Journal of Industrial and Management Optimization

JIMO, 22(4): 1928–1955.
DOI: 10.3934/jimo.2026071
Received: 27 May 2025
Revised: 23 February 2026
Accepted: 11 March 2026
Published: 20 March 2026

https://www.aimspress.com/journal/jimo

Research article

Developing a stacked ensemble model for construction labor productivity prediction with metaheuristic-optimized random forest

Abdirisak Mohamed Abdillahi[1,2] and Savaş Bayram[3,*]

[1] Graduate School of Natural and Applied Sciences, Erciyes University, Kayseri 38039, Türkiye
[2] African Institute for Multidisciplinary Studies, Mogadishu, Somalia
[3] Department of Civil Engineering, Faculty of Engineering, Erciyes University, Kayseri 38039, Türkiye

* **Correspondence:** Email: sbayram@erciyes.edu.tr; Tel: +90 352 207 66 66; Fax: +90 352 437 57 84.

Abstract: Accurately predicting construction labor productivity is essential for effective project management, particularly in resource-constrained settings like Somaliland. However, existing studies primarily rely on single-model structures, which often struggle to generalize across varying site conditions and activity types. Moreover, while metaheuristic optimization has been used to tune individual models, its integration into ensemble architectures remains limited, leaving a gap in methods capable of combining complementary learning behaviors. To address this, this study developed a stacked ensemble model designed to improve prediction stability and robustness. The proposed approach integrates two metaheuristically optimized Random Forest models, tuned using Particle Swarm Optimization and Artificial Bee Colony, with an Extreme Gradient Boosting (XGBoost) meta-learner. The model was evaluated on 1422 real-world productivity records from 52 construction projects involving concrete, formwork, and bricklaying tasks. The stacked ensemble achieved competitive and more stable performance, particularly for high-variance activities, demonstrating improved robustness, stability, and generalization. Feature selection results emphasized the importance of safety-related disruptions, weather conditions, and communication efficiency in determining labor productivity. The proposed model offers a practical and scalable tool for more reliable construction labor productivity (CLP) prediction and provides decision-making support for construction management, particularly in resource-limited environments.

Keywords: construction productivity; metaheuristic optimization; random forest; ensemble learning; XGBoost
Mathematics Subject Classification: Primary: 90B30; Secondary: 68T05, 68T20, 90C59

1. Introduction

Construction labor productivity is a crucial determinant of project success, directly influencing cost, schedule adherence, and overall efficiency. As labor constitutes one of the most significant cost components in construction projects [1], low productivity has been widely recognized as a key contributor to cost overruns and delays. This issue is particularly evident in developing economies, where resource constraints and inefficient management practices worsen these challenges. In Somaliland's emerging construction sector, factors such as lack of experience, financial constraints, payment delays, shortages of tools and equipment, and cultural influences significantly impact labor productivity, leading to financial and time overruns [2]. Addressing CLP challenges is important for ensuring project viability and fostering economic development.

Productivity, in its simplest form, is defined as the relationship between output and input [3]. However, estimating productivity in construction is complex, as it depends on many interrelated influencing factors [4]. Traditional CLP estimation methods, such as expert judgment and simple statistical models, have long been used to assess productivity. These methods often rely on historical data and subjective assessments, which may become outdated and fail to account for evolving industry conditions. Without a structured methodology, they risk oversimplifying productivity and overlooking complex interactions that influence it [5]. More advanced predictive techniques are thus required to improve accuracy and capture the dynamic nature of construction productivity.

Recent advancements in machine learning (ML) have demonstrated its effectiveness in analyzing large datasets, enhancing productivity management, and improving progress tracking. ML's data-driven capabilities and computational power may enable more precise estimations, optimize resource allocation, and automate scheduling and monitoring processes. As a result, ML-based models present a promising approach for improving CLP prediction and addressing the limitations of traditional estimation methods [6].

Studies have applied various ML algorithms, including Decision Trees, Support Vector Machines (SVMs), Artificial Neural Networks (ANNs), Random Forest (RF), Deep Neural Networks (DNN), and Gradient Boosting, among others, to model CLP [7–11]. These models have demonstrated superior predictive capabilities compared to traditional methods. However, ML models still face challenges, including overfitting, hyperparameter tuning, and difficulty selecting relevant input features from large datasets [12]. Although ML models such as ANN, SVM, K-Nearest Neighbors (KNN), and RF have demonstrated notable success in CLP prediction, each method has important limitations when applied in isolation. ANN models are powerful in capturing nonlinear relationships, yet they often suffer from low interpretability, sensitivity to parameter settings, and a tendency to overfit when datasets are moderate in size, as is common in construction studies [8]. SVM models can achieve strong accuracy, but their performance typically depends on careful kernel selection and becomes computationally expensive as the dataset size grows [9]. KNN is simple and intuitive, but its predictions are highly sensitive to data scaling and local noise, which can lead to instability in field-collected datasets [7]. RF models provide robustness and built-in feature importance analysis; however, they may struggle in

high-variance datasets where productivity fluctuates sharply between observations, potentially reducing generalization across activity types. These complementary strengths and weaknesses suggest that no single algorithm consistently performs best across heterogeneous construction conditions. This motivates the use of ensemble strategies, particularly stacking, which can leverage the unique learning behavior of multiple optimized models to improve overall predictive reliability.

Metaheuristic optimization techniques, such as Particle Swarm Optimization (PSO) [13], Genetic Algorithms (GA) [14,15], Ant Colony Optimization (ACO) [16,17], Artificial Bee Colony (ABC) [18,19], Whale Optimization Algorithm (WOA) [20,21], and Grey Wolf Optimizer (GWO) [22,23], among others, have been used to enhance prediction by optimizing feature selection and model parameters in many research fields. These algorithms mimic natural evolutionary and swarm intelligence (SI) processes to search large spaces, improving ML model strength and accuracy. Metaheuristic-optimized models outperform conventional ML approaches regarding generalizability and prediction reliability by filtering out irrelevant features and fine-tuning hyperparameters. However, many studies focus on standalone metaheuristic-optimized models rather than their integration into advanced ensemble frameworks.

While machine learning models provide valuable insights into CLP without an optimization component, ML models estimate productivity levels without offering actionable recommendations for improvement. Optimization techniques can be integrated with ML models to explore and adjust key variables to bridge this gap.

This research proposes a novel stacked ensemble learning model for CLP prediction, integrating metaheuristically optimized base models. The proposed approach leverages two Random Forest models, each optimized using a different metaheuristic to refine feature selection and parameter tuning. An Extreme Gradient Boosting model (XGBoost) also serves as the meta-learner in this methodology, combining the outputs of the optimized RF models to enhance predictive accuracy. This method addresses key limitations in previous research by improving model generalization and reducing error margins through ensemble learning. The study's contributions are as follows: (1) developing a metaheuristic-optimized RF framework for CLP, (2) introducing a stacked ensemble model combining RF-PSO and RF-ABC with XGBoost, and (3) demonstrating the effectiveness of this approach using actual construction project data, with a focus on applications relevant to the construction sector.

To contextualize the ensemble approach adopted in this study, it is important to first understand the strengths and limitations of common ensemble techniques in construction-related prediction tasks. Bagging and boosting are widely used ensemble techniques that improve model performance by reducing variance and bias, respectively, and have been widely adopted in construction engineering to enhance predictive accuracy [24,25]. However, boosting is sensitive to noisy data and prone to overfitting as it focuses on misclassified instances, while bagging mainly reduces variance but struggles with high-bias models, especially simpler ones. Stacking addresses these limitations by leveraging diverse models to enhance generalization and strength [26].

While numerous studies have applied machine-learning models such as ANN, SVM, RF, GRNN, MART, and various hybrid or evolutionary approaches for construction labor productivity prediction, most existing work is limited by relatively small datasets, single-algorithm modeling structures, or evaluation strategies that do not adequately test generalization across projects [8,12,27,28]. This study advances the field by introducing a relatively large, multi-project dataset (1422 records from 52 projects), a comprehensive metaheuristic optimization framework applied to Random Forest using five algorithms (ACO, ABC, PSO, GWO, GA), and a stacked ensemble that integrates the strongest

metaheuristic-optimized models. In addition, the study employs project-grouped cross-validation, an evaluation strategy rarely reported in CLP prediction studies, thereby providing a more realistic assessment of performance on unseen projects. Collectively, these contributions provide a more reliable, scalable, and generalizable predictive framework compared with existing ML-based CLP studies.

While previous studies have explored metaheuristic optimization for individual ML models, little attention has been given to integrating multiple optimized models within a stacked ensemble framework. The rest of this paper is structured as follows: Section 2 details the methodology, including data overview, model development, and optimization strategies. Section 3 presents a discussion of the results, and Section 4 summarizes key findings and outlines directions for future research.

2. Materials and methods

2.1. Data overview

This study utilizes a dataset comprising 1422 records collected from 52 construction projects across four major urban centers in Somaliland: Berbera, Borama, Burao, and Hargeisa. It covers labor productivity across three key activity types: bricklaying (532 records), formwork (480 records), and concrete (410 records). The data was collected through direct on-site observations of daily work activities.

The dataset includes nine input variables and one output variable (productivity). The input variables consist of both ordinal categorical and continuous factors. Ordinal variables include material availability, weather conditions, communication, equipment availability, and working space conditions, each rated on a three-point scale. Trained data collectors assigned the ordinal ratings under the supervision of the authors using predefined qualitative criteria applied consistently across all sites. Ratings were recorded daily based on on-site observations. Within each project, the same assigned observer performed the evaluations throughout the data collection period, which helped limit subjectivity and ensure internal consistency across observations.

The three-point scores were assigned using a consistent rubric: “1” indicates frequent disruption or inadequacy affecting daily work, “2” indicates occasional or moderate limitation, and “3” indicates adequate conditions with no meaningful disruption. Prior to data collection, observers were briefed using example site scenarios to standardize the interpretation of the rubric across projects.

Continuous variables include rework frequency, accident frequency, working hours, and gang size. The output variable, productivity, is defined as the daily output per laborer per hour. In this study, work done is measured using activity-specific physical output units consistent with site practice. For bricklaying activities, work done corresponds to the number of bricks laid per day, recorded by the site supervisor. For formwork activities, work done is measured in square meters of formwork installed, while for concrete activities, it is measured in cubic meters of concrete placed. Although the productivity formula is identical across activities, these different output units naturally result in different numerical productivity scales.

Daily activity outputs were recorded from the site’s daily measurement sheets and progress logs and cross-checked with the responsible site supervisor/foreman. Bricks were counted as completed units, formwork quantity was recorded as installed area (m^2) based on measured panels, and concrete quantity was recorded as placed volume (m^3) based on pour records.

The variables and their coding structures are described in Table 1.

Table 1. Description of variables used in the study.

No	Variable	Type	Description	Coding/unit
1	Material availability	Ordinal	Consistency of material supply on site	1 = low, 2 = medium, 3 = high
2	Weather conditions	Ordinal	Suitability of the weather for site operations	1 = poor, 2 = moderate, 3 = good
3	Communication	Ordinal	Efficiency of communication among workers	1 = poor, 2 = medium, 3 = good
4	Equipment availability	Ordinal	Availability of tools and equipment required for work	1 = low, 2 = medium, 3 = high
5	Working space conditions	Ordinal	Level of space adequacy for crew movement	1 = cramped, 2 = adequate, 3 = spacious
6	Rework frequency	Continuous	Number of rework events per day	Count/day
7	Accident frequency	Continuous	Number of accidents involving the crew per day	Count/day
8	Working hours	Continuous	Total hours worked in a day	Hours/day
9	Gang size	Continuous	Number of laborers in the crew	Count
10	Productivity	Continuous (Target)	Output per labor-hour	Work done / (crew size × hours)

These variables were selected because they are consistently identified in construction research as primary drivers of labor productivity. They capture material flow, crew coordination, work conditions, and operational disruptions, which directly influence output rates on site.

2.2. *Variable description and preprocessing*

To ensure data quality and consistency for machine learning applications, several preprocessing steps were applied to account for the heterogeneous nature of the dataset. The input variables include both ordinal categorical and continuous features. Ordinal variables were numerically encoded based on predefined ranking scales reflecting their relative qualitative importance.

All continuous variables were standardized using the StandardScaler technique, which transforms the data to have a mean of zero and a standard deviation of one [29]. This transformation normalizes the dataset by subtracting the mean from each value and dividing by the standard deviation. The standardization process is expressed as follows:

$$X_{Standardized} = \frac{X - \bar{X}}{\sigma X} \tag{1}$$

where X represents the original feature value, $\bar{x}$ is the feature mean, and σX denotes the standard deviation.

These preprocessing steps were applied consistently across all datasets before model development. Feature selection and model training procedures are described separately in the subsequent methodological sections.

2.3. Model evaluation strategy

Two evaluation strategies were employed in this study. Baseline models (MLR, KNN, SVMR, ANN) and standalone Random Forest (RF) models were initially evaluated using a conventional 70/30 train/test split, consistent with prior CLP prediction studies. All preprocessing steps were fitted on the training data and applied to the test data only.

Because multiple observations originate from the same construction project, the final stacked ensemble model was evaluated using project-grouped cross-validation to prevent project-level data leakage and to assess generalization to unseen projects. To enable a direct and fair comparison under identical validation conditions, the Random Forest baseline and metaheuristic-optimized Random Forest variants were also evaluated using the same project-grouped cross-validation strategy. Under this protocol, all preprocessing steps and model training were performed exclusively within each training fold, with no project overlap between training and testing data. In the grouped cross-validation setting, scaling (StandardScaler), feature selection, and model fitting were performed within each training fold only and then applied to the corresponding held-out fold to ensure a leak-free evaluation pipeline.

For project-grouped cross-validation, a 5-fold GroupKFold strategy was employed, where the grouping variable was the construction project identifier, ensuring that all records from the same project were assigned to the same fold. Given the total of 52 projects, each fold contained approximately 10–11 projects, with the number of observations per fold varying according to project size. The fold partitioning was deterministic (i.e., using a fixed group partition without repetition), and model performance was summarized using the mean and standard deviation of evaluation metrics across the five folds. This approach provides a leakage-free and transparent assessment of generalization to unseen projects.

Project grouping was implemented using GroupKFold (n_splits = 5) with the construction project identifier as the group label. Because GroupKFold does not shuffle groups by default, the split was deterministic. Results are reported as mean ± standard deviation across the five folds.

2.4. Exploratory data analysis

The descriptive statistics indicate substantial variability in both crew-related and operational factors across the three activity categories. Accident frequency and rework frequency show the highest coefficients of variation, reflecting their irregular but disruptive nature on site. Gang size and productivity exhibit broader ranges, suggesting differences in resource allocation and task complexity among projects (Table 2). The different numerical ranges observed for productivity across activities reflect the use of activity-specific output units (brick counts for bricklaying, square meters for formwork, and cubic meters for concrete), rather than differences in calculation methodology. In contrast, working hours remain relatively stable across activities. The variability observed in the dataset indicates nonlinear interactions between input factors and labor productivity. This justifies the application of machine-learning models, which are capable of capturing these complex, non-additive relationships more effectively than traditional linear approaches.

Table 2. Summary of descriptive statistics for the key variables across activity types.

Variable	Activity	Mean	Std	CV (%)	Range (min–max)
Material availability	Bricklaying	2.68	0.59	22.02	1–3
	Formwork	2.70	0.54	19.98	1–3
	Concrete	2.66	0.55	20.56	1–3
Accident frequency	Bricklaying	0.14	0.40	280.30	0–2
	Formwork	0.27	0.60	227.40	0–3
	Concrete	0.22	0.56	253.10	0–3
Rework frequency	Bricklaying	0.22	0.48	223.67	0–2
	Formwork	0.24	0.56	234.70	0–4
	Concrete	0.22	0.47	218.11	0–2
Working hours	Bricklaying	7.22	0.96	13.35	5–8
	Formwork	7.46	0.84	11.21	5–8
	Concrete	7.49	0.84	11.25	5–8
Gang size	Bricklaying	6.72	3.60	53.53	2–13
	Formwork	9.21	3.26	35.35	6–20
	Concrete	10.41	3.13	30.11	6–16
Productivity (target)	Bricklaying	23.91	4.35	18.20	12.5–35
	Formwork	1.45	0.15	9.95	1.10–1.80
	Concrete	1.36	0.10	7.65	1.10–1.55

Bricklaying exhibits a wider productivity range with several high-output outliers, while formwork and concrete show narrower, more stable productivity distributions as illustrated in Figure 1.

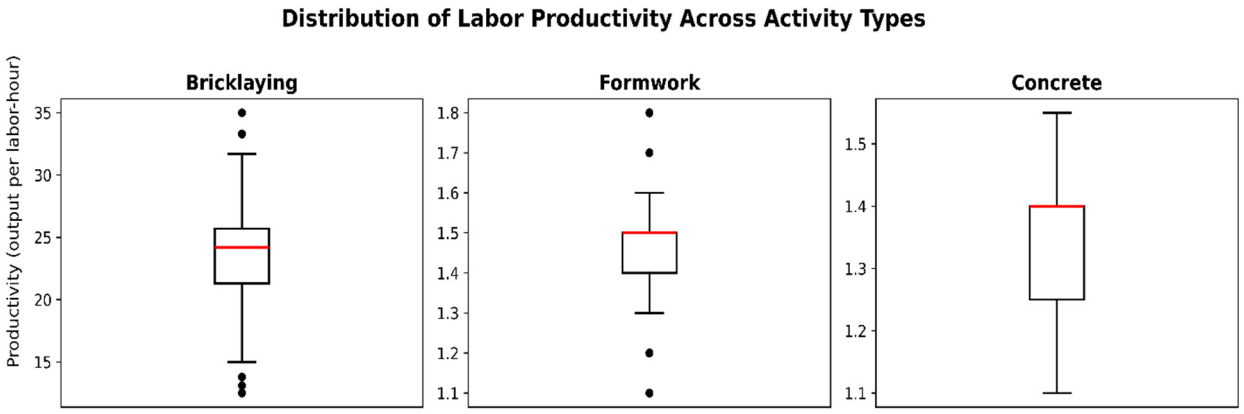

Figure 1. Distribution of labor productivity across the three construction activity types.

These statistical patterns show the presence of heterogeneous and interacting factors influencing labor productivity, forming the basis for the subsequent modeling and optimization approaches described in Section 2.5 onward.

2.5. Baseline predictive models

To ensure that the performance of the proposed stacked ensemble is evaluated fairly, four widely used predictive models were included as baselines: multiple linear regression (MLR), K-nearest neighbors (KNN), support vector machine regression (SVMR), and artificial neural networks (ANN). These models are standard benchmarks in construction labor productivity prediction research and are commonly used to evaluate improvements in nonlinear modeling capability and generalization performance [7,9,27,30]. The same training–testing procedure and performance metrics were applied across all models to ensure comparability.

Random Forest (RF) was used as the primary nonlinear reference model because it consistently demonstrated strong predictive performance and robustness in construction labor productivity estimation. In this study, RF serves as the strongest single-model benchmark against which the improvements achieved by the stacked ensemble are evaluated. As an ensemble learning method, RF combines multiple decision trees trained on bootstrapped subsets of the data, reducing overfitting and improving generalization through a technique known as bagging [31,32]. It effectively captures nonlinear relationships, offers stable predictions, and provides built-in mechanisms for feature importance analysis. To enhance RF performance, hyperparameters must be carefully tuned, as suboptimal configurations can significantly impact accuracy. Grid search with cross-validation is commonly used for this purpose [29]. However, manual tuning can be time-consuming and limited in scope.

2.6. Metaheuristic optimization framework

Building on the limitations of manual RF hyperparameter tuning, the study incorporated metaheuristic optimization techniques capable of simultaneously selecting optimal feature subsets and tuning model hyperparameters. These algorithms are well-suited for complex search spaces, offering a balanced exploration–exploitation process without requiring gradient information. A dual-objective function was formulated to guide the optimization toward models that are both accurate and efficient. The two objectives, minimizing prediction error and minimizing feature deviation, are detailed below:

Goal 1: Minimizing prediction error

The first objective is to minimize the root mean squared error (RMSE) to improve prediction accuracy for CLP.

The RMSE is computed as [33]:

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

where y^i and $\hat{y}_i$ represent the actual and predicted values, respectively, and n is the number of observations. The optimization goal is expressed as:

$$\text{Goal 1} = \min (\text{RMSE}) \tag{3}$$

RMSE was selected as the primary optimization objective because it places greater weight on larger errors, which is important in construction settings where even occasional severe productivity drops can result in significant schedule delays and cost overruns. Model evaluation incorporated multiple complementary performance metrics to ensure a comprehensive assessment of both predictive

accuracy and generalization. This choice is appropriate for strategic (non–real-time) planning contexts, where preventing large deviations is more critical than minimizing uniform error magnitudes.

Goal 2: Minimizing feature deviation

The second objective emphasizes practical applicability, aligning with insights from Ebrahimi et al. [13], who noted that organizations often favor minimal, manageable adjustments over large-scale changes to enhance productivity. Such an approach avoids the need for extensive overhauls or new strategies, focusing instead on cost-effective and operationally feasible improvements within existing frameworks.

The average feature deviation is computed as:

$$\text{Average Feature Deviation} = \frac{1}{k}\sum_{j=1}^{k}\left|\mu_j^{selected} - \mu_j^{global}\right| \tag{4}$$

Where k is the number of selected features, $\mu_j^{selected}$ is the mean of the selected subset, and μ_j^{global} is the mean across the entire dataset. The optimization goal is:

$$\text{Goal 2} = \min\ (\text{Average Feature Deviation}) \tag{5}$$

To balance accuracy and practical feature selection, the two objectives are combined in a weighted function:

$$\text{Objective} = w_1\,(RMSE) + w_2\,(\text{Average Feature Deviation}) \tag{6}$$

Where $w_1 + w_2 = 1$.

To assess the influence of the weighting scheme in the bi-objective function, a sensitivity analysis was conducted using six candidate values of $w_1 = \{0.2, 0.5, 0.6, 0.7, 0.8, 0.9\}$ with $w_2 = 1 - w_1$. For each configuration, the complete optimization procedure was repeated across all three activity datasets using the same train/test setup applied during model development. Across all datasets and weight values, the resulting RMSE, average feature deviation, and selected feature subsets remained effectively unchanged. In several cases, the optimization procedure converged to identical feature combinations and identical RMSE and deviation values (to three decimal places), indicating that the optimization landscape is highly stable with respect to moderate variations in the weighting scheme.

Table 3 summarizes the observed ranges of RMSE and feature deviation across the evaluated weight values, demonstrating that the model outcomes are not materially affected by the choice of w_1 within this interval. This empirical robustness confirms that the proposed framework is not sensitive to the specific weighting choice.

Table 3. Sensitivity analysis of the weighted objective function across different values of w_1.

W_1	RMSE (range across datasets)	Feature deviation (range)
0.2	0.014–2.074	0.002–0.015
0.5	0.014–2.074	0.010–0.015
0.6	0.014–2.074	0.010–0.015
0.7	0.014–2.074	0.010–0.015
0.8	0.014– 2.074	0.010–0.015
0.9	0.014–2.074	0.010–0.015

The RMSE and feature deviation ranges reported in Table 3 correspond to the minimum and maximum values observed across all evaluated weight configurations (w_1 = 0.2–0.9) and across the three activity datasets. The values are therefore pooled across activities rather than reported separately for each dataset. Importantly, across all tested weight settings, the resulting RMSE values and the selected feature subsets remained effectively unchanged, indicating that the optimization outcomes are robust to moderate variations in the weighting scheme.

The pooled RMSE range is wide because the three activities have different output units and target scales (brick counts versus m^2 and m^3), whereas within each activity, the RMSE and selected feature subset remained unchanged across the tested weights.

Given this stability, $w_1 = 0.7$ was retained as a balanced and interpretable setting, representing a commonly used compromise in weighted multi-objective optimization and avoiding overemphasis on any single objective [34].

The observed insensitivity of RMSE, feature deviation, and selected feature subsets to moderate variations in w_1 indicates that the two objectives are well aligned for the studied datasets rather than redundant. In practice, the feature-deviation term constrains the search by discouraging non-representative feature subsets, while the RMSE term determines the final optimum once this constraint is satisfied. As a result, the optimization converges to stable solutions that simultaneously satisfy predictive accuracy and practical representativeness.

To ensure reproducibility, fixed random seeds were used for model training and evaluation (e.g., random_state = 42 for Random Forest models and data splitting). Each metaheuristic optimization was executed once per activity dataset using a fixed iteration budget, consistent with common practice in construction productivity studies. Although metaheuristic optimization increases offline training time, runtime remained manageable and does not affect practical deployment, as the proposed CLP models are intended for strategic planning rather than real-time prediction.

All experiments were implemented in Python 3.12.6 on a Windows 11 (64-bit) environment. The models were developed using scikit-learn (v1.5.2) for preprocessing and Random Forest implementation, and XGBoost (v2.1.4) for the meta-learner. Supporting libraries included NumPy (v1.26.4) and pandas (v2.3.3). Fixed random seeds (random_state = 42) were applied wherever supported to ensure reproducibility.

2.7. Metaheuristic algorithms

To solve this dual-objective optimization problem, five metaheuristic algorithms were applied to optimize the RF models: Ant Colony Optimization (ACO), Artificial Bee Colony (ABC), Grey Wolf Optimizer (GWO), Genetic Algorithm (GA), and Particle Swarm Optimization (PSO). These algorithms were selected based on their proven track record in solving complex search problems in engineering and data-driven modeling contexts. Each algorithm was implemented to search for optimal combinations of RF hyperparameters and relevant feature subsets. Their comparative performance across different activity datasets is analyzed in Section 3 to determine their suitability for predicting construction labor productivity.

The first optimization method used in this endeavor is ACO, introduced by Dorigo et al. [35] and based on the pheromone-guided foraging behavior of ants. As a SI technique, ACO has been widely used in combinatorial optimization due to its adaptability and efficiency in discovering near-optimal

solutions [36]. In machine learning applications, ACO has proven effective for feature selection and tuning, particularly in problems involving nonlinear relationships.

The second optimization algorithm considered, ABC, was developed by Karaboga [37] and simulates the foraging strategies of honey bees. It divides the bee population into employed, onlooker, and scout bees that collaboratively balance exploration and exploitation in the search space [38]. ABC has been shown to perform well in multi-modal optimization problems [39], with studies demonstrating its strength and simplicity across a range of tasks [40].

The third technique, GWO, was introduced by Mirjalili et al. [41] and models the leadership hierarchy and hunting patterns of grey wolves. The algorithm categorizes candidate solutions into alpha, beta, delta, and omega roles, which guide search behavior. GWO has been successfully applied to engineering and optimization problems due to its structured exploration-exploitation balance [42,43], making it suitable for improving predictive models in construction contexts.

The fourth algorithm, GA, is inspired by the principles of natural selection [44]. It evolves candidate solutions through genetic operators like selection, crossover, and mutation, effectively avoiding local optima in complex search spaces. GA's flexibility and strength have led to its widespread use in feature selection and hyperparameter optimization, especially when seeking to enhance predictive accuracy [43].

Finally, PSO, developed by Kennedy and Eberhart [45], mimics the coordinated movements of bird flocks and fish schools. As another SI algorithm, PSO enables fast convergence and has been successfully used in data mining, engineering, and computational intelligence tasks. Its ability to balance speed and accuracy makes it a strong candidate for tuning machine learning models [46].

The implementation of each metaheuristic followed a structured tuning procedure. The parameter ranges evaluated during grid search, together with the final selected values, are summarized below to describe the implementation framework.

In the case of ACO, the grid search evaluated 10–20 ants, an evaporation rate of 0.3–0.5, 30 iterations, and pheromone–heuristic factors with α fixed at 1.0 and β in the range 1.0–2.0 (α was not varied in the grid search). Across datasets, the final configuration consistently selected was 10 ants, 30 iterations, an evaporation rate of 0.5, and $\alpha = \beta = 1.0$.

ABC tuning examined the number of employed bees (10–20), onlooker bees (10–20), and scout limit (10–50); 30–50 iterations were tested, with 20 employed bees, 20 onlookers, a scout limit of 10, and 30 iterations selected.

PSO tuning evaluated 30–50 particles, 30–50 iterations, inertia weights between 0.5 and 0.9, cognitive/social coefficients in the range 1.5–2.0, and maximum velocity values of 2–4, with the final configuration using 40 particles, 30 iterations, a fixed inertia weight of 0.6, $c_1 = c_2 = 1.5$, and max velocity = 3.

GWO experiments examined population sizes of 10–20 wolves, 20–30 iterations were examined, and 20 wolves with 30 iterations were used in the final model.

GA tuning explored population sizes 20–30, crossover rates of 0.6–0.9, mutation rates of 0.01–0.1, and fixed 30 iterations were tested, with the final configuration using a population of 30, a crossover rate of 0.8, and a mutation rate of 0.05. Across all algorithms, the stopping criterion was a fixed iteration limit rather than a convergence threshold, ensuring consistency across models.

To evaluate model performance, the study used eight widely used regression metrics: mean absolute percentage error (MAPE), root mean square error (RMSE), coefficient of determination (R^2), relative root means square error (RRMSE), Nash–Sutcliffe efficiency (NSE), Kling–Gupta efficiency

(KGE), overall index of model performance (OI), and mean square error (MSE). These metrics were selected to ensure a comprehensive and strong assessment of model accuracy, consistency, and generalization.

2.8. Stacked ensemble framework

In this study, a stacked ensemble model was developed to enhance CLP prediction by integrating multiple metaheuristically optimized base learners. Unlike simple ensemble methods such as bagging or voting, stacked generalization allows for learning how to best combine the strengths of individual models using a secondary learner. This two-level architecture consists of optimized base learners at the first level and a meta-learner at the second level that synthesizes their outputs to produce the final prediction.

The base learners in this framework are Random Forest models optimized separately using Particle Swarm Optimization (PSO) and Artificial Bee Colony (ABC). These two algorithms were selected because they demonstrated consistently strong and stable predictive performance across the activity datasets during comparative evaluation. In contrast, while ACO, GWO, and GA achieved competitive performance on concrete and formwork, they exhibited weaker or less stable results on bricklaying, including higher error magnitudes and greater sensitivity to data variability. Including these less stable optimizers as base learners did not provide additional performance benefits while increasing ensemble complexity and variance. Retaining only RF-PSO and RF-ABC, therefore, ensured that the stacking inputs were derived from the most robust and consistently performing optimizers, resulting in a more stable and computationally efficient ensemble without unnecessary complexity.

Extreme Gradient Boosting (XGBoost) was employed as the meta-learner due to its ability to model nonlinear relationships and interactions among base predictions efficiently. XGBoost combines gradient boosting and regularization, making it well-suited for learning from the outputs of multiple predictive models. Its success in structured data tasks and previous evidence supporting its use in stacking architectures motivated its adoption in this study [47,48].

To prevent information leakage and ensure valid generalization assessment, all base-model predictions used in stacking were generated through an out-of-fold procedure. For each fold, a base model was trained only on that fold's training portion, and its predictions for the corresponding held-out portion were stored. The meta-learner was then trained exclusively on these out-of-fold predictions, ensuring that it never received predictions obtained from data that the base models had already seen. This procedure ensures that both training and evaluation strictly reflect unseen conditions, eliminating leakage at both the base-model and meta-learner levels.

While stacking remains relatively underexplored in construction management research, emerging evidence supports its potential. For instance, Karatas and Budak [27] showed that stacking outperformed individual models in labor productivity prediction tasks. By leveraging the complementary strengths of different optimization strategies, the proposed stacking framework in this study aims to improve both the accuracy and generalizability of construction labor productivity predictions.

3. Results and discussion

3.1. Optimization algorithm applications

The benchmarking results show clear performance differences among the baseline models. MLR performed adequately only on the concrete dataset, where factor relationships were mostly linear. KNN and SVMR improved prediction in nonlinear cases but were sensitive to data variability, particularly in the bricklaying dataset. ANN captured complex interactions but showed less stable generalization. These comparisons, summarized in Table 4, confirm that traditional regression and standard machine learning methods provide useful benchmarks but exhibit limited consistency across different construction activities.

Table 4. Baseline model performance across trades (test set results).

Dataset	Model	RMSE	MAPE (%)	R^2
Concrete	KNN	0.031	0.947	0.918
	SVMR	0.060	3.970	0.689
	RF	0.017	0.539	0.976
	ANN	0.049	3.149	0.770
	MLR	0.024	1.263	0.952
Formwork	KNN	0.074	3.132	0.733
	SVMR	0.081	4.436	0.677
	RF	0.067	2.614	0.777
	ANN	0.084	4.801	0.637
	MLR	0.075	3.954	0.718
Bricklaying	KNN	2.920	8.309	0.589
	SVMR	2.732	8.703	0.641
	RF	2.245	7.287	0.757
	ANN	2.373	7.615	0.688
	MLR	3.061	10.161	0.549

A Random Forest (RF) model tuned via grid search with 10-fold cross-validation demonstrated the strongest performance among the single models and therefore serves as the primary benchmark (Table 4). On the concrete dataset, RF achieved high accuracy (RMSE = 0.017, MAPE = 0.539%, R^2 = 0.976). The formwork dataset also showed strong performance (RMSE = 0.067, MAPE = 2.614%, R^2 = 0.777), with stable train/test consistency. Although the bricklaying dataset exhibited higher variability, RF remained competitive (RMSE = 2.245, MAPE = 7.287%, R^2 = 0.757). These results indicate that RF provides a strong and reliable foundation for further performance improvement.

Figure 2 presents the comparative performance of the baseline models using the OI (test set), where RF consistently outperforms the other models across all three activity types.

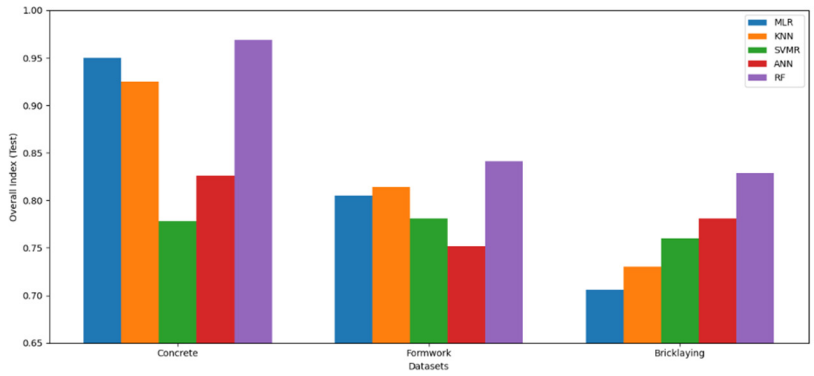

Figure 2. Baseline model performance comparison across activity datasets.

These results confirm that the performance enhancement of the stacked ensemble cannot be attributed to incremental tuning of a single model. Instead, the ensemble benefits from combining complementary learning patterns across the base learners, leading to improved generalization and better predictive reliability under varying site conditions.

Following the dual objective outlined above, first, the ACO was used to optimize feature selection and reduce prediction error in CLP estimation. Parameters such as ant count, pheromone influence, and evaporation rate were tuned via grid search. Across all three datasets, RF-ACO performed consistently well (Tables 5–7). On concrete, it achieved RMSE = 0.0157 and R^2 = 0.9789; on formwork, RMSE = 0.062 and R^2 = 0.81. Bricklaying showed higher variability (RMSE = 2.185), yet RF-ACO maintained solid generalization.

Table 5. Performance comparison of metaheuristic-optimized RF models on the concrete dataset.

Metric		RF-ACO	RF-ABC	RF-GWO	RF-GA	RF-PSO
MSE	Train	0.0001	0.0001	0.0001	0.0001	0.0001
	Test	0.0002	0.0002	0.0002	0.0003	0.0002
RMSE	Train	0.0113	0.0110	0.0112	0.0090	0.0107
	Test	0.0157	0.0150	0.0160	0.0161	0.0155
MAPE (%)	Train	0.3591	0.3510	0.3510	0.2640	0.3230
	Test	0.4959	0.4560	0.4669	0.4981	0.4880
R^2	Train	0.9875	0.9870	0.9877	0.9910	0.9890
	Test	0.9789	0.9790	0.9780	0.9780	0.9790
RRMSE	Train	0.8331	0.8250	0.8265	0.6950	0.7862
	Test	1.1637	1.1470	1.1579	1.1990	1.1503
NSE	Train	0.9875	0.9870	0.9877	0.9911	0.9890
	Test	0.9789	0.9790	0.9780	0.9780	0.9790
KGE	Train	0.9881	0.9881	0.9889	0.9900	0.9890
	Test	0.9856	0.9801	0.9703	0.9800	0.9760
Overall Index	Train	0.9811	0.9810	0.9814	0.9850	0.9830
	Test	0.9720	0.9720	0.9700	0.9710	0.9720

Next, the ABC algorithm was implemented, with grid search used to tune bee roles and scout parameters. ABC delivered strong predictive results across datasets, particularly for concrete and bricklaying ($R^2 > 0.97$ and 0.79, respectively). The OI remained high, affirming its strength.

The GWO was tuned using grid search for pack size, iteration count, and control coefficients. GWO maintained high accuracy across datasets, with RMSE values of 0.0160 (concrete), 0.0632 (formwork), and 2.0667 (bricklaying). R^2 scores remained stable (0.9795, 0.8020, and 0.7902), confirming strong generalization.

The GA used a structured grid search to adjust population, crossover, and mutation rates. GA performed competitively on concrete (RMSE: 0.016, R^2: 0.978) and formwork (RMSE: 0.062, R^2: 0.809), with minimal overfitting. Performance on bricklaying declined (RMSE: 2.626, R^2: 0.668), likely due to higher variability. Despite this, GA maintained strong KGE values and overall indices above 0.76.

The PSO was applied using grid search to fine-tune particle size, iteration count, and coefficients. PSO achieved the most consistent results across all three datasets. On concrete and formwork, it matched or exceeded the performance of other models (Tables 5 and 6). For bricklaying, the model achieved one of the lowest errors among the models on bricklaying (RMSE: 2.05, R^2: 0.798), as shown in Table 7.

Table 6. Performance comparison of metaheuristic-optimized RF models on the formwork dataset.

Metric		RF-ACO	RF-ABC	RF-GWO	RF-GA	RF-PSO
MSE	Train	0.0030	0.0030	0.0029	0.0026	0.0030
	Test	0.0040	0.0040	0.0038	0.0039	0.0040
RMSE	Train	0.0500	0.0550	0.0505	0.0510	0.0500
	Test	0.0620	0.0630	0.0632	0.0620	0.0620
MAPE (%)	Train	1.9080	2.0890	1.9142	1.9140	1.9090
	Test	2.5440	2.6630	2.6908	2.5490	2.5490
R^2	Train	0.8800	0.8700	0.8795	0.8800	0.8800
	Test	0.8100	0.8020	0.8020	0.8090	0.8090
RRMSE	Train	3.4620	3.7850	3.4753	3.4750	3.4680
	Test	4.2560	4.3240	4.3720	4.2701	4.2710
NSE	Train	0.8800	0.8700	0.8795	0.8801	0.8800
	Test	0.8100	0.8020	0.8020	0.8090	0.8090
KGE	Train	0.8950	0.8830	0.8943	0.8940	0.8960
	Test	0.9000	0.8940	0.8983	0.8980	0.9000
Overall Index	Train	0.9040	0.8960	0.9037	0.9040	0.9040
	Test	0.8610	0.8560	0.8500	0.8600	0.8600

Table 7. Performance comparison of metaheuristic-optimized RF models on the bricklaying dataset.

Metric		RF-ACO	RF-ABC	RF-GWO	RF-GA	RF-PSO
MSE	Train	2.3460	1.5200	1.5231	2.8690	1.5260
	Test	4.7750	4.2310	4.2390	6.8950	4.2050
RMSE	Train	1.5310	1.2330	1.2317	1.6940	1.2350
	Test	2.1850	2.0570	2.0667	2.6260	2.0500
MAPE (%)	Train	4.0730	3.2160	3.2175	4.6070	3.2350
	Test	6.7690	6.3500	6.4352	7.8122	6.3270
R^2	Train	0.8700	0.9160	0.9162	0.8420	0.9160
	Test	0.7700	0.7960	0.7902	0.6681	0.7980
RRMSE	Train	6.4041	5.1560	5.1500	7.0820	5.1650
	Test	9.1472	8.6100	8.6856	10.9910	8.5830
NSE	Train	0.8700	0.9160	0.9162	0.8420	0.9160
	Test	0.7700	0.7960	0.7902	0.6680	0.7980
KGE	Train	0.8880	0.9220	0.9212	0.8700	0.9210
	Test	0.7740	0.7880	0.7882	0.7190	0.7890
Overall Index	Train	0.9010	0.9310	0.9307	0.8830	0.9310
	Test	0.8360	0.8520	0.8400	0.7760	0.8530

Among all optimized models, RF-PSO and RF-ABC consistently demonstrated the most reliable performance across all datasets, achieving the lowest error margins and strongest generalization. While other models, such as RF-ACO and RF-GWO, also performed well, particularly on concrete and formwork, they showed more sensitivity to the variability of the bricklaying dataset. These findings reinforce the strength of PSO and ABC as effective optimization techniques for CLP prediction.

All five models demonstrated stable convergence trends. The RF-ABC and RF-ACO convergence plots are shown in Figures 3–8.

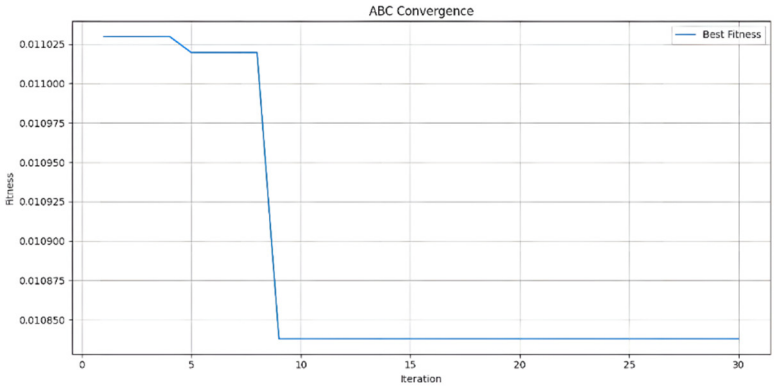

Figure 3. Convergence plot of RF-ABC concrete data.

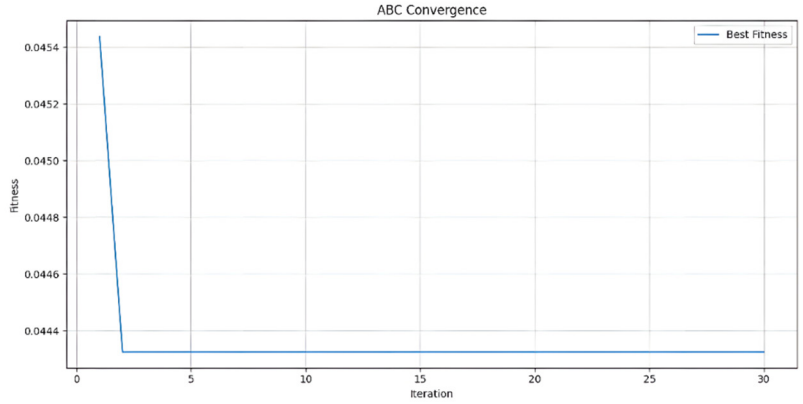

Figure 4. Convergence plot of RF-ABC formwork data.

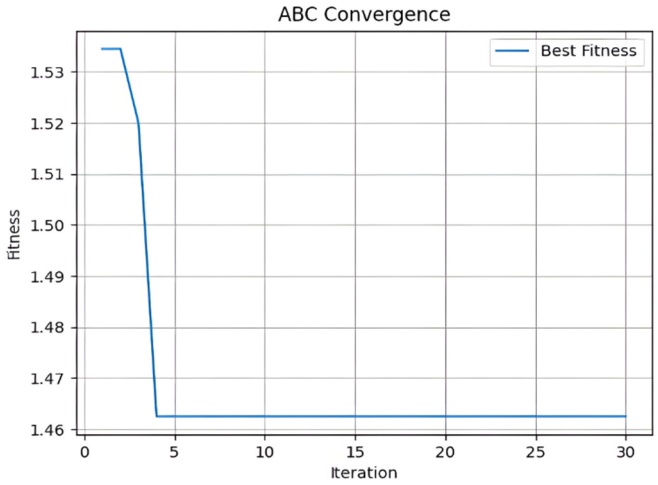

Figure 5. Convergence plot of RF-ABC bricklaying data.

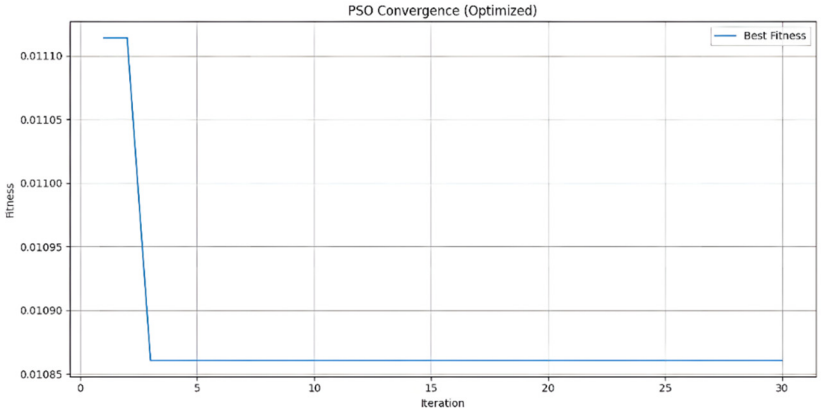

Figure 6. Convergence plot of RF-PSO concrete data.

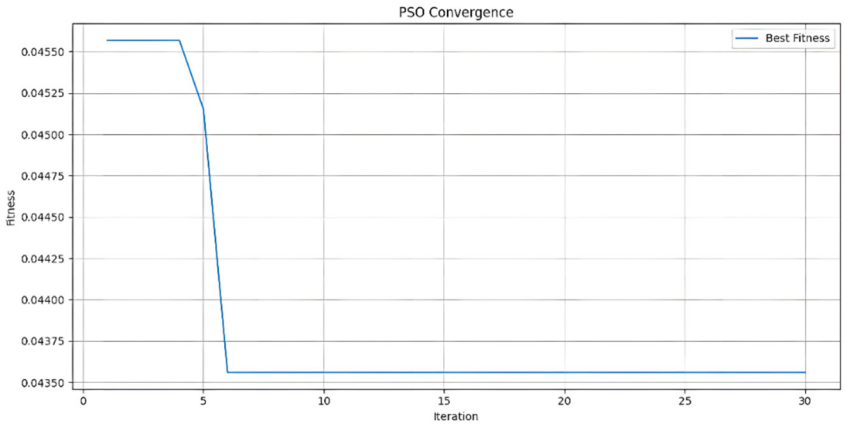

Figure 7. Convergence plot of RF-PSO formwork data.

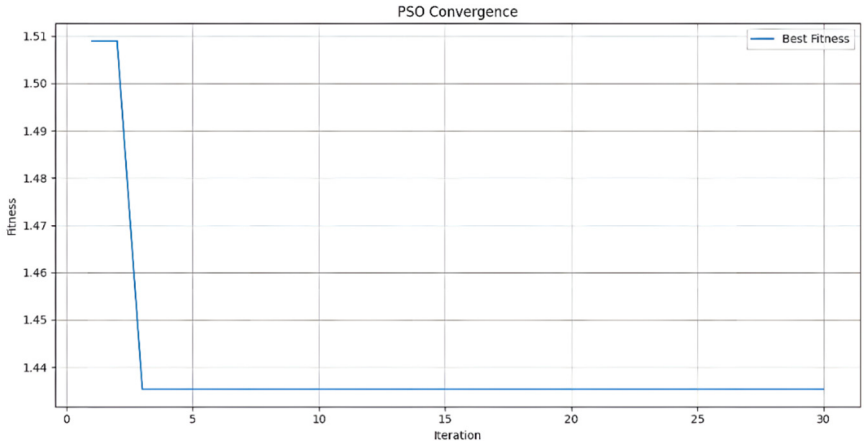

Figure 8. Convergence plot of RF-PSO bricklaying data.

3.2. Feature selection analysis

Across all five metaheuristic-optimized Random Forest models, a consistent set of influential variables emerged as the strongest predictors of CLP. Accident frequency, weather conditions, communication among workers, working space conditions, and working hours were selected by 100% of the models (Figure 9), indicating that these factors play a central role in shaping labor performance within the dataset analyzed.

These results closely align with the literature. For example, safety-related disruptions, captured through accident frequency, have long been recognized as a major source of productivity loss on construction sites, as accidents interrupt workflow continuity and reduce effective working time [2,4]. Likewise, the universal selection of weather conditions reflects well-documented evidence that environmental factors strongly affect productivity, especially in developing-country contexts where projects often rely on manual labor and lack mechanization buffers [2,4].

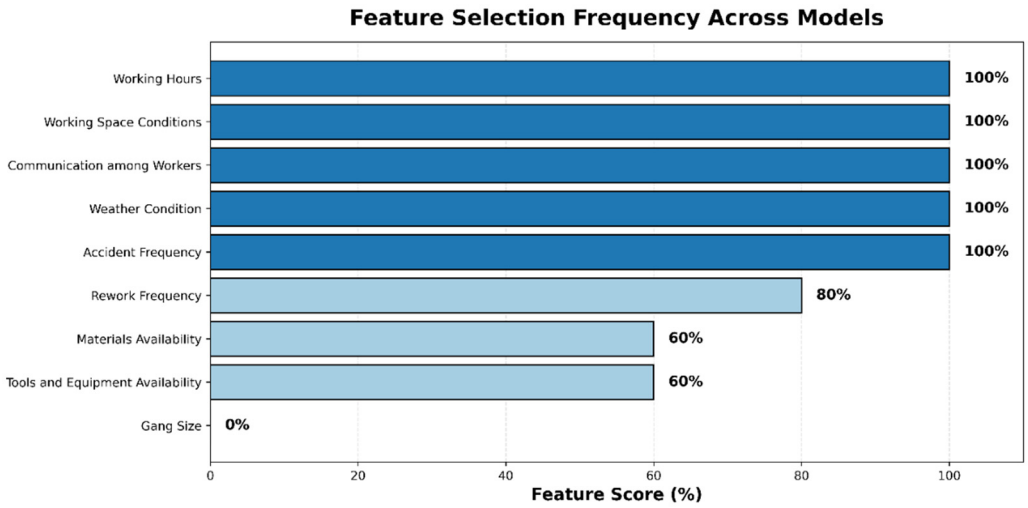

Figure 9. Feature selection frequency across models.

Communication among workers, also selected in all models, supports earlier discussions that coordination efficiency is a major determinant of site performance. Poor communication is associated with task delays, errors, and frequent stoppages, which are widely noted as common root causes of low productivity in construction environments [2,5]. Working space conditions were another universally selected feature, reinforcing the importance of task accessibility, site layout, and ease of movement. Congested sites impede workflow and increase idle time, consistent with prior research describing physical constraints as a key barrier to productive operations [4,5].

Working hours, selected by all models, reflect the direct relationship between labor time, fatigue, and daily output. This finding is consistent with productivity definitions and operational factors highlighted earlier in the manuscript, where labor time and work schedules were identified as critical drivers of output rates [3]. Rework frequency, selected in 80% of models, further supports the evidence that quality-related disruptions, rework, errors, and corrective cycles significantly reduce productivity [4,5]. Material and equipment availability, selected in 60% of models, also align with known site challenges in Somaliland, where shortages and delivery inconsistencies frequently interrupt workflows [2]. Interestingly, gang size was excluded by all models, suggesting that increases in crew size alone do not necessarily lead to higher productivity. This observation supports the notion that excessive crew sizes can introduce coordination difficulties and reduce efficiency due to overcrowding, reduced supervision, or diminishing marginal returns.

The feature-selection results highlight that safety management, communication effectiveness, site organization, and balanced working hours should be prioritized when aiming to improve CLP. These findings offer clear guidance for project managers, helping them focus on the factors that most strongly influence productivity and reduce common sources of disruption.

3.3. Stacking ensemble model development

This subsection reports stacked ensemble results under two evaluation settings: (i) a conventional 70/30 train/test split for model development comparison, and (ii) project-grouped cross-validation for leakage-free generalization assessment.

A stacking ensemble was developed to enhance CLP prediction by combining optimized base models. The framework includes two levels: base learners and a meta-learner. RF-PSO and RF-ABC were chosen as base learners due to their consistently strong performance across datasets. Stacking blends outputs from multiple base learners via a meta-learner. XGBoost was selected as the meta-learner for its generalization strength and performance on structured data. It was trained on the prediction outputs of the RF-PSO and RF-ABC models using the concrete, formwork, and bricklaying datasets. Hyperparameters, including learning rate, number of estimators, and tree depth, were tuned via grid search, with each dataset trained and evaluated independently. The performance in this subsection (Table 8 and Figures 10–11) is based on the conventional 70/30 train/test split used for baseline and optimized RF models. Project-grouped cross-validation results are reported separately in Tables 9 and 10 to assess generalization under project-level separation.

Under the conventional 70/30 evaluation setting, the stacked model outperformed individual models across all datasets, achieving RMSE = 0.0074 and R^2 = 0.9953 on concrete, while on formwork, it maintained an OI of 0.9168. Even on the more variable bricklaying dataset, the model achieved RMSE = 1.3207 and the highest R^2 (0.9160) and OI (0.9287).

The stacked ensemble model shows close alignment with the 1:1 reference line across all three activity categories, indicating high predictive accuracy and minimal systematic bias (Figure 10). Dispersion is lowest for concrete and formwork, while bricklaying shows slightly wider variability.

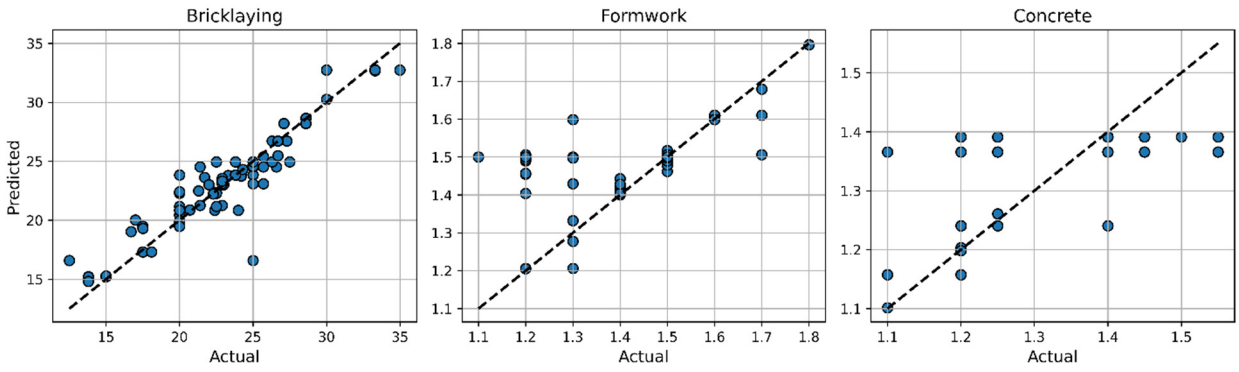

Figure 10. Actual vs. predicted labor productivity for the stacked ensemble model.

This visual diagnostic evaluation complements the numerical performance metrics and provides additional evidence of the model's robustness.

Table 8. Comparison of the RF-PSO and RF-ABC models and the ensemble model across selected metrics.

Dataset	Model	MSE	MAPE (%)	Overall Index
Concrete	Stacked ensemble	0.00005	0.20390	0.98940
	RF-PSO	0.00020	0.48801	0.97200
	RF-ABC	0.00020	0.45600	0.97200
Formwork	Stacked ensemble	0.00210	1.70600	0.916810
	RF-PSO	0.00401	2.54911	0.860102
	RF-ABC	0.00400	2.66300	0.85600
Bricklaying	Stacked Ensemble	1.74431	3.61260	0.928700
	RF-PSO	4.20501	6.32700	0.853003
	RF-ABC	4.23100	6.35011	0.852020

Under the conventional 70/30 train/test split, the ensemble model outperformed the individual models across all datasets, confirming the added value of stacking (Table 8). Although RF-PSO and RF-ABC each delivered strong performance on their own, their combined strengths resulted in a more accurate and stable prediction system. These findings align with those of Lu et al. [47], who showed that stacking ensemble models significantly outperform individual machine learning models in complex prediction tasks. Similarly, Karatas and Budak [27] demonstrated the effectiveness of stacking meta-ensemble models in CLP construction labor productivity prediction, where their stacking model outperformed both standalone and traditional ensemble learners.

Figure 11 illustrates the ensemble's advantage in lowering error and improving stability across datasets under the conventional evaluation setting, while later project-grouped cross-validation results further assess its generalization behavior.

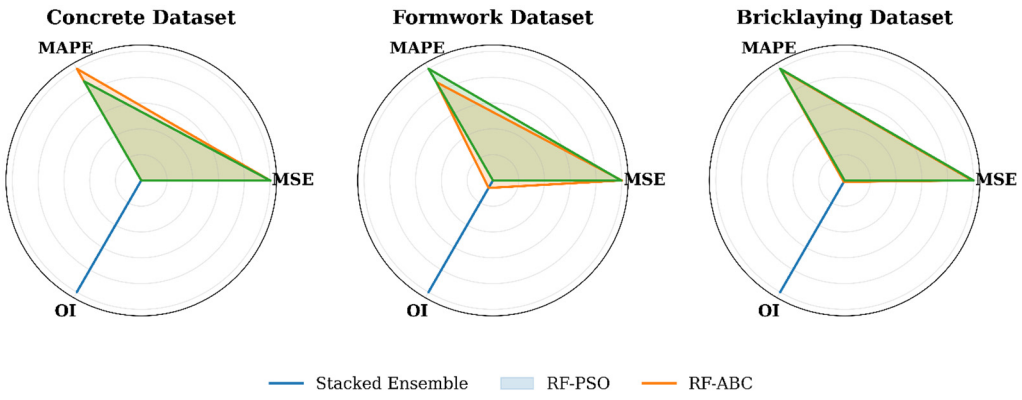

Figure 11. Radar chart comparison of the five RF-based metaheuristic models and the ensemble model.

To prevent data leakage across projects, the final stacked model was evaluated using grouped cross-validation, where all records belonging to the same project were kept within the same fold. This ensured that no project-level information appeared in both the training and testing subsets. Tables 9 and 10 report performance under project-grouped cross-validation and represent the primary evidence of generalization performance.

Table 9. Grouped cross-validation performance of the final stacked model.

Dataset	RMSE (mean ± SD)	R^2 (mean ± SD)
Concrete	0.0132 ± 0.0033	0.9824 ± 0.0082
Formwork	0.0615 ± 0.0096	0.8087 ± 0.0551
Bricklaying	1.7284 ± 0.1363	0.8367 ± 0.0159

Across the three datasets, the stacked model maintained consistently competitive predictive performance even when entire projects were withheld during training, with particularly robust behavior under the higher variability observed in the bricklaying activity. The RMSE values correspond closely to the underlying productivity scales; bricklaying exhibits a larger RMSE due to its wider productivity range (12.5–35), whereas the more constrained ranges in formwork (1.10–1.80) and concrete (1.10–1.55) naturally yield smaller RMSE values. The corresponding R^2 values indicate that the model captures a substantial proportion of productivity variance in all trades, with particularly strong performance observed for concrete ($R^2 \approx 0.98$).

Table 10. Performance comparison of RF-based models under project-grouped cross-validation.

Dataset	Model	RMSE (mean ± SD)	R^2 (mean ± SD)
Concrete	RF	0.0151 ± 0.0028	0.9777 ± 0.0079
	RF-PSO	0.0125 ± 0.0030	0.9845 ± 0.0076
	RF-ABC	0.0126 ± 0.0029	0.9843 ± 0.0075
	Stacked	0.0132 ± 0.0033	0.9824 ± 0.0082
Formwork	RF	0.0688 ± 0.0087	0.7593 ± 0.0680
	RF-PSO	0.0613 ± 0.0093	0.8094 ± 0.0538
	RF-ABC	0.0614 ± 0.0097	0.8086 ± 0.0561
	Stacked	0.0615 ± 0.0096	0.8087 ± 0.0551
Bricklaying	RF	1.9499 ± 0.2422	0.7935 ± 0.0246
	RF-PSO	1.7933 ± 0.1775	0.8253 ± 0.0097
	RF-ABC	1.8022 ± 0.1847	0.8237 ± 0.0094
	Stacked	1.7284 ± 0.1363	0.8367 ± 0.0159

Table 10 presents a direct comparison between the standalone RF models and the proposed stacked ensemble under an identical project-grouped cross-validation protocol. The RF-PSO and RF-ABC models achieve notable performance gains over the RF baseline across all three activity types, confirming the effectiveness of metaheuristic optimization in improving generalization. While the stacked ensemble does not uniformly outperform both optimized RF models in every dataset, it consistently delivers the most stable and competitive performance, particularly for the highly variable bricklaying activity. These results indicate that the stacked model's performance gains are not due to differences in validation procedures but rather stem from the complementary learning behavior of the integrated optimized models.

To further examine generalization behavior, a learning-curve analysis was conducted for each dataset, training the stacked model on gradually increasing portions of the data and monitoring both training and validation RMSE to assess convergence and stability.

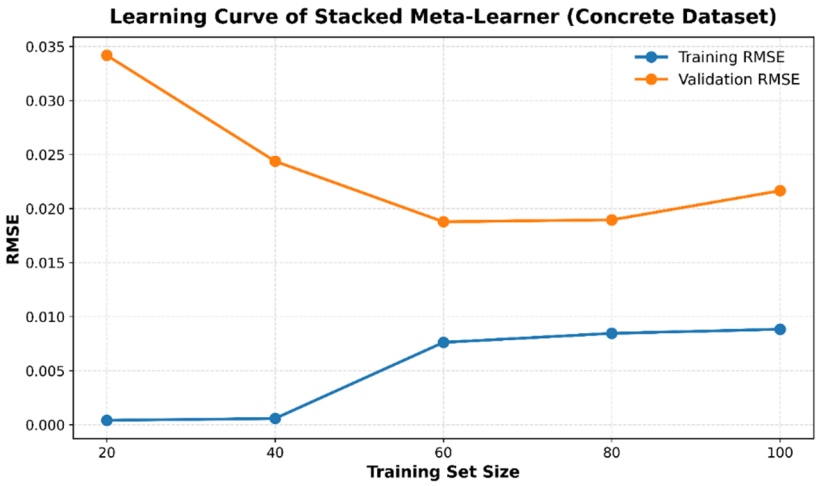

Figure 12. Learning curve of the stacked ensemble model on the concrete dataset.

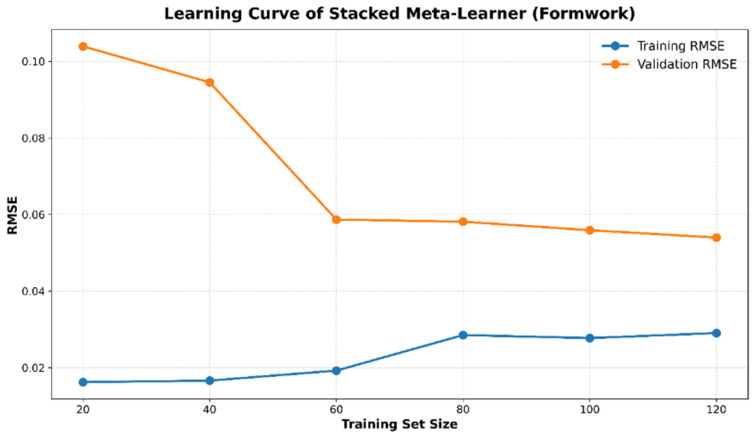

Figure 13. Learning curve of the stacked ensemble model on the formwork dataset.

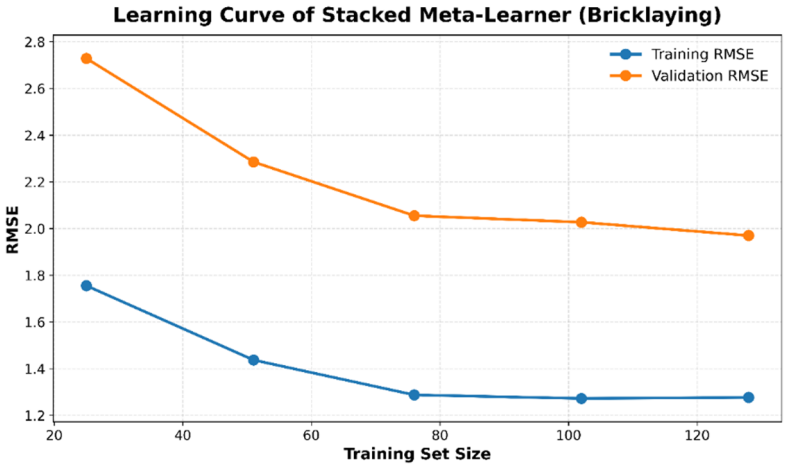

Figure 14. Learning curve of the stacked ensemble model on the bricklaying dataset.

As shown in Figures 12–14, the validation RMSE decreases and stabilizes as the training size increases, gradually converging toward the training RMSE. The absence of widening gaps between the curves confirms that the ensemble model generalizes well and does not rely on memorization, even for the more variable bricklaying dataset. This supports the robustness of the stacked approach and reinforces the reliability of its performance improvements over the individual RF-based models. Importantly, re-evaluation of the RF baseline and optimized RF variants using project-grouped cross-validation confirms that the observed performance gains of the proposed ensemble are robust and not influenced by data leakage or validation bias.

4. Conclusion

This study developed a stacked ensemble framework for construction labor productivity prediction, integrating two metaheuristic-optimized Random Forest models (RF-PSO and RF-ABC) with an XGBoost meta-learner. This architecture represents a key methodological innovation, combining optimization-enhanced base learners within a stacking structure to address limitations of previous single-model and non-integrated approaches.

In doing so, the study advances prior CLP prediction research in three concrete ways: (i) it applies metaheuristic optimization more comprehensively than earlier ML-based CLP studies, which rarely optimize feature subsets and hyperparameters jointly; (ii) it uses a relatively large and multi-project dataset, supporting model behavior across diverse site conditions; and (iii) it evaluates generalization using project-grouped cross-validation, providing a more realistic assessment of predictive performance on previously unseen projects.

Using data from 52 construction projects covering concrete, formwork, and bricklaying tasks, the proposed model demonstrated competitive and more stable performance, particularly for high-variance activities, relative to individual optimized models, consistently outperforming four additional baselines (MLR, KNN, SVMR, ANN).

The feature-importance results further revealed a small but influential set of predictors, accident frequency, weather conditions, communication quality, working space, and working hours, which strongly govern labor productivity across activities. These insights offer practical value by helping practitioners focus monitoring efforts on the most impactful variables, especially in data-scarce environments. The model's predictive capability can support more informed scheduling decisions, resource allocation, and proactive mitigation of productivity-disrupting factors.

While metaheuristic optimization introduces additional computation during training, this does not hinder practical application, as CLP models are used for strategic planning rather than real-time prediction. In this context, enhanced reliability justifies the added training effort. Future research may investigate lighter optimization strategies or cloud-based training solutions to improve scalability for organizations with limited computational capacity.

Overall, this study shows that integrating machine learning, metaheuristic optimization, and stacked ensemble learning provides a robust and generalizable approach for CLP prediction. With appropriate adaptation, the proposed framework can serve as a practical decision-support tool for improving productivity management in both developing and established construction markets.

Author contributions

Abdirisak Mohamed Abdillahi: Conceptualization, Data curation, Methodology, Software, Formal analysis, Investigation, Visualization, Writing – original draft, Writing – review & editing.

Savaş Bayram: Conceptualization, Methodology, Supervision, Validation, Writing – review & editing.

Funding

The authors declare that this research did not receive any specific grant from funding agencies in the public, commercial, or not-for-profit sectors.

Acknowledgments

The authors would like to thank the participating construction firms, site supervisors, and field data collectors for their support during data collection.

Conflict of interest

All authors declare no conflicts of interest in this paper.

Data Availability Statement

The data may be made available by the corresponding author upon reasonable request.

Use of Generative-AI tools declaration

The authors declare that no generative-AI tools were used in the preparation of this manuscript.

References

1. D. Przywara, A. Rak, Analysis construction industry on the basis of price trends of labor cost, *MATEC Web Conf.,* **146** (2018), 04005. https://doi.org/10.1051/matecconf/201817404005
2. A. M. Abdillahi, A. Kazaz, Evaluating Factors Effecting Building Construction Labor Productivity in Somaliland, *J. Constr. Eng. Technol. Manag.,* **11** (2021), 19–28.
3. O. Moselhi, Z. Khan, Significance ranking of parameters impacting construction labour productivity, *Constr. Innov.,* **12** (2012), 272–296. https://doi.org/10.1108/14714171211244541
4. T. Mahfouz, A productivity decision support system for construction projects through machine learning (ML), *Proc. CIB W78,* 2012
5. K. P. Kisi, N. Mani, E. M. Rojas, E. Terence Foster, Estimation of Optimal Productivity in Labor-Intensive Construction Operations: Advanced Study, *J. Constr. Eng. Manage.,* **144** (2018), 04018097. https://doi.org/10.1061/(ASCE)CO.1943-7862.0001551
6. Y. T. Lim, W. Yi, H. Wang, Application of Machine Learning in Construction Productivity at Activity Level: A Critical Review, *Appl. Sci.,* **14** (2024), 10605. https://doi.org/10.3390/app142210605

7. L. Florez-Perez, Z. Song, J. C. Cortissoz, Predicting construction productivity with machine learning approaches, *Proc. Int. Symp. Autom. Robot. Constr.*, 2022. https://doi.org/10.22260/ISARC2022/0017
8. P. Joshi, S. K. Shrestha, Analysis of Labor Productivity during concreting operation in building construction of Kathmandu Valley, *J. Adv. Res. Constr. Urban Archit.,* **4** (2019), 1–7.
9. M. H. Momade, S. Shahid, M. R. bin Hainin, et al., Modelling labour productivity using SVM and RF: a comparative study on classifiers performance, *Int. J. Constr. Manag*., **22** (2022), 1924–1934. https://doi.org/10.1080/15623599.2020.1744799
10. S. Golnaraghi, Z. Zangenehmadar, O. Moselhi, S. Alkass, Application of Artificial Neural Network(s) in Predicting Formwork Labour Productivity, *Adv. Civ. Eng*., **2019** (2019). https://doi.org/10.1155/2019/5972620
11. P. Goodarzizad, E. Mohammadi Golafshani, M. Arashpour, Predicting the construction labour productivity using artificial neural network and grasshopper optimisation algorithm, *Int. J. Constr. Manag.,* **23** (2023), 763–779. https://doi.org/10.1080/15623599.2021.1927363
12. S. Ebrahimi, Developing hybrid artificial intelligence model for construction labour productivity prediction and optimization, MSc Thesis, University of Alberta, Canada, 2021.
13. S. Ebrahimi, A. R. Fayek, V. Sumati, Hybrid artificial intelligence hfs-rf-pso model for construction labor productivity prediction and optimization, *Algorithms*, **14** (2021). https://doi.org/10.3390/a14070214
14. B. P. Kaur, H. Singh, R. Hans, S. K. Sharma, C. Sharma, M. M. Hassan, A Genetic algorithm aided hyper parameter optimization based ensemble model for respiratory disease prediction with Explainable AI, *PLoS One*, **19** (2024), e0308015. https://doi.org/10.1371/journal.pone.0308015
15. G. Feng, Feature selection algorithm based on optimized genetic algorithm and the application in high-dimensional data processing, *PLoS One*, **19** (2024), e0303088. https://doi.org/10.1371/journal.pone.0303088
16. D. Yilmaz Eroglu, U. Akcan, An Adapted Ant Colony Optimization for Feature Selection, *Appl. Artif. Intell.,* **38** (2024), 2335098. https://doi.org/10.1080/08839514.2024.2335098
17. E. Saraç, S. A. Özel, An Ant Colony Optimization Based Feature Selection for Web Page Classification, *Sci. World J.,* **2014** (2014), 1–16. https://doi.org/10.1155/2014/649260
18. H. Abubakar, S. Boukari, A. Y. Gital, F. U. Zambuk, A hyper-parameter tuned Random Forest algorithm-based on Artificial Bee Colony for improving accuracy, precision and interpretability of crime prediction, *Dutse J. Pure Appl. Sci.,* **10** (2024), 371–381. https://doi.org/10.4314/dujopas.v10i4a.34
19. M. Kaya Keles, U. Kilic, A. E. Keles, Proposed artificial bee colony algorithm as feature selector to predict the leadership perception of site managers, *Comput. J.,* **64** (2021), 408–417. https://doi.org/10.1093/comjnl/bxaa163
20. A. Brodzicki, M. Piekarski, J. Jaworek-Korjakowska, The whale optimization algorithm approach for deep neural networks, *Sensors*, **21** (2021), 8003. https://doi.org/10.3390/s21238003
21. M. A. Kahya, S. A. Altamir, Z. Y. Algamal, Improving whale optimization algorithm for feature selection with a time-varying transfer function, *Numer. Algebra Control Optim*., **11** (2020), 87–98. https://doi.org/10.3934/naco.2020017

22. A. T. Maadapoosi, V. Balamurugan, V. Vedanarayanan, S. A. Nisha, R. Narmadha, Grey Wolf Optimization-Based Artificial Neural Network in the Development of an Automated Heart Disease Prediction Model, *Int. J. Fuzzy Log. Intell. Syst.*, **24** (2024), 231–241. http://doi.org/10.5391/IJFIS.2024.24.3.231
23. Y. C. Kuyu, N. Ozekmekci, Grey wolf optimizer to the hyperparameters optimization of convolutional neural network with several activation functions, 2022 *Int. Symp. Multidiscip. Stud. Innov. Technol. (ISMSIT), IEEE,* 2022, 13–17.
24. Z. Chen, W. Fan, Freeway Travel Time Prediction Based on Ensemble Learning Approaches, *Int. Conf. Transp. Dev. 2023, Austin, Texas, American Society of Civil Engineers,* 2023, 410–423. https://doi.org/10.1061/9780784484876.036
25. B. Ozturk, A. Kodsy, M. Iskander, Forecasting the Bearing Capacity of Open-Ended Pipe Piles Using Machine Learning Ensemble Methods, IFCEE 2024, Dallas, Texas, *American Society of Civil Engineers*, 2024, 146–156. https://doi.org/10.1061/9780784485408.016
26. U. Park, Y. Kang, H. Lee, S. Yun, A stacking heterogeneous ensemble learning method for the prediction of building construction project costs, *Appl. Sci.,* **12** (2022), 9729. https://doi.org/10.3390/app12199729
27. I. Karatas, A. Budak, Development and comparative of a new meta-ensemble machine learning model in predicting construction labor productivity, *Eng. Constr. Archit. Manag.,* **31** (2024), 1123–1144. https://doi.org/10.1108/ECAM-08-2021-0692
28. N. Lawaju, N. Parajuli, S. K. Shrestha, Analysis of Labor Productivity of Brick Masonry Work in Building Construction in Kathmandu Valley, *J. Adv. Coll. Engin. Mgt.,* **6** (2021), 159–175. https://doi.org/10.3126/jacem.v6i0.38356
29. F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirio, O. Grisel, et al., Scikit-learn: Machine learning in Python, *J. Mach. Learn. Res.,* **12** (2011), 2825–2830.
30. S. Shoar, N. Chileshe, J. D. Edwards, Machine learning-aided engineering services' cost overruns prediction in high-rise residential building projects: Application of random forest regression, *J. Build. Eng.,* **50** (2022), 104102. https://doi.org/10.1016/j.jobe.2022.104102
31. M. Čeh, M. Kilibarda, A. Lisec, B. Bajat, Estimating the performance of random forest versus multiple regression for predicting prices of the apartments, *ISPRS Int. J. Geo-Inf.,* **7** (2018), 168. https://doi.org/10.3390/ijgi7050168
32. H. Sun, D. Gui, B. Yan, Y. Liu, W. Liao, Y. Zhu, et al., Assessing the potential of random forest method for estimating solar radiation using air pollution index, *Energy Convers. Manag.,* **119** (2016), 121–129. https://doi.org/10.1016/j.enconman.2016.04.051
33. T. O. Hodson, Root mean square error (RMSE) or mean absolute error (MAE): When to use them or not, *Geosci. Model Dev. Discuss.,* **2022** (2022), 1–10. https://doi.org/10.5194/gmd-15-5481-2022
34. K. Deb, Multiobjective optimization using evolutionary algorithms, New York: Wiley, 2001.
35. M. Dorigo, V. Maniezzo, A. Colorni, Ant system: optimization by a colony of cooperating agents, *IEEE Trans. Syst. Man Cybern. B Cybern.,* **26** (1996), 29–41. https://doi.org/10.1109/3477.484436
36. S. T. Ng, Y. Zhang, Optimizing Construction Time and Cost Using Ant Colony Optimization Approach, *J. Constr. Eng. Manage.,* **134** (2008), 721–728. https://doi.org/10.1061/(ASCE)0733-9364(2008)134:9(721)

37. D. Karaboga, An idea based on honey bee swarm for numerical optimization, Technical Report TR06, Computer Engineering Department, Engineering Faculty, Erciyes University, Kayseri, Türkiye, 2005.
38. M. Yahya, M. P. Saka, Construction site layout planning using multi-objective artificial bee colony algorithm with Lévy flights, *Autom. Constr.,* **38** (2014), 14–29. https://doi.org/10.1016/j.autcon.2013.11.001
39. D. H. Tran, M. Y. Cheng, M. T. Cao, Hybrid multiple objective artificial bee colony with differential evolution for the time–cost–quality tradeoff problem, *Knowl.-Based Syst.,* **74** (2015), 176–186. http://dx.doi.org/10.1016/j.knosys.2014.11.018
40. A. Sharma, A. Sharma, S. Choudhary, R. K. Pachauri, A. Shrivastava, D. Kumar, A review on artificial bee colony and it's engineering applications, *J. Crit. Rev.,* **7** (2020), 4097–4107.
41. S. Mirjalili, S. M. Mirjalili, A. Lewis, Grey wolf optimizer, *Adv. Eng. Softw*., **69** (2014), 46–61. https://doi.org/10.1016/j.advengsoft.2013.12.007
42. E. Dada, S. Joseph, D. Oyewola, A. A. Fadele, H. Chiroma, S. I. M. Abdulhamid, Application of grey wolf optimization algorithm: recent trends, issues, and possible horizons, *Gazi Univ. J. Sci.,* **35** (2022), 485–504. https://doi.org/10.35378/gujs.820885
43. S. Katoch, S. S. Chauhan, V. Kumar, A review on genetic algorithm: past, present, and future, Multimed. *Tools Appl*., **80** (2021), 8091–8126. https://doi.org/10.1007/s11042-020-10139-6
44. S. N. Sivanandam, S. N. Deepa, Genetic Algorithms, in: Introduction to Genetic Algorithms, Berlin, Heidelberg: Springer Berlin Heidelberg, 2008, 15–37. https://doi.org/10.1007/978-3-540-73190-0_2
45. J. Kennedy, R. Eberhart, Particle swarm optimization, *Proc. ICNN'95-Int. Conf. Neural Netw., IEEE,* 1995, 1942–1948. https://doi.org/10.1109/ICNN.1995.488968
46. H. Yiğit, S. Ürgün, S. Mirjalili, Comparison of recent metaheuristic optimization algorithms to solve the SHE optimization problem in MLI, *Neural Comput. Appl.,* **35** (2023), 7369–7388. https://doi.org/10.1007/s00521-022-07980-1
47. M. Lu, Q. Hou, S. Qin, L. Zhou, D. Hua, X. Wang, et al., A stacking ensemble model of various machine learning models for daily runoff forecasting, *Water*, **15** (2023), 1265. https://doi.org/10.3390/w15071265
48. D. R. I. M. Setiadi, K. Nugroho, A. R. Muslikh, S. W. Iriananda, A. A. Ojugo, Integrating SMOTE-Tomek and Fusion Learning with XGBoost Meta-Learner for Robust Diabetes Recognition, *J. Future Artif. Intell. Technol.,* **1** (2024), 23–38. https://doi.org/10.62411/faith.2024-11

AIMS Press