Journal of Manufacturing Engineering, September 2026, Vol. 21, Issue. 3, pp 077-093
DOI. https.//doi.org/10.37255/jme.v21i3pp077-093

INTELLIGENT DECISION-MAKING IN SUSTAINABLE MANUFACTURING: COUPLING MACHINE LEARNING WITH MULTI-OBJECTIVE OPTIMIZATION FOR CARBON REDUCTION

Zhenguang Shao

Hisense International Co., Ltd, Qingdao, China.

Abstract

Smart manufacturing systems increasingly face the dual challenge of maximizing productivity while minimizing environmental impacts. However, existing approaches often treat emission forecasting and operational optimization as isolated tasks. This study proposes an integrated intelligent decision-making framework that couples machine learning with multi-objective optimization (MOO) to achieve eco-efficient production scheduling. In the predictive phase, Support Vector Regression (SVR) is hybridized with four advanced metaheuristic algorithms: Hunger Games Search (HGS), Slime Mould Algorithm (SMA), Chaos Game Optimizer (CGO), and Grey Wolf Optimizer (GWO) to forecast production volume and carbon emissions. When evaluated on a representative industrial case study, the SVR–CGO model demonstrated superior predictive accuracy, achieving R^2 values of 0.980 and 0.989 for production and emission targets, respectively. In the optimization phase, the best-performing SVR model serves as the fitness function for MOO, with the NSGA-II and SPEA2 algorithms addressing the inherent trade-offs between productivity and carbon reduction. Finally, a TOPSIS-based multi-criteria decision-making approach is applied to evaluate distinct operational strategies, including high-production, energy-saving, and green-energy modes. Results reveal that the proposed expert system can effectively identify Pareto-optimal configurations, enabling up to a 17% reduction in CO_2 emissions while sustaining over 95% of maximum production capacity. The primary novelty of this research lies in unifying carbon footprint modeling, metaheuristic-driven prediction, and evolutionary optimization into a cohesive decision-support tool that provides manufacturing stakeholders with actionable insights for sustainable operations.

Keywords. *Sustainable Manufacturing, Carbon Reduction, Support Vector Regression, Metaheuristic Optimization, Multi-Objective Optimization, NSGA-II.*

1. Introduction

In the past few years, the manufacturing sector has undergone significant change through the adoption of advanced digital technologies, enabling smart manufacturing systems [1]. These smart manufacturing systems use automation, real-time data, and interrelated cyber-physical systems to improve productivity, flexibility, and responsiveness. Likewise, increased environmental awareness and regulations have led industries to adopt more sustainable practices. Thus, SMEs now face competing demands to maintain production efficiencies while pursuing environmentally sustainable manufacturing practices, and this remains one of the most pressing challenges in modern manufacturing [2]. Despite significant progress in smart manufacturing, numerous methodologies continue to prioritize productivity improvements while overlooking long-term environmental implications [3]. More importantly, existing research frequently fails to provide comprehensive frameworks that integrate sustainability metrics and production objectives in a dynamic, multi-scenario decision-making context. This discrepancy underscores the urgent need for integrated models to help manufacturers reconcile the inherent trade-offs between environmental responsibility and productivity. It is imperative to confront this dual challenge to achieve global climate and energy objectives and advance sustainable manufacturing [4].

To achieve this vision, decision-makers must be able to effectively evaluate and reconcile competing objectives. The integration of life cycle modeling with advanced multi-objective optimization (MOO) offers a powerful approach that enables manufacturers to understand the simultaneous effects of alternative

*Corresponding Author -E-mail: zhenguangshao@outlook.com

 © SME

Journal of Manufacturing Engineering, September 2026, Vol. 21, Issue. 3, pp 077-093
DOI. https.//doi.org/10.37255/jme.v21i3pp077-093

production strategies on operational efficiency and environmental performance. These methods not only promote more responsible manufacturing practices but also assure compliance in markets that are becoming increasingly sustainability-driven and strengthen long-term competitiveness [3].

Recent research highlights the growing role of machine learning (ML) in enhancing the LCA framework [5]. ML techniques have been used to predict missing life cycle inventory (LCI) data [6], estimate the environmental impacts of chemicals, and assist in the interpretation phase of LCAs [7]. While several review studies have examined ML–LCA linkages, their coverage has often narrowed, focusing on specific domains such as the building sector [8] Click or tap here to enter text.or on broader scales such as urban systems [9].

In another relevant study, Tang et al. (2024) [10] developed a multi-objective optimization model for mechanical processing that integrates an improved particle swarm optimization (PSO) algorithm with a carbon emission quantification framework. Their approach effectively reduces emissions while improving machining efficiency, providing a practical reference for sustainable manufacturing decision-making. Chen et al. (2024) [11] employed a hybrid PSO–SVR model to forecast carbon emissions in China's logistics sector. The combination of the predictive power of support vector regression and the population-based optimization of PSO resulted in highly accurate emission forecasts and demonstrated the capabilities of ML–metaheuristic models for emission forecasting in industrialized sectors. Harsányi et al. (2024) [12] evaluated the predictive capability of two machine learning algorithms, gradient boosting regression (GBR) and support vector regression (SVR), and two deep learning algorithms, feedforward neural networks (FNN) and convolutional neural networks (CNN), for estimating CO_2 emissions from maize fields in Hungary and Iran. Across three input-variable scenarios, results revealed that GBR achieved the best training performance ($R^2 = 0.8778$), while FNN showed superior testing accuracy ($R^2 = 0.918$). Overall, the models ranked in performance as FNN > GBR > CNN > SVR, providing actionable insights for climate-smart agricultural practices and emission mitigation strategies. Shen et al. (2022) [13] proposed a multi-objective optimization framework based on life-cycle assessment and data-driven modeling for the steam-cracking process. In their study, they developed economic-environmental optimization models and solved them with NSGA-II to demonstrate how to find Pareto-optimal solutions. Results showed that optimization yields a better representation of environmental impacts than traditional energy-cost models, achieving a better trade-off between economic performance and sustainable ethylene production. Li et al. (2023) [14] investigated life cycle assessment and multi-objective optimization of industrial utility systems. Their framework balances operational costs and environmental impacts, showing how MOO can be effectively combined with optimization techniques to support sustainable system design.

1.1. Research Gaps & Novelties

Despite the notable advancements in smart manufacturing and applied machine learning, major research gaps persist in the literature. First, while machine learning and metaheuristic models have been widely explored for isolated emission predictions, their integration into dynamic, multi-scenario expert systems for smart manufacturing remains scarce. Second, carbon footprint modeling and operational optimization are frequently treated as disjointed stages, creating a significant disconnect between data-driven forecasting (predictive analytics) and strategic decision-making (prescriptive analytics). Third, existing industrial frameworks rarely address the explicit, simultaneous trade-off between maximizing production volume and minimizing carbon emissions, using a unified, intelligent pipeline that culminates in robust multi-criteria scenario evaluation.

To address these limitations, this study proposes an integrated intelligent decision-making framework that acts as a comprehensive expert system for sustainable manufacturing. The primary novelty of this research lies in pioneering the seamless coupling of SVR–metaheuristic predictive modeling with carbon-aware multi-objective optimization (MOO). Unlike prior works that isolate forecasting from operational planning, this research bridges predictive and prescriptive analytics, enabling manufacturing stakeholders to proactively forecast outcomes and optimize operational strategies under strict sustainability constraints. Furthermore, the study provides a rigorous comparative evaluation of four advanced metaheuristic optimizers (HGS, SMA, CGO, and GWO) for SVR hyperparameter tuning, as well as two state-of-the-art evolutionary MOO algorithms (NSGA-II and SPEA2). This comprehensive benchmarking offers deep insights into algorithmic efficiency, robustness, and convergence in complex industrial applications. Finally, by incorporating the TOPSIS multi-criteria decision-making method, the proposed framework translates complex Pareto-optimal trade-offs into practical, actionable industrial scenarios such as energy-saving, high-production, and green-energy modes. Through these cohesive contributions, the paper advances existing knowledge by providing an AI-driven toolkit that directly supports managerial decision-making to align high industrial productivity with environmental responsibility.

Journal of Manufacturing Engineering, September 2026, Vol. 21, Issue. 3, pp 077-093
DOI. https.//doi.org/10.37255/jme.v21i3pp077-093

2. Methodology

The proposed framework is designed in two main phases: (i) prediction of carbon emissions and production volume using Support Vector Regression (SVR) optimized with metaheuristic algorithms, and (ii) multi-objective optimization (MOO) to simultaneously minimize carbon emissions and maximize production volume using evolutionary algorithms. Figure 1 illustrates the overall workflow of the methodology.

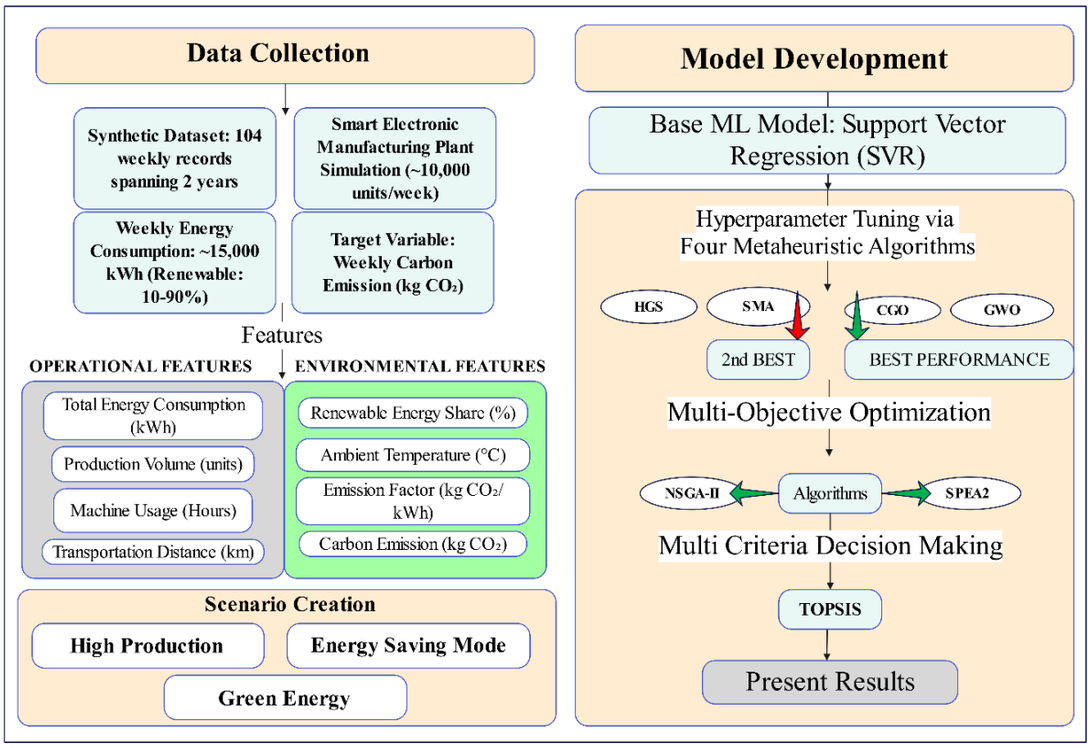


Fig. 1. Flowchart Showing the Workflow of the Methodology

2.1. Data Description

In this study, a representative case study was conducted focusing on the operational and environmental conditions of a mid-sized smart electronic components manufacturing plant. The dataset consisted of 2 years of weekly records, totaling 104 data points. Data spanning 104 weeks (two years) were used to capture seasonal differences and operational variability embedded in manufacturing plants. The factory produced about 10,000 electronic components per week, relying on automated and semi-automated production lines, and used about 15,000 kWh per week, with a share of renewable energy ranging from 10% to 90%, depending on changes in energy sourcing policies.

Key features in the dataset (Table 1) include total energy consumption (kWh), renewable energy share (%), production volume (number of units produced), machine usage hours, transportation distance (km), ambient temperature (°C), and an emission factor representing the kg CO_2 emitted per kWh consumed. The first target variable for prediction is weekly carbon emissions (kg CO2), calculated as total energy consumption multiplied by the emission factor.

This dataset serves as a comprehensive empirical basis for developing and validating predictive models and multi-objective optimization methods to minimize carbon emissions while maintaining production efficiency.

The contour plots in Figure 2 illustrate how different features affect both carbon emissions and production volume in the context of smart manufacturing. The first set shows that longer transportation distances and higher ambient temperatures significantly increase carbon emissions, while renewable energy share and total energy consumption influence production efficiency. The second set highlights the combined effect of transportation distance and emission factors: higher emission factors amplify carbon output, while production volume responds in ways that depend on operational trade-offs. Together, these plots emphasize the complex interactions between environmental and production objectives, justifying the need for multi-objective optimization.

 © SME

Journal of Manufacturing Engineering, September 2026, Vol. 21, Issue. 3, pp 077-093
DOI. https://doi.org/10.37255/jme.v21i3pp077-093

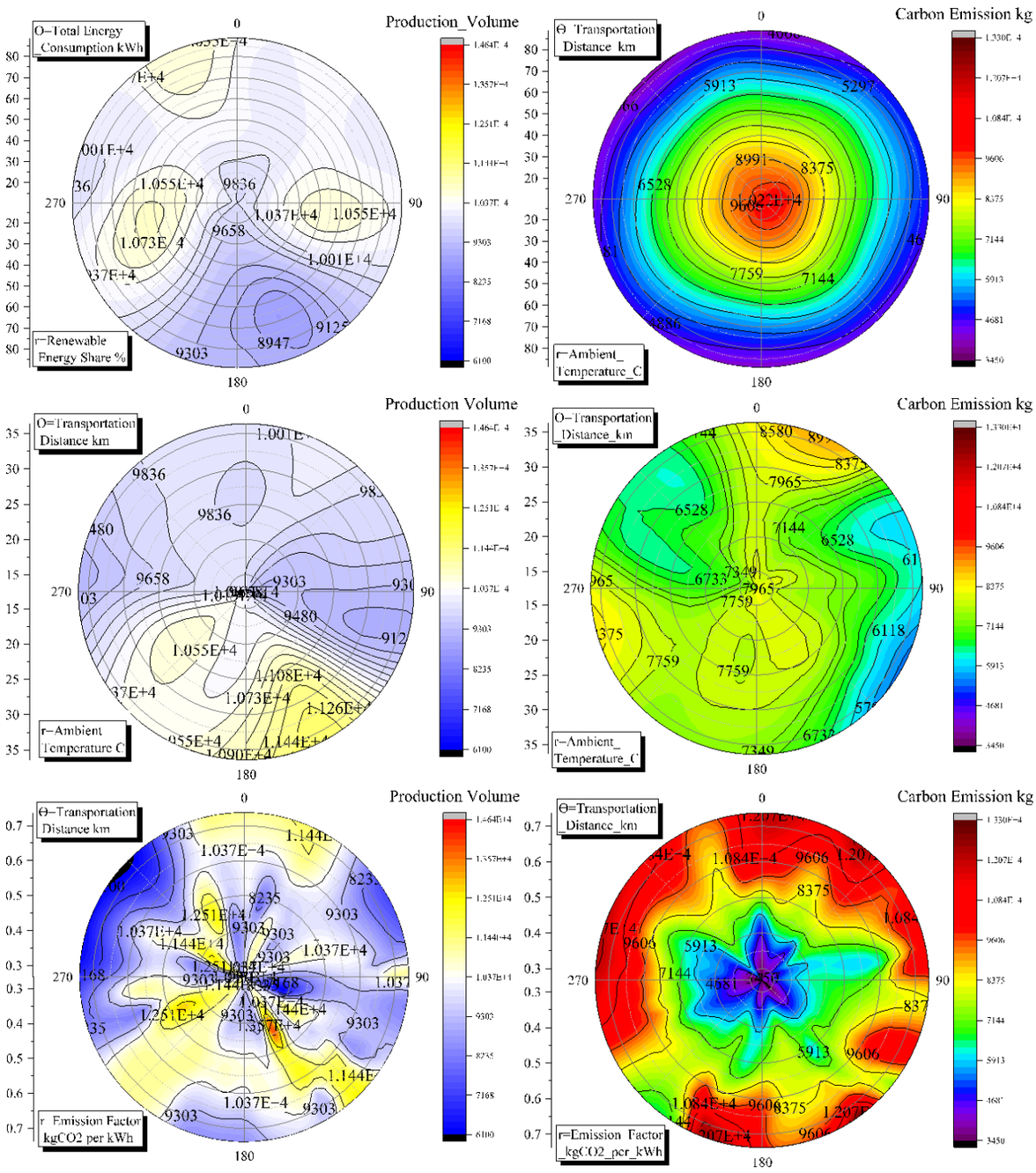


Fig 2. Effects of operational parameters on carbon emissions and production volume in smart manufacturing

2.2. Prediction Framework

SVR is employed as the base predictive model to estimate two key outputs: carbon emissions and production volume. Support Vector Regression (SVR) is a supervised machine learning method based on Support Vector Machines (SVMs), designed specifically for regression problems. SVM, and by extension SVR, do not attempt to minimize the prediction error directly; instead, SVR attempts to fit a function such that it is within ε of the target value. Hence, support vector regression is a form of regression modeling that is highly robust to overfitting and

 © SME

can accurately model complex, nonlinear relationships, making it ideally suited to modeling the real-world processes we study, such as production efficiency and carbon emissions in manufacturing.

Table 1. Key Features Provided in This Study

Feature	Unit	Description
Total Energy Consumption	kWh	Total weekly energy consumption
Renewable Energy Share	%	Percentage of energy from renewable sources
Production Volume	Number of units	Number of electronic components produced weekly
Machine Usage Hours	Hours	Total machine operating hours per week
Transportation Distance	Kilometers	Weekly transportation distance for products
Ambient Temperature	°C	Average ambient temperature in the factory
Emission Factor	kg CO_2 per kWh	Emission factor for energy consumption
Carbon Emission	kg CO_2	Weekly carbon emission (target variable)

The core idea of SVR is to approximate the target function $f(x)$ by mapping the input features $x \in \mathbb{R}$ into a high-dimensional feature space using a nonlinear mapping function $\phi(x)$. In this space, a linear regression function is constructed as:

$$f(x) = w^T \phi(x) + b \qquad (1)$$

Where, w is the weight vector, $\phi(x)$ is the feature transformation, and b is the bias term.

SVR introduces the concept of a ε-insensitive loss function, where errors smaller than ε are ignored. The optimization problem aims to find a flat function while allowing certain deviations beyond ε, penalized by slack variables ξ_i and ξ_i^*. The primal optimization problem can be formulated as:

$$\min \frac{1}{2}\|w\|^2 + C\sum_{i=1}^{n}(\xi_i + \xi_i^*) \qquad (2)$$

Subject to:

$$\begin{cases} y_i - \langle w, \phi(x_i)\rangle - b \le \epsilon + \xi_i \\ \langle w, \phi(x_i)\rangle + b - y_i \le \epsilon + \xi_i^* \\ \xi_i, \xi_i^* \ge 0 \end{cases} \qquad (3)$$

Here, $C > 0$ is a regularization parameter that controls the trade-off between model complexity and tolerance to deviations larger than ε.

By applying the Lagrangian method and kernel functions $K(x_i, x_j) = \phi(x_i)^T \phi(x_j)$, the dual form of SVR is derived. The prediction function can then be expressed as:

$$f(x) = \sum_{i=1}^{n}(a_i - a_i^*)\, K(x_i, x) + b \qquad (4)$$

Where a_i, a_i^* are Lagrange multipliers, and $K(x_i, x)$ is the kernel function (e.g., linear, polynomial, radial basis function). The choice of kernel function enables SVR to capture nonlinear dependencies effectively [15], [16].

In this study, SVR is used as the baseline predictive model for estimating carbon emissions and production volume. By combining it with metaheuristic optimization algorithms (HGS, SMA, CGO, and GWO), the SVR parameters (C, ε, and kernel parameters) are fine-tuned to improve prediction accuracy and robustness.

2.3. Metaheuristic Optimization

The performance of SVR strongly depends on its hyperparameters. To optimize these parameters, four metaheuristic algorithms are applied:

Hunger Game Search Optimizer (HGS)

The Hunger Games Search (HGS) is a recently developed metaheuristic optimization algorithm inspired by the survival strategies of animals competing for food under limited resources. The algorithm simulates the natural principle that individuals in a population adapt their behavior to balance exploration (searching for new areas of the solution space) and exploitation (refining known good solutions) based on their "hunger level." This adaptive mechanism makes HGS well-suited to solving complex, nonlinear, and multidimensional optimization problems, such as parameter tuning in machine learning models [17].

In the context of Support Vector Regression (SVR), the main challenge lies in selecting the optimal set of hyperparameters (C, ε, and kernel parameters such as γ).

 © SME

HGS addresses this by framing hyperparameter tuning as an optimization task, with the objective of minimizing prediction error (e.g., RMSE, MAE) while maintaining generalization.

The math model of HGS is based on a hunger-driven fitness update mechanism. Each candidate solution (i.e. each agent) in the population is associated with a hunger level H_i which evolves in relation to the agent's fitness and interactions with others. Hunger level directs the agent's movement through the search space while facilitating a balance of the global/local search components. The change in position of each agent is derived as:

$$X_i^{t+1} = X_i^t + r \cdot H_i \cdot \left(X_j^t - X_k^t\right) \quad (5)$$

Where X_i^t is the position (candidate solution) of agent i at iteration t, X_j^t and X_k^t are positions of two randomly selected agents, r is a random coefficient in the range of [0, 1], H_i is the hunger level of agent i, updated dynamically based on its fitness.

The hunger update function makes poorly fit agents "hungrier," encouraging them to explore new areas and leading to a more positive increase in agent fitness, while well-fit agents return to the center of their area, stabilizing exploitation.

The optimization process in HGS generally follows four steps [17]:

i. Initialization – Create an original population of candidate solutions (SVR hyperparameter sets).
ii. Hunger Evaluation - Hunger levels are based on candidates' fitness relative to the population.
iii. Position Update: Move candidate solutions in the search space according to the hunger-driven update equation.
iv. Termination - Repeat until either convergence is achieved or a maximum number of iterations is reached.

In this study, HGS is applied to optimize SVR hyperparameters for predicting carbon emissions and production volume. Its dynamic hunger mechanism enhances the ability to avoid premature convergence and ensures a better balance between accuracy and robustness compared with traditional optimization methods.

Slime Mould Optimizer (SMA)

The Slime Mould Algorithm (SMA) is a bio-inspired metaheuristic optimization algorithm that imitates the oscillatory behavior and foraging behavior of slime mould organisms in nature. Slime molds are single-celled organisms that are able to form fruiting networks in space to locate and exploit food sources. They can dynamically extend and contract their pseudopodia, which allows them to explore (search for new food sources) or exploit (improve paths to existing food sources), making them a meaningful metaphor for addressing global optimization problems [18].

The SMA search mechanism is mathematically modeled using adaptive weights and position update equations. The fitness of each candidate solution is evaluated against the objective function, such as minimizing RMSE or maximizing prediction accuracy. The position update of each slime mould agent is described as:

$$X_i^{t+1} = X_{best}^t + vb \cdot \left(W \cdot X_j^t - X_k^t\right) \quad (6)$$

Where X_i^{t+1} is the updated position of the agent i at iteration $t+1$, X_{best}^t is the current best solution, X_j^t and X_k^t are positions of randomly chosen agents, vb is a random value in the range of $[-a, a]$ controlling exploration and exploitation, W is an adaptive weight based on the fitness ranking of each solution.

The adaptive weights in SMA are crucial, as they simulate the contraction and expansion behavior of slime moulds when approaching a food source. Better fitness values correspond to stronger weights, which guide other agents toward promising regions of the search space [18].

The optimization process of SMA typically follows these steps:

i. Categorization - Create an initial candidate solutions population (SVR parameter options).
ii. Fitness evaluation - Evaluating all candidate solutions using prediction error methodology.
iii. Weighting - Ranking candidate solutions and applying adaptive weights to explore the search behavior.
iv. Position update - Changing agents' positions according to slime mould foraging properties.
v. Exit algorithm - Continuing the process until convergence or the maximum number of iterations has been reached.

By integrating SMA into the SVR training procedure, this research leveraged SMA's dynamic search adaptability and strong convergence properties. SMA helps balance global exploration with local exploitation, thereby avoiding premature convergence and improving the SVR model's generalization for predicting sustainability-related outcomes in manufacturing.

Journal of Manufacturing Engineering, September 2026, Vol. 21, Issue. 3, pp 077-093
DOI. https.//doi.org/10.37255/jme.v21i3pp077-093

Chaos Game Optimizer (CGO)

The Chaos Game Optimizer (CGO) is a new metaheuristic algorithm inspired by chaos game theory and fractal formation, such as the Sierpinski triangle. In the chaos game, a point moves, step by step, towards the vertices of a polygon based on some probabilistic rules, which ultimately looks like a fractal. CGO takes this idea of moving through the number of solutions associated to a moving point in space (the search space) using chaotic rules and geometric transformations to derive the next point, as part of the optimization process [19].

The algorithm combines exploration (diversifying the search in the solution space) and exploitation (intensifying the search around promising solutions) by adjusting movement patterns to move towards "attractive" points determined by previously found solutions (best-found solutions). This method of balancing exploration and exploitation allows CGO to find solutions to difficult lower-dimensional and nonlinear optimization problems.

The general mathematical formulation of CGO can be expressed as:

$$X_i^{t+1} = X_i^t + \beta \cdot (A - X_i^t) \quad (7)$$

Where A is a reference point selected based on chaotic rules (e.g., best solution, random vertex), and β is a scaling parameter controlling the step size, influenced by chaotic sequences.

The chaotic sequences introduce stochasticity and irregularity into the search process, preventing premature convergence and enabling CGO to explore the solution space more thoroughly than traditional optimizers [19], [20].

The CGO optimization process typically includes:

Initialization - Generate initial solutions (SVR hyperparameter sets).

Chaotic Mapping - Define movement rules using an inspiration based on the chaos game.

Update Position - Move towards the reference point based on chaotic dynamics, with an exploration and exploitation phase.

Fitness Evaluation - Evaluate each solution using error metrics (i.e. RMSE, MAE).

Convergence - Stop the algorithm when a stopping criterion is fulfilled (i.e., max number of iterations).

Incorporating CGO into SVR optimization, the work presented here benefits from CGO's fractal-driven chaotic search, which supports an expansive search of the parameter landscape and enables dynamic adjustment of exploration and exploitation. This makes CGO highly responsive for margin-optimizing predictive models in manufacturing sustainability contexts, where capturing the nonlinear relationships between ecological and production factors is critical.

Gray Wolf Optimizer (GWO)

The Gray Wolf Optimizer (GWO) is a population-based metaheuristic algorithm influenced by the leadership hierarchy and hunting behavior of gray wolves in their natural state. GWO was first proposed by Mirjalili et al. (2014) [21] and has been successfully applied due to its popularity, simplicity, fast convergence, and ability to balance global and local search.

In nature, gray wolves live and hunt in packs organized into a social hierarchy:

i. Alpha (α): The leader responsible for decision-making and guiding the pack (best solution).
ii. Beta (β): The subordinate wolves assisting the alpha in decision-making (second-best solutions).
iii. Delta (δ): Wolves that dominate the rest of the pack but follow alpha and beta (third-best solutions).
iv. Omega (ω): The lowest-ranking wolves, following the guidance of higher-ranked wolves (remaining solutions).

This hierarchy is mathematically translated into the optimization process by considering α, β, and δ as the best three candidate solutions, that guide the rest of the population (omega)during the search. The hunting process of gray wolves is modeled in three main steps: encircling prey, hunting, and attacking prey (exploitation).

Encircling Prey:

Gray wolves encircle prey by updating their positions relative to the best solution found:

$$\vec{D} = \left|\vec{C} \cdot \vec{X}_p(t) - \vec{X}(t)\right| \quad (8)$$

$$\vec{X}(t+1) = \vec{X}_p(t) - \vec{A} \cdot \vec{D} \quad (9)$$

 © SME

Where $\vec{X}_p(t)$ is the prey's position (best solution so far), $\vec{X}(t)$, is the current wolf's position, $\vec{A}$ and $\vec{D}$ are coefficient vectors controlling exploration and exploitation.

Hunting (Guidance by α, β, δ):

Since the exact prey location is unknown, GWO assumes that the top three wolves (α, β, δ) guide the search, with each wolf updating its position according to:

$$\vec{X}(t+1) = \frac{\vec{X}_1 + \vec{X}_2 + \vec{X}_3}{3} \quad (10)$$

Where $\vec{X}_1, \vec{X}_2, \vec{X}_3$ are position estimates based on $\alpha, \beta,$ and δ.

Attacking Prey (Exploitation):

As iterations proceed, the parameter $|\vec{A}|$ decreases linearly, shifting the algorithm from exploration (searching new regions) to exploitation (refining around the best solutions).

In this research, GWO is used to optimize the hyperparameters of Support Vector Regression (SVR) in predicting carbon emissions and production volume in smart manufacturing. GWO is a type of metaheuristic algorithm inspired by the wolves' cooperative hunting strategy. GWO designates wolves based on the best solutions found. This strategy allows for exploration of the solution space in a diverse manner while discovering promising solutions within a localized search. Overall, this approach provides greater accuracy and generalization for the SVR model than random or manual parametric tuning. GWO's ability to compute and avoid local optima without sacrificing convergence makes it useful in sustainability analyses, where finding the best trade-offs between environmental impacts and production performance is non-linear [21], [22].

2.4. Multi-Objective Optimization Algorithms

Two evolutionary algorithms are applied to solve the formulated problem:

Non-dominated Sorting Genetic Algorithm II (NSGA-II)

The Non-dominated Sorting Genetic Algorithm II (NSGA-II) is one of the most widely adopted multi-objective evolutionary algorithms, designed to effectively solve optimization problems with conflicting objectives. Introduced by Deb et al. (2002) [23], NSGA-II improves upon its predecessor (NSGA) by introducing mechanisms that balance convergence towards the Pareto-optimal front and diversity among solutions. The core idea is to simultaneously approximate a set of optimal trade-off solutions rather than a single point, thereby enabling decision-makers to evaluate multiple scenarios based on the relative importance of objectives.

The process starts with a randomly initialized population; selection, crossover, and mutation are then performed to generate offspring solutions. NSGA-II uses a fast-non-dominated sorting technique to classify solutions, using fronts to reflect different levels of domination in the Pareto-optimal front. The algorithm introduces the idea of crowding distance, a density measure that preserves diversity by favoring solutions in less-populated regions of the search space. This results in solutions that aren't only close to the Pareto-optimal front but also well distributed on it.

Mathematically, given a population P and offspring Q, NSGA-II combines them into a set $R = P \cup Q$. The set R is sorted into several non-dominated fronts $F_1, F_2, \ldots$. Starting from F_1, solutions are filled into the next generation until the population limit is reached. If the last front does not completely fit, individuals are selected bin descending order of crowding distance The overall fitness function is evaluated with respect to multiple objectives $f_i(x),\ i = 1, 2, \ldots, m,$ and the algorithm minimizes or maximizes them subject to the Pareto dominance rule:

$$x_1 < x_2 \quad (11)$$

If and only if $\forall i: f_i(x_1) \leq f_i(x_2) \wedge \exists j: f_j(x_1) < f_j(x_2)$

Where x_1 dominates x_2 if it is no worse in all objectives and strictly better in at least one objective.

In the context of this study, NSGA-II is utilized to optimize multiple competing targets, such as improving predictive accuracy while minimizing carbon emission and enhancing production volume. Its ability to discover a diverse set of Pareto-optimal solutions makes it particularly well-suited for engineering and energy-related applications, where trade-offs between performance, efficiency, and sustainability are often unavoidable [23], [24].

Strength Pareto Evolutionary Algorithm 2 (SPEA2)

The Strength Pareto Evolutionary Algorithm 2 (SPEA2) is a widely used multi-objective optimization algorithm that efficiently approximates the Pareto-optimal front (the set of nondominated solutions) while preserving solution diversity. SPEA2 is an extension of the original

 © SME

Journal of Manufacturing Engineering, September 2026, Vol. 21, Issue. 3, pp 077-093
DOI. https.//doi.org/10.37255/jme.v21i3pp077-093

SPEA conceptual framework and includes improvements in fitness assignment, diversity preservation, and environmental selection that were not performed by previous evolutionary approaches. SPEA2 is widely recognized for robustness and for producing near-optimal, well-distributed Pareto fronts for many engineering and decision problems.

The underlying principle of SPEA2 is the strength-based fitness assignment mechanism. To every solution, a strength is assigned, which depends on how many individuals in the population are dominated by that solution. The fitness of a solution is the sum of the strengths of every solution that dominates it, combining dominance and density information during selection. This is mathematically expressed as:

$$F(i) = \sum_{j \in P \cup A,\ j \neq i \wedge j < i} S(j) \qquad (12)$$

Where $F(i)$ is the fitness of individual i, $S(j)$ is the strength of individual j, P is the current population, A is the external archive, and $j < i$ denotes that solution j dominates i. SPEA2 ensures diversity using a k-th nearest neighbor density estimation based on a distance measure relative to the k-th nearest neighbor in objective space. This promotes uniform spreading of solutions on the Pareto front and reduces the risk for premature convergence in certain areas. Another distinctive feature of SPEA2 is the external archive that stores non-dominated solutions across generations. If the archive exceeds its predefined size, a distance-based truncation operator selectively removes solutions to preserve diversity while retaining representative Pareto-optimal solutions. This archive plays a critical role in guiding the evolutionary process towards convergence and maintaining elite solutions.

SPEA2 is a great tool for balancing trade-offs among competing objectives, such as predictive accuracy, energy efficiency, and computational cost. SPEA2 is advantageous because it combines a dominance-based fitness with a density estimate to obtain a diverse and converged set of Pareto-optimal solutions. Therefore, SPEA2 is especially useful as a multi-objective problem-solving algorithm for complex problems in engineering and sustainability studies, where decision-makers need a well-distributed set of optimal alternatives rather than a single solution [25], [26].

2.5. Selecting the Best Solution

To select the Pareto-optimal solutions in the MOO process, the TOPSIS method is implemented. The TOPSIS method [27] is a multi-criteria decision-making (MCDM) approach used to rank and select alternatives based on their relative closeness to the ideal solution. Unlike optimization algorithms, TOPSIS does not search for Pareto fronts; rather, it evaluates pre-obtained solutions (e.g., from MOO or model results) using a structured ranking process.

Let the decision matrix be $X = [x_{ih}]$, where $i = 1, 2, \dots, m$ represents alternatives and $j = 1, 2, \dots, n$ represents criteria. The TOPSIS method proceeds as follows:

Normalization

Each criterion value is normalized to make it dimensionless:

$$r_{ij} = \frac{x_{ij}}{\sqrt{\sum_{i=1}^{m} x_{ij}^2}} \qquad (13)$$

Weighted Normalized Matrix

The normalized matrix is multiplied by the weight vector w_j (importance of each criterion):

$$v_{ij} = w_j \times r_{ij} \qquad (14)$$

Determine Ideal and Negative-Ideal Solutions

The positive ideal solution (best values) and the negative ideal solution (worst values) are defined as:

$$A^+ = \{v_1^+, v_2^+, \dots, v_n^+\}, \quad v_j^+ = \max_i \left(v_{ij}\right)$$
$$A^- = \{v_1^-, v_2^-, \dots, v_n^-\}, \quad v_j^- = \min_i \left(v_{ij}\right) \qquad (15)$$

Calculate Separation Measures

The Euclidean distance of each alternative from both ideal and negative-ideal solutions is computed as:

$$S_i^+ = \sqrt{\sum_{j=1}^{n} \left(v_{ij} - v_j^+\right)^2}, \qquad (16)$$

$$S_i^- = \sqrt{\sum_{j=1}^{n} \left(v_{ij} - v_j^-\right)^2}$$

Compute Relative Closeness to Ideal Solution

$$C_i = \frac{S_i^-}{S_i^+ + S_i^-} \qquad (17)$$

Where $0 \leq C_i \leq 1$. A higher C_i value indicates a solution closer to the ideal.

 © SME

Journal of Manufacturing Engineering, September 2026, Vol. 21, Issue. 3, pp 077-093
DOI. https.//doi.org/10.37255/jme.v21i3pp077-093

Ranking

Alternatives are ranked in descending order of C_i.

2.6. Simulation of Operational Scenarios and Intelligent Decision-Making

To evaluate the applicability and robustness of the proposed framework, multiple operational scenarios were simulated based on the established case study parameters. These scenarios represented diverse configurations of production levels, renewable energy shares, and machine usage schedules, reflecting possible real-world conditions.

The simulations enabled assessment of how strategic operational changes, such as increasing renewable energy integration or optimizing machine operating hours, affect carbon emissions and production efficiency. Specifically, three representative scenarios were considered: Base Case, representing the reference operational configuration with balanced emphasis on carbon emission and production objectives; High Production, characterized by increased output accompanied by a reduced share of renewable energy, potentially due to resource constraints; Green Energy Focus, emphasizing a high share of renewable energy with lower production to minimize carbon emissions; and Energy Saving Mode, involving reduced energy consumption and machine operating hours to enhance energy conservation.

To determine the most suitable operational configuration under multiple, and often conflicting, objectives, a Multi-Criteria Decision-Making (MCDM) approach was applied using the TOPSIS method. As shown in Table 2, each operational scenario was evaluated based on two main criteria: carbon emissions and production volume, with specific weightings reflecting different strategic priorities. These weights were used as input parameters for the TOPSIS analysis to identify the best compromise (Pareto-optimal) solutions among the simulated scenarios. The weighting coefficients for the objectives were determined based on the strategic priorities defined for each operational scenario. Specifically, the weights were assigned to reflect different managerial focuses, such as emission minimization or production maximization, under realistic industrial conditions. This approach enables the TOPSIS method to evaluate trade-offs between conflicting objectives and identify the most balanced (Pareto-optimal) configurations.

2.7. Evaluation and Validation

The predictive models were validated using standard performance metrics, including the coefficient of determination (R^2), root mean square error (RMSE), mean absolute error (MAE), mean squared error (MSE), relative absolute error (RAE), variance accounted for (VAF) and A10 index. The effectiveness of the optimization approach was assessed by analyzing the distribution of solutions along the Pareto front and comparing performance across scenarios.

3. Results

This section provides results of the first step, containing performance of optimized models in predicting target variables, along with the second step encompassing MOO results and Pareto points. In the first step, Figure 3 shows the fitness plots, which indicate the prediction accuracy of four optimized models. The fitness plots for predicting production volume indicate better performance of the CGO-optimized model, with predicted samples more closely matching the target samples. The CGO-SVR model also outperforms the other models in predicting carbon emissions. The fitness plot clearly denotes the remarkable accuracy of the CGO-optimized model compared to the other hybrid models.

Table 2. Assigned weights of carbon emission and production objectives across operational scenarios

	Base Case	Green Energy Focus	High Production	Energy Saving Mode
Carbon Emission	0.5	0.92869508	0.243101508	0.460978887
Production Volume	0.5	0.07130492	0.756898492	0.539021113

 © SME

Journal of Manufacturing Engineering, September 2026, Vol. 21, Issue. 3, pp 077-093
DOI. https://doi.org/10.37255/jme.v21i3pp077-093

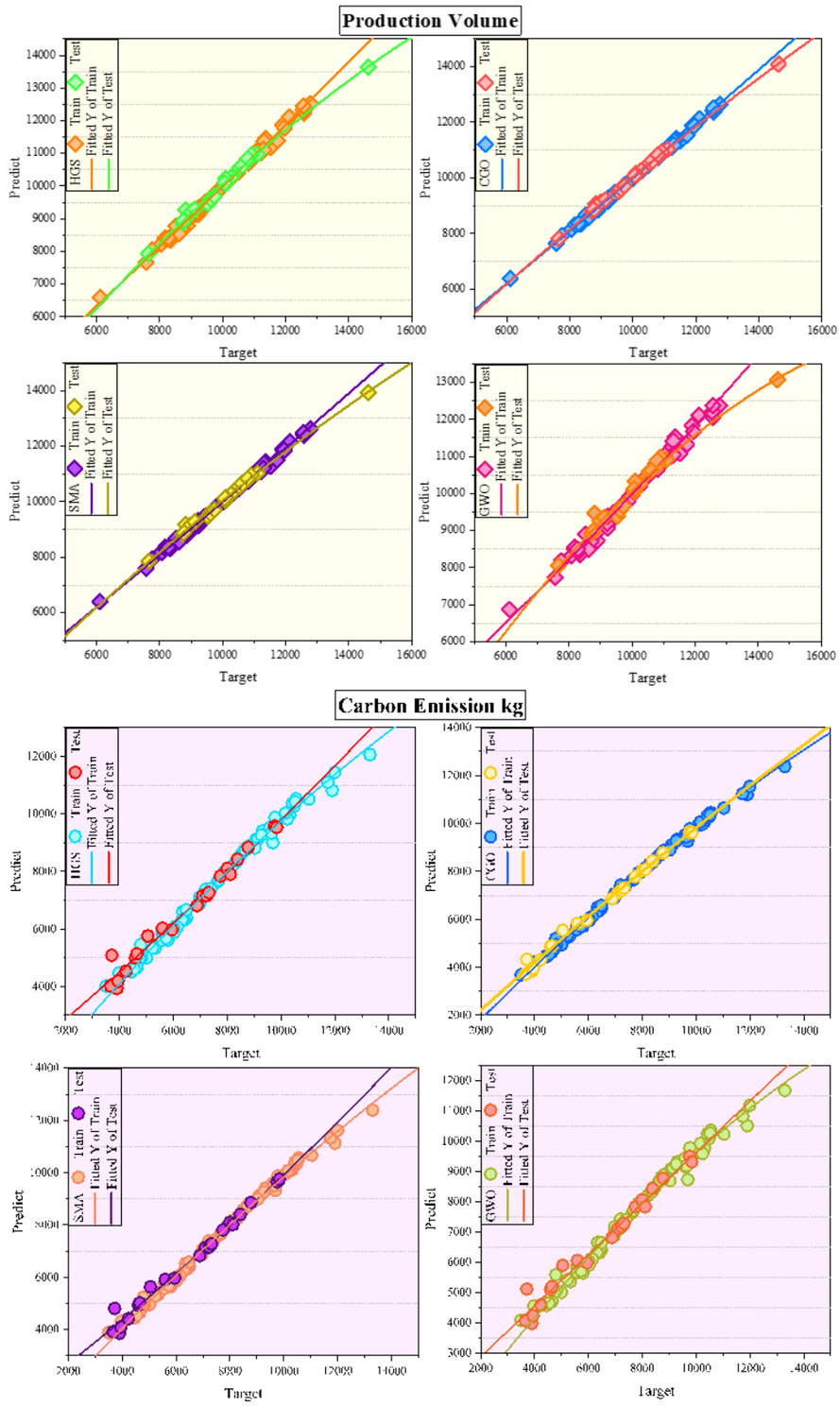


Fig. 3. Fitness Plots Comparing the Prediction Accuracy of Models

Journal of Manufacturing Engineering, September 2026, Vol. 21, Issue. 3, pp 077-093
DOI. https://doi.org/10.37255/jme.v21i3pp077-093

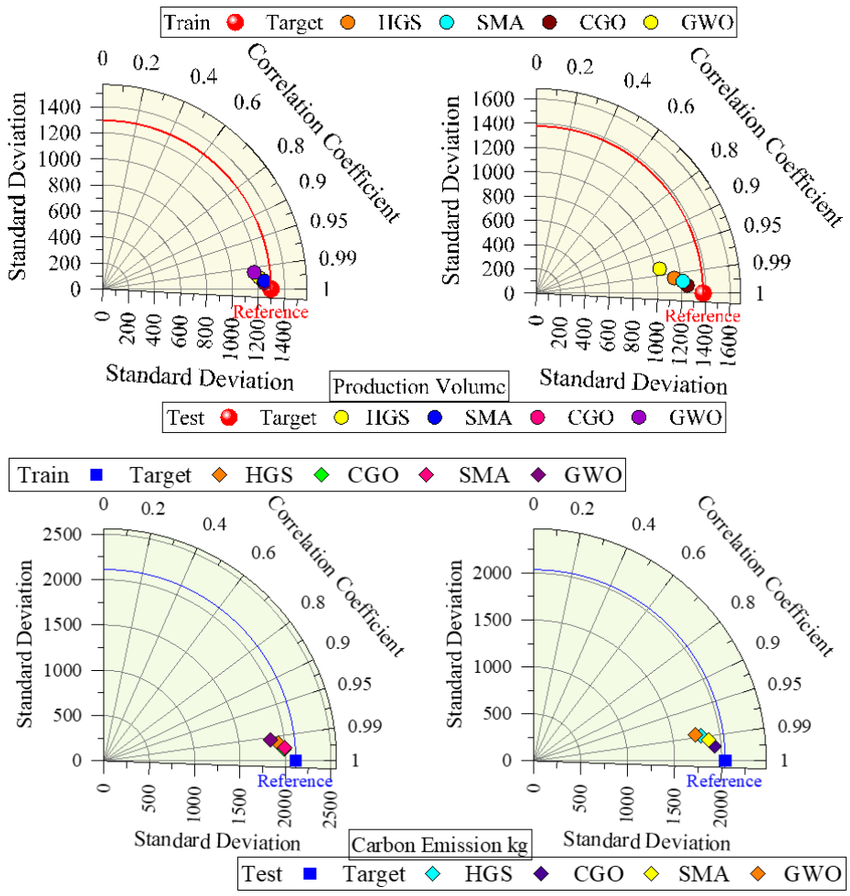


Fig. 4. Taylor Diagrams Comparing the Predictive Performance of Models

Fig. 4 presents Taylor diagrams that analyze the predictive capability of SVR models optimized using HGS, CGO, SMA, and GWO. In production volume prediction, the CGO-optimized model and then the SMA-optimized model demonstrate more robust performance, achieving a higher correlation coefficient and a closer fit to the reference point. The SVR-CGO also achieves the lowest standard deviation. The carbon emissions prediction also confirms the CGO-optimized model's superior performance, bringing it closer to the target. The SVR-SMA is the second-best model with high correlation coefficient. Fig. 5 presents the prediction performance analysis of the optimized models using statistical evaluation metrics. For carbon emissions prediction, the SVR-CGO achieves the highest R2 and the lowest RMSE, MAE, and MAPE.

The prediction of production volume also denotes higher R^2 and lower RMSE, MAE, and MAPE for the CGO-optimized model. This indicates comparatively significant performance of SVR-CGO in the prediction tasks. Table 3, provides the obtained metric values by models in predicting the targets. For the prediction of production volume, the integration of SVR and CGO achieved an R2 of 0.98 and an RMSE of 247.65; the SVR-SMA also shows promising performance with an R2 of 0.95 and an RMSE of 407.49, being the second-best model.

For carbon emissions prediction, the SVR-CGO achieved an R2 of 0.989 and lower error metrics, including an RMSE of 206.7 and an MAE of 126.06, indicating that this model outperforms the other optimized SVRs.Furthermore, the model runtimes are compared in Figure 6. For carbon emissions, the CGO algorithm is the least efficient, requiring the longest runtime (104.89), whereas the SMA algorithm is the most efficient, with the shortest runtime (28.19), closely followed by GWO (29.74). Conversely, when optimizing for Production Volume, the CGO algorithm remains the least efficient (106.00), but the relative efficiencies of the other algorithms change significantly. For this objective, HGS proves to be the most efficient with the lowest runtime (26.09).

Journal of Manufacturing Engineering, September 2026, Vol. 21, Issue. 3, pp 077-093
DOI. https://doi.org/10.37255/jme.v21i3pp077-093

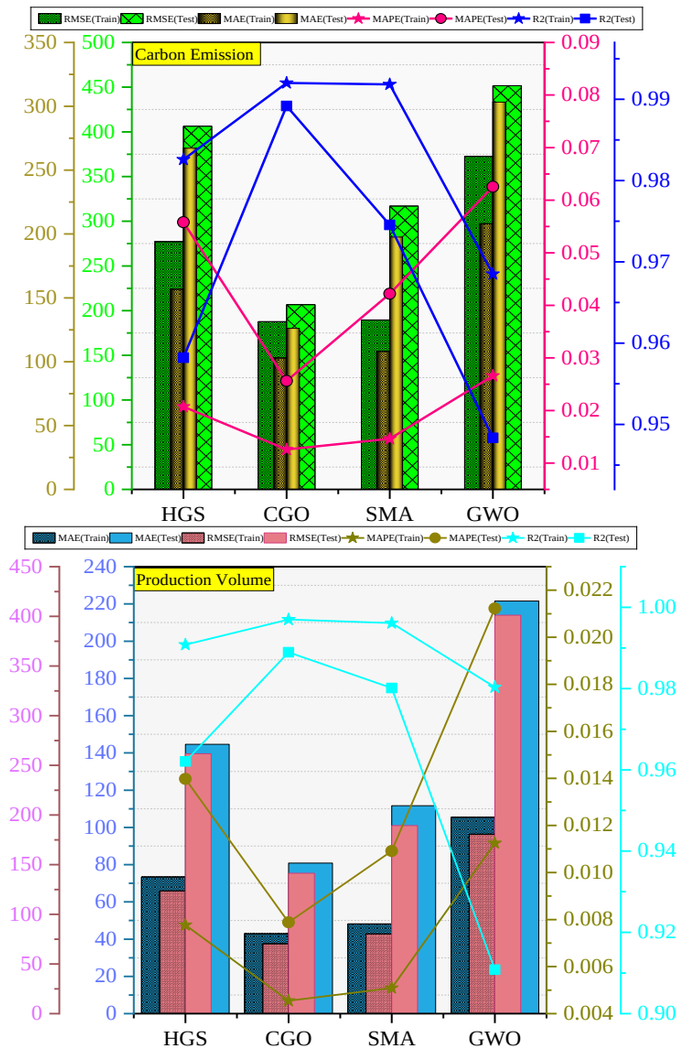


Fig. 5. Comparison of the Optimized Models in Predicting Carbon Emission and Production Volume according to the Evaluation Metrics

Table 3. Predictive Performance Evaluation of Optimized SVR Models across Various Metrics

	Production Volume							
	HGS-SVR		CGO-SVR		SMA-SVR		GWO-SVR	
	Train	Test	Train	Test	Train	Test	Train	Test
MAE	211.8432	385.2298	110.2084	154.7903	175.0006	269.0934	190.851	544.698
MSE	139377.7	322544.7	39691.3	61331.25	99355.93	166050.6	165411.3	678383.3
RMSE	373.3333	567.9302	199.2268	247.6515	315.2078	407.4931	406.7079	823.6403
R^2	0.968384	0.918317	0.990997	0.984468	0.977463	0.957949	0.962479	0.828203
VAF	96.92661	93.47339	99.13868	98.7182	97.84855	96.53917	96.3208	87.61564
RAE	0.047	0.084753	0.025081	0.036957	0.039682	0.060811	0.051201	0.122913
A10	0.927711	0.714286	1	0.952381	0.939759	0.904762	0.927711	0.666667

 © SME

Carbon Emission								
	HGS-SVR		CGO-SVR		SMA-SVR		GWO-SVR	
	Train	Test	Train	Test	Train	Test	Train	Test
MAE	156.6426	267.1333	102.8407	126.0633	108.0498	197.5957	207.9951	303.2746
MSE	76772.61	165067.5	35195	42727.82	35948.06	100484	138817	204022.9
RMSE	277.0787	406.285	187.6033	206.7071	189.5997	316.9921	372.5815	451.6889
R^2	0.982585	0.958198	0.992017	0.989179	0.991846	0.974553	0.968512	0.948332
VAF	98.30428	96.66141	99.24014	99.08639	99.19684	98.04183	97.006	95.69883
RAE	0.034882	0.060631	0.023618	0.030847	0.023869	0.047305	0.046905	0.067406
A10	0.951807	0.904762	1	0.952381	0.987952	0.904762	0.927711	0.809524

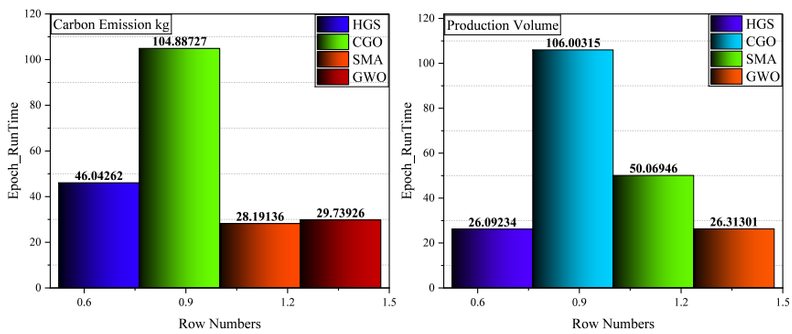


Fig. 6. Computational Efficiency Comparison of the Optimized Models

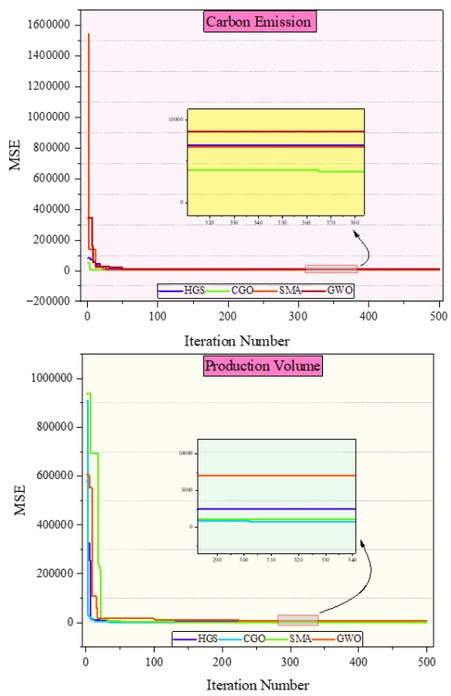


Fig 7. Convergence behavior of the Optimized Models for the production volume and Carbon Emission Predictions

Fig 7 illustrates the convergence performance of four optimized models. From the convergence trends for production volume, all algorithms demonstrate rapid improvement in the early iterations, achieving a significant reduction in the objective function within the first few cycles. Among them, HGS and CGO exhibit the fastest convergence rates, stabilizing quickly around optimal regions, indicating a strong balance between exploration and exploitation. SMA shows slightly slower convergence but maintains a consistent trajectory toward stability, demonstrating robustness and avoidance of premature stagnation. In contrast, GWO converges at a moderate rate but achieves a higher final objective value, suggesting lower accuracy in reaching the global optimum compared to the other algorithms.

The convergence rates in carbon emission prediction, the CGO and HGS achieve the lowest convergence rates at the earliest iterations, stabilizing quickly. Although the SMA-optimized model starts with a high convergence rate, it rapidly reaches optimal regions and remains stable during iterations.

Fig. 8 illustrates the Pareto-optimal solutions obtained from the multi-objective optimization process using the NSGA-II and SPEA2 algorithms. The optimization simultaneously maximized production volume and minimized carbon emissions (kg CO_2),

Journal of Manufacturing Engineering, September 2026, Vol. 21, Issue. 3, pp 077-093
DOI. https://doi.org/10.37255/jme.v21i3pp077-093

generating a well-defined Pareto front that represents the trade-offs between environmental and operational objectives.

In both optimization scenarios, each dot signifies a candidate solution within the population, while the darker edge points represent the Pareto front, denoting non-dominated optimal trade-offs. The NSGA-II (Figure 8a) yielded a more extensive and smoother Pareto front, illustrating its robust convergence and diversity-preserving properties. Conversely, the SPEA2 (Figure 8b) attained a more compact front, highlighting high-performing solutions near the ideal region with reduced dispersion.

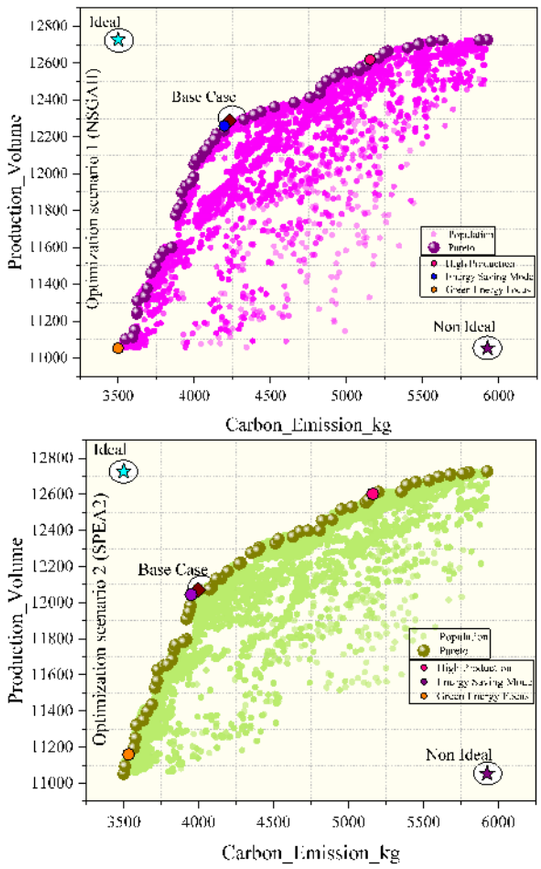


Fig 8. Pareto front analysis and multi-objective optimization results obtained using NSGA-II and SPEA2 algorithms

The figure also highlights key operational modes: High Production, Green Energy Focus, and Energy Saving Mode. The High Production point shows maximum output with relatively higher emissions, whereas Green Energy Focus minimizes emissions at the expense of production efficiency. The Energy Saving Mode represents a balanced configuration that achieves energy conservation at moderate production levels.

The comparison also reveals that NSGA-II provides a more extensive exploration of the solution space, while SPEA2 yields slightly more stable convergence toward optimal trade-offs, demonstrating its efficiency in multi-objective industrial sustainability optimization.

4. Discussion

The proposed two-phase intelligent decision-making framework establishes an integrated approach that unites predictive machine learning modeling and multi-objective optimization (MOO) to foster sustainable smart manufacturing. This framework addresses the persistent gap between production optimization and environmental sustainability by seamlessly coupling Support Vector Regression (SVR) optimized via advanced metaheuristic algorithms, with evolutionary MOO methods. Unlike earlier studies that often treat emission forecasting and operational planning as isolated stages, this research unifies them into a cohesive expert system, enabling simultaneous forecasting and data-driven decision-making.

The primary novelty of this research is the integration of carbon footprint modeling and machine learning to evaluate environmental impacts and industrial productivity concurrently. By explicitly modeling the trade-offs between carbon emission minimization and production volume maximization, the framework demonstrates how competing industrial objectives can be dynamically balanced.

From a predictive standpoint, evaluating multiple metaheuristic optimizers (HGS, SMA, CGO, and GWO) provided comprehensive insights into their algorithmic efficiency and robustness. The results confirm that the SVR-CGO model achieved the highest predictive accuracy, attaining R^2 values of 0.989 for carbon emissions and 0.980 for production volume, along with the lowest RMSE and MAE values. These outcomes confirm the strong convergence capability and adaptability of the Chaos Game Optimizer (CGO) in handling complex, nonlinear manufacturing data.

From an optimization perspective, integrating the SVR-CGO model as the fitness function for NSGA-II and SPEA2 successfully generated well-defined Pareto fronts. NSGA-II produced a more diverse and continuous front, indicating superior convergence diversity and broad exploration of trade-off solutions. In contrast, SPEA2 yielded a more compact front closer to the ideal region, confirming its stability and precision in localized searches.

Furthermore, the scenario-based simulations evaluated via the TOPSIS method (Base Case, High Production, Green Energy Focus, and Energy Saving Mode) provided critical managerial implications. The decision-support tool revealed that higher renewable energy shares effectively reduced carbon footprints but

 © SME

introduced production constraints. Conversely, optimized machine usage in the Energy Saving Mode enhanced productivity with minimal increases in energy demand. These insights empower factory managers to proactively schedule operations, achieving up to a 17% reduction in CO_2 emissions while maintaining over 95% of maximum production capacity.

Despite the comprehensive methodological basis of the proposed expert system, certain limitations provide avenues for future research. First, while the employed representative industrial case study effectively validates the framework's capabilities, future applications should encompass a broader range of stochastic variability present in highly volatile industrial sectors. Second, the temporal dynamics of production and energy availability were not explicitly modeled; integrating time-series forecasting and adaptive scheduling could further refine the system. Finally, broadening the objective space to encompass economic costs, water footprint, or waste generation would facilitate a more holistic multi-criteria sustainability assessment.

5. Conclusion

This study successfully developed an intelligent decision-support system that couples machine learning, metaheuristic algorithms, and evolutionary multi-objective optimization to advance sustainable practices in smart manufacturing. The proposed methodology integrates carbon footprint modeling directly into the operational optimization pipeline, providing industrial stakeholders with a reliable expert system to reconcile high productivity demands with strict environmental regulations.

Numerical findings demonstrated that the SVR-CGO model attained exceptional predictive accuracy (R^2 = 0.989 for carbon emissions and R^2 = 0.980 for production volume), significantly outperforming other hybrid variants (HGS, SMA, and GWO). During the multi-objective optimization phase, NSGA-II facilitated enhanced exploration by generating a broader Pareto front, while SPEA2 demonstrated concentrated convergence toward near-optimal solutions. Furthermore, applying the TOPSIS multi-criteria decision-making method across various operational scenarios demonstrated the framework's practical utility. Notably, the Energy Saving Mode achieved an equitable and actionable balance between production efficiency and emission reduction, proving the system's capability to guide sustainability-oriented strategic planning.

Ultimately, the results highlight the profound potential of coupling machine learning with evolutionary optimization to create actionable decision-making tools for modern industries. Subsequent investigations will focus on expanding this framework by incorporating deep reinforcement learning or transformer-based predictive models for real-time adaptive scheduling, and broadening the optimization parameters to include comprehensive socio-economic metrics for holistic, sustainable manufacturing.

Statements and Declarations

Availability of data and material

The data supporting the findings of this study are available from the corresponding author upon reasonable request.

Competing interests

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

Author Contributions

Zhenguang Shao: Data curation, Formal analysis, Methodology, Resources, Software, Visualization, Writing – original draft, Writing – review & editing, Supervision, Project administration.

Funding

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

References

[1]. M. Ghobakhloo, "The future of manufacturing industry: a strategic roadmap toward Industry 4.0," Journal of Manufacturing Technology Management, vol. 29, no. 6, pp. 910–936, Jul. 2018, doi: 10.1108/JMTM-02-2018-0057.

[2]. D. I. Handayani, I. Masudin, A. Rusdiansyah, and J. Suharsono, "Production-Distribution Model Considering Traceability and Carbon Emission: A Case Study of the Indonesian Canned Fish Food Industry," Logistics, vol. 5, no. 3, p. 59, Sep. 2021, doi: 10.3390/logistics5030059.

[3]. X. X. Romeiko et al., "A review of machine learning applications in life cycle assessment studies," Science of The Total Environment, vol. 912, p. 168969, Feb. 2024, doi: 10.1016/j.scitotenv.2023.168969.

[4]. B. Hasanain, "The Role of Ergonomic and Human Factors in Sustainable Manufacturing: A Review," Machines, vol. 12, no. 3, p. 159, Feb. 2024, doi: 10.3390/machines12030159.

[5]. M. Dick, M. Abreu da Silva, and H. Dewes, "Mitigation of environmental impacts of beef cattle production in southern Brazil – Evaluation using farm-based life cycle assessment," J Clean Prod, vol. 87, pp. 58–67, Jan. 2015, doi: 10.1016/j.jclepro.2014.10.087.

[6]. P. Hou, O. Jolliet, J. Zhu, and M. Xu, "Estimate ecotoxicity characterization factors for chemicals in life cycle assessment using machine learning models," Environ Int, vol. 135, p. 105393, Feb. 2020, doi: 10.1016/j.envint.2019.105393.

[7]. R. Azari, S. Garshasbi, P. Amini, H. Rashed-Ali, and Y. Mohammadi, "Multi-objective optimization of building envelope design for life cycle environmental performance,"

Energy Build, vol. 126, pp. 524–534, Aug. 2016, doi: 10.1016/j.enbuild.2016.05.054.

[8]. *A. N. N. Barros and R. C. Ruschel, "Machine Learning for Whole-Building Life Cycle Assessment: A Systematic Literature Review," 2021, pp. 109–122.* doi: 10.1007/978-3-030-51295-8_10.

[9]. *A. Ghoroghi, Y. Rezgui, I. Petri, and T. Beach, "Advances in application of machine learning to life cycle assessment: a literature review," Int J Life Cycle Assess*, vol. 27, no. 3, pp. 433–456, Mar. 2022, doi: 10.1007/s11367-022-02030-3.

[10]. *L. Tang, "Multi-objective optimization model for mechanical processing based on improved PSO algorithm and carbon emission quantification algorithm," Sustainable Energy Research*, vol. 11, no. 1, p. 37, Nov. 2024, doi: 10.1186/s40807-024-00133-z.

[11]. *L. Chen, Y. Pan, and D. Zhang, "Prediction of Carbon Emissions Level in China's Logistics Industry Based on the PSO-SVR Model," Mathematics*, vol. 12, no. 13, p. 1980, Jun. 2024, doi: 10.3390/math12131980.

[12]. *E. Harsányi et al., "Assessment of Advanced Machine and Deep Learning Approaches for Predicting CO2 Emissions from Agricultural Lands: Insights Across Diverse Agroclimatic Zones," Earth Systems and Environment*, vol. 8, no. 4, pp. 1109–1125, Dec. 2024, doi: 10.1007/s41748-024-00424-x.

[13]. *W. Shen, Z. Tian, L. Zhao, and F. Qian, "Life Cycle Assessment and Multiobjective Optimization for Steam Cracking Process in Ethylene Plant," ACS Omega*, vol. 7, no. 18, pp. 15507–15517, May 2022, doi: 10.1021/acsomega.2c00189.

[14]. *H. Li and L. Zhao, "Life cycle assessment and multi-objective optimization for industrial utility systems," Energy*, vol. 280, p. 128213, Oct. 2023, doi: 10.1016/j.energy.2023.128213.

[15]. *D. Basak, S. Pal, and D. C. Patranabis, "Support Vector Regression," 2007. [Online]. Available: https://api.semanticscholar.org/CorpusID:264683345*

[16]. *M. Awad and R. Khanna, "Support Vector Regression," in Efficient Learning Machines, Berkeley, CA: Apress, 2015*, pp. 67–80. doi: 10.1007/978-1-4302-5990-9_4.

[17]. *Y. Lin, A. A. Heidari, S. Wang, H. Chen, and Y. Zhang, "An Enhanced Hunger Games Search Optimization with Application to Constrained Engineering Optimization Problems," Biomimetics*, vol. 8, no. 5, p. 441, Sep. 2023, doi: 10.3390/biomimetics8050441.

[18]. *S. Li, H. Chen, M. Wang, A. A. Heidari, and S. Mirjalili, "Slime mould algorithm: A new method for stochastic optimization," Future Generation Computer Systems*, vol. 111, pp. 300–323, Oct. 2020, doi: 10.1016/j.future.2020.03.055.

[19]. *S. Talatahari and M. Azizi, "Chaos Game Optimization: a novel metaheuristic algorithm," Artif Intell Rev*, vol. 54, no. 2, pp. 917–1004, Feb. 2021, doi: 10.1007/s10462-020-09867-w.

[20]. *N. Khodadadi, L. Abualigah, Q. Al-Tashi, and S. Mirjalili, "Multi-objective chaos game optimization," Neural Comput Appl*, vol. 35, no. 20, pp. 14973–15004, Jul. 2023, doi: 10.1007/s00521-023-08432-0.

[21]. *S. Mirjalili, S. M. Mirjalili, and A. Lewis, "Grey Wolf Optimizer," Advances in Engineering Software*, vol. 69, pp. 46–61, Mar. 2014, doi: 10.1016/j.advengsoft.2013.12.007.

[22]. *A. Dagal, A.-W. Ibrahim, A. Harrison, W. F. Mbasso, A. O. Hourani, and I. Zaitsev, "Hierarchical multi step Gray Wolf optimization algorithm for energy systems optimization," Sci Rep*, vol. 15, no. 1, p. 8973, Mar. 2025, doi: 10.1038/s41598-025-92983-w.

[23]. *K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan, "A fast and elitist multiobjective genetic algorithm: NSGA-II," IEEE Transactions on Evolutionary Computation*, vol. 6, no. 2, pp. 182–197, Apr. 2002, doi: 10.1109/4235.996017.

[24]. *A. Golchha and S. G. Qureshi, "Non-Dominated Sorting Genetic Algorithm-II – A Succinct Survey," 2015. [Online]. Available: https://api.semanticscholar.org/CorpusID:1794065*

[25]. *E. Zitzler, M. Laumanns, and L. Thiele, "SPEA2: Improving the strength pareto evolutionary algorithm," TIK Report*, vol. 103, 2001.

[26]. *Y. Guan et al., "Application of Strength Pareto Evolutionary Algorithm II in Multi-Objective Water Supply Optimization Model Design for Mountainous Complex Terrain," Sustainability*, vol. 15, no. 15, p. 12091, Aug. 2023, doi: 10.3390/su151512091.

[27]. *C.-L. Hwang and K. Yoon, Multiple Attribute Decision Making*, vol. 186. Berlin, Heidelberg: Springer Berlin Heidelberg, 1981. doi: 10.1007/978-3-642-48318-9.