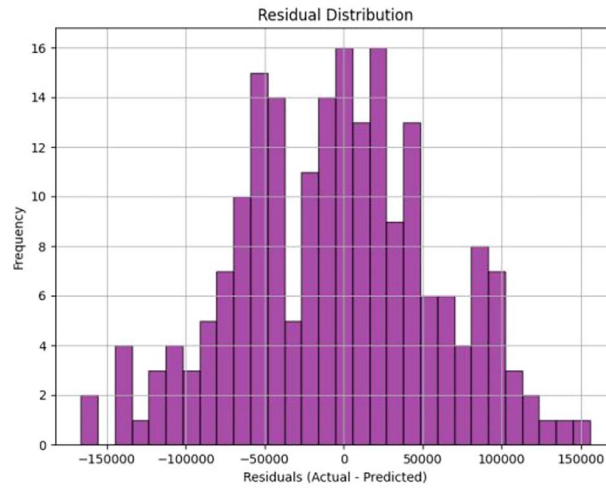


FIG. A3 Residual distribution of model predictions.

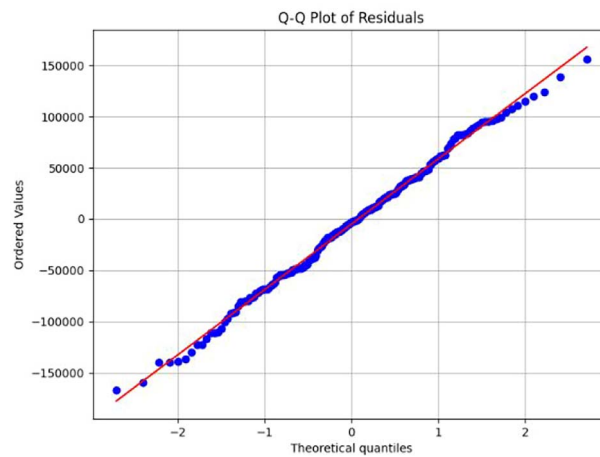


FIG. A4 Q-Q plot of residuals.

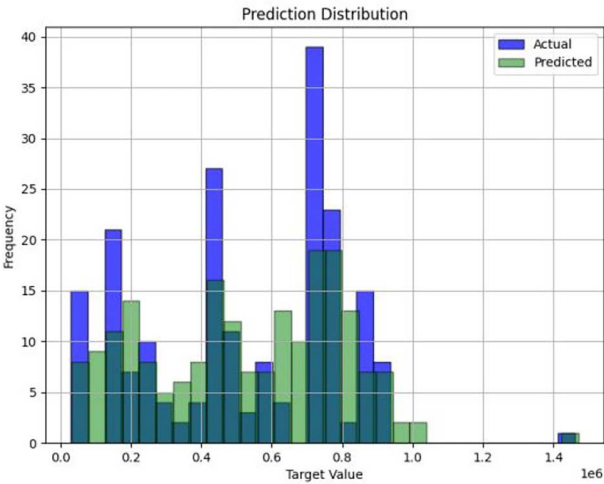


FIG. A5 Prediction distribution comparison.

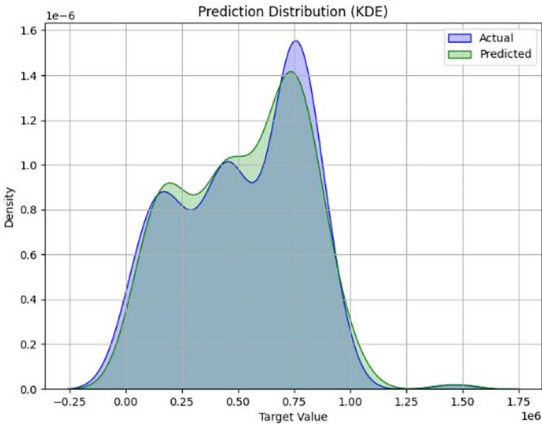


FIG. A6 Kernel density estimation of actual vs. predicted values.