Automation in Construction 152 (2023) 104891

Contents lists available at ScienceDirect

Automation in Construction

journal homepage: www.elsevier.com/locate/autcon

Machine learning for construction crew productivity prediction using daily work reports

Amir Sadatnya [a], Naimeh Sadeghi [b], Sina Sabzekar [c], Mohammad Khanjani [c], Ala Nekouvaght Tak [d], Hosein Taghaddos [a,*]

[a] School of Civil Engineering, University of Tehran, Tehran, Iran
[b] Faculty of Civil Engineering, K. N. Toosi University of Technology, Tehran, Iran
[c] Department of Civil Engineering, Sharif University of Technology, Tehran, Iran
[d] Sonny Astani Dept. of Civil and Environmental Engineering, University of Southern California, Los Angeles, CA, USA

ARTICLE INFO

Keywords:
Productivity estimation
Machine learning
Construction productivity
Ensemble methods
Feature engineering
Daily work reports
Neural network
Data-driven prediction models
Data mining

ABSTRACT

Construction productivity estimation lacks a comprehensive, standard, and task-type-independent framework to generate and serialize Machine Learning (ML) models. This research aims to develop an ML management framework for estimating the work crew productivity (crew outputs over their working hours) by addressing various operation and project types. The framework takes advantage of historical data, including information regarding operations' progress, weather conditions, the number of resources, and their composition in a work crew. Daily work reports are used as a principal source of historical data. Various hyperparameters-tuned ML algorithms are adopted and ranked based on their computational complexity and prediction accuracy. The generated productivity prediction models have the flexibility to be reused for the effective planning of various construction projects. Applying the proposed framework to a case study of nine disciplines provided estimation models with high accuracies. This study also discusses the theoretical and practical implications of the presented model development procedure.

1. Introduction

Construction projects entail a variety of resources, including labor, equipment, and materials, that demand planning for efficient operations [1]. Significant effort has been made to develop models and methods to enhance project planning in the construction industry to avoid time delays and cost overruns [2]. To achieve the best in planning, we need to provide an accurate estimate of the time needed to finish each operation in the project and assess construction productivity.

Work Crew Productivity (WCP), which is herein defined as crew-produced outputs over the crew working hours, is integral to effectively managing construction operations, given its direct impact on the overall cost and duration of projects [3–5]. Since the WCP's unit is work volume per time, the productivity for each construction operation is based on context and specified differently. For instance, the productivity for earthmoving operations is the volume of soil moved per unit of time (e.g., m^3/h), while this definition for wall painting activity is the amount of painted surface per unit of time (e.g., m^2/h). Furthermore, reliable productivity forecasting is critical to assisting construction managers in making early decisions to avoid cost overruns and project delays [6]. Previous research has shown that productivity is influenced by a broad range of factors. To improve productivity, it is vital to investigate the elements influencing it, whether favorably or adversely [6,7].

Several studies have been conducted to investigate the factors affecting productivity [8–11]. Due to the varying environmental characteristics and management conditions of each project, construction productivity rates are different by project [12]. Some of the major factors influencing productivity are [13–17]: Technology, labor, management, external factors, design complexity, construction management, crew supervision, project type, work scope, project complexity, environmental factors, including weather conditions, temperature, and air humidity and their interaction with other factors.

The conventional method for estimating construction WCP is based on prior experience, expert opinion, engineering judgment, and production rate charts [18,19]. The primary shortcoming of this approach is that it relies on inaccurate judgments and insufficient information [20].

* Corresponding author.
E-mail addresses: amirsadatnya@ut.ac.ir (A. Sadatnya), sadeghi@kntu.ac.ir (N. Sadeghi), sina.sabzekar@sharif.edu (S. Sabzekar), mohammad.khanjani@sharif.edu (M. Khanjani), nekouvag@usc.edu (A.N. Tak), htaghaddos@ut.ac.ir (H. Taghaddos).

https://doi.org/10.1016/j.autcon.2023.104891
Received 13 December 2022; Received in revised form 16 March 2023; Accepted 19 April 2023
Available online 5 May 2023
0926-5805/© 2023 Elsevier B.V. All rights reserved.

Using methods based on historical data is another common approach to WCP estimation [13]. These methods involve gathering and analyzing project records with details of production rates and worksite circumstances [19]. Daily Work Reports (DWRs) have been used as a common source of historical data. DWRs usually contain records of environmental conditions such as weather, ground conditions, rainfalls, job-site conditions, labor and equipment work hours, production rates, and facilities costs (e.g., Fig. 1) [21]. DWRs are frequently utilized in practice to determine the correlations between production rates and worksite conditions [20,22]. Therefore, there is an ideal opportunity for researchers to acquire a set of rich, inexpensive, and multifaceted (e.g., the number and type of resources and weather conditions) information on the construction projects' progress without spending extra time and money. The only effort required is to extract and gather DWRs' raw data into a standard and predefined format to be useable for training machine learning models. Since most of the data in DWRs describes the project conditions (independent variables), which directly affect the daily progress of activities and productivity (dependent variables), DWRs can be considered as a suitable source for estimating the WCP. However, they have not been used appropriately to extract data, and there is still a lack of a developed framework for crew productivity prediction in the literature.

Developing modern statistical or Machine Learning (ML) models for WCP estimation using the obtained data from the DWRs is the next plausible step in WCP estimation. The term “statistical methods” refers to a set of broad concepts and procedures, routinely employed in data gathering, analysis, and interpretation. Traditional linear statistical approaches might be restricted in capturing the complexities of complicated modeling situations with high dimensional and high-volume data, such as the case of WCP modeling, decreasing their prediction potential [23]. ML, on the other hand, presents a viable alternative to traditional productivity estimating approaches for modeling complicated linear or non-linear relationships involving large amounts of high-dimensional data [23].

ML is a branch of Artificial Intelligence (AI) that focuses on using data and algorithms to empower systems to learn from data and recognize patterns automatically [24]. ML is comprised of a variety of algorithms and models, including Artificial Neural Network (ANN), Fuzzy Logic (FL), Support Vector Machine (SVM), and Random Forest (RF), which all have been developed over the years for modeling non-linear relationships [23,25,26]. These ML models have been proven to be quite effective in WCP estimation in a variety of construction projects [25]. Since the accuracy of WCP estimation is critical for planning construction projects, improving the performance of AI-based models is always a priority [6]. Various ML methods must be developed and evaluated to gain the highest accuracy in WCP estimation.

This research aims to develop a framework for estimating the WCP (i.e., the productivity of laborers and equipment's composition) for various operations and project types. The study aims to facilitate the framework's applicability for project managers with minimum specialized ML knowledge. In this regard, the paper intends to develop the framework by minimizing interactions between users and technical topics of ML. Another study objective is to use productivity estimation models in project planning and scheduling. Thus, the framework stores and maintains productivity estimation models along with other properties required to enable future use.

The rest of the paper is organized as follows. Section 2 reviews related studies concentrating on construction productivity prediction using ML methods. Section 3 clarifies existing gaps and expresses the objectives of the paper. The proposed methodology is explained in Section 4. The framework's implementation environment is described in Section 5. Section 6 implements the framework in a linear project with nine main activities. Section 7 discusses the framework based on the case study and the obtained results. The paper concludes with Section 8, where the presented system is summarized, the weaknesses are expressed, and extensible sections of the current framework are proposed.

2. Literature review

Researchers have studied the impact of improved WCP estimation and emphasized the criticality of striving for better accuracies, particularly in labor-intensive construction tasks [8,27,28]. The current industry practice still lacks standard and comprehensive procedures for collecting, reporting, tracking, and analyzing historical data together with well-defined productivity measures and benchmarks [13,20,29–32]. For effective and reliable productivity analysis, companies must gather data at the activity or operation level rather than rely on a summary level [7,13]. The repetitive and cyclic nature of many activities in construction projects increases the impact of improved productivity estimation based on historical data gathered from previous projects [33,34].

Based on the identified factors, Various modeling approaches have been introduced to estimate productivity. These methods are primarily based on Artificial Intelligence (AI), including regression analysis, expert systems, and ML [35,36]. Implementing search algorithms, such

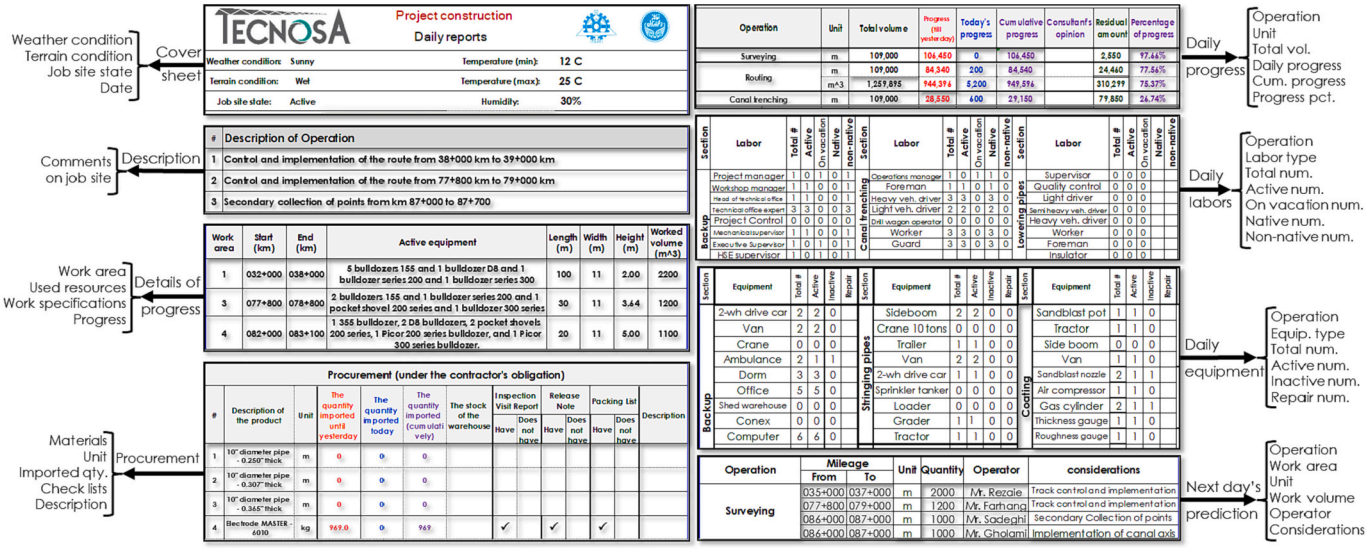

Fig. 1. Different sections of a sample DWR and their information.

as ant colony, artificial bee colony, particle swarm optimization, and self-organizing maps, have also been discussed for the productivity prediction problem [37]. Several researchers have proposed using computer simulation and modeling to estimate the performance of a system in a virtual environment [34]. Khanzadi et al. [38] proposed a system dynamic simulation approach to predict productivity in three types of construction activities. Fayek and Oduba [39] developed a fuzzy expert model to estimate labor productivity in pipe welding activities. Tsehayae and Fayek [40] proposed a context-specific fuzzy inference system–based construction labor productivity model. Malara et al. [17] also adopted a fuzzy logic approach for the mathematical description of the influential factors and productivity modeling. Most recently, Kim et al. [20] used a predetermined motion time system, 3D models (e.g., biped models, building information modeling), and discrete-event simulation to develop a motion-based productivity estimation model. Researchers have also used regression models for productivity estimation purposes. For instance, Srinavin and Mohamed [41] proposed a polynomial regression model for painting, bricklaying, and excavation under different thermal environments. Zhao et al. [14] developed a multiple linear regression model for productivity prediction in hot and humid environments. In a recent effort, Khanh et al. [8] leveraged a binary logistic regression model to predict the productivity for the masonry work of brick walls.

Most past efforts relied on statistical methods to discover patterns and gain knowledge from the collected data. However, statistical methods cannot fully exploit the large sets of data in the construction industry [42]. Following the ubiquitous trend, the volume of data is growing unprecedentedly in the construction industry [43]. Data-driven approaches are increasingly implemented to model, predict, and optimize issues throughout the whole lifecycle of the actual complex project [6,44]. ML methods can facilitate finding generalizable predictive patterns and linear or non-linear relationships by drawing inferences from different historical data sources [23,45]. Several methods, such as SVM, ANN, and ensemble models, are often used and compared with each other in terms of accuracy [46]. SVM generates optimal hyperplanes in higher-dimensional space to identify a global solution. Inspired by biological neural networks, ANN imitates human learning processes with an interconnected neuron architecture. Ensemble learning methods (e.g., RF, adaptive boosting, and extreme gradient boosting) combine decisions from multiple weak models and attain the best result through voting.

Researchers in construction engineering and management have proposed many applications of ML methods for predicting productivity [7]. For instance, Ezeldin and Sharara [47] measured concreting productivity using ANN. Kassem et al. [48] proposed a deep neural network model for estimating excavators' productivity. In [49], a machine learning-based model was developed to predict the productivity of the prefabricated external insulation system. This method enables rapid analysis of construction productivity without using real data. Gurmu and Ongkowijoyo [35] developed a logistic regression model to predict construction labor productivity. This study investigated the correlation between productivity, company profiles, human resource management, and project characteristics. Al-Zwainy et al. [50] achieved around 90% accuracy in estimating the productivity of construction projects using ANNs. Mahfouz [51] used the SVM development model technique for productivity estimation of steel structures and found Naive Bayes (NB) model to be the most suited among the developed ones, achieving more than 70% prediction accuracy. Mady [52] developed an ANN model to predict the production rate for slab works and obtained the best accuracy through the traditional trial and error process. Al-Zwainy et al. [53] employed a linear regression technique to forecast the productivity of floors' marble finishing works and achieved a prediction accuracy of 90.6%. Kaya et al. [54] employed decision trees to classify the factors that affect the productivity of ceramic tiling activity based on the daily collected productivity values. The forecasted productivities were presented qualitatively in three classes of low, medium, and high. Heravi and Eslamdoost [55] applied ANN to predict labor productivity in concreting activity for industrial plant construction projects. Ok and Sinha [56] also employed ANN to estimate the daily productivity of earthmoving equipment. Mirahadi and Zayed [34] proposed a modified version of a neural network-driven fuzzy reasoning structure to enhance the accuracy of productivity estimation for construction operations. El-Ghohary et al. [7] employed feedforward back-propagation ANN using the hyperbolic tangent as the transfer function for formwork, steel fixing, and concrete-pouring activities. However, the human-dependent input data suffer from biases that introduce inaccuracies and random noises and limit the effectiveness of any such prediction models. More recently, Nasirzadeh et al. [57] proposed the application of ANN-based prediction intervals to achieve a more reliable prediction of labor productivity using historical data. Golnaraghi et al. [58] implemented several ANN-based prediction models to estimate the productivity of formwork activity. The features used in the study were in three classes of climate, crew, and project-related factors. Bai et al. [59] also compared various ML models (e.g., RF, k-Nearest Neighbors (KNN), and NB) to estimate the productivity of cutter suction dredger operation. Feature selection was a critical phase in this study to achieve a prediction accuracy of more than 90%.

Recently, several studies have focused on comparing different ML models and finding models with minimum prediction error. For example, Fu et al. [23] studied four ML models, including support vector regression, extreme gradient boosting, back-propagation neural network, and long short-term memory for cutter suction dredger productivity estimation, and achieved a determination coefficient R2 of over 80%. Momade et al. [25] proposed a data-driven method for the preparation of construction labor productivity models using influencing labor factors. The developed models were meant to prepare a reliable work schedule and budget estimation before and during the construction stages. Oral et al. [37] analyzed the prediction performance of KNN and generalized neural network when used for the dataset associated with three different operations (formwork, tiling, and masonry). Cheng et al. [6] introduced a hybridization of least square SVM, symbiotic organisms search, and a feature selection method to accurately forecast a construction project's productivity. This study tried to optimize the model by simultaneously determining hyperparameters and highly relevant construction productivity attributes. Lee et al. [60] used historical DWR data to calculate actual production rates to evaluate contractors' production performance which is rarely considered in current evaluation systems.

The performance of trained ML models mostly depends on input data, model type, and preprocessing procedure. Feature selection is a part of the preprocessing procedure that decreases the number of input features by selecting the most relevant ones to the target variable. Reducing the number of input variables decreases training models' computational costs and improves models' performance in some cases. Wrapper and filter methods are categorized as supervised feature selection techniques. Filter methods use statistical tests to evaluate the correlation of input variables and target variables and filter the ones with the lowest scores. The wrapper method creates multiple models, each with a subset of variables. These models are evaluated, and the best combination of features is selected to maximize the model performance [61]. The wrapper methods usually result in higher accuracy than filter methods. However, since wrappers search in the space of all possible combinations of features, they are considered computationally costly methods [62].

Various studies have been conducted to present feature selection methods. Cheng et al. [6] proposed an AI model called SOS-LSSVM-FS (i.e., symbiotic organisms search-least square support vector machine-feature selection), which integrates a dynamic feature selection technique. Yu et al. [63] proposed a hybrid feature selection method based on fuzzy information entropy. The authors searched for feature subsets using a Binary-Chaotic Multi-Objective Particle Swarm Optimization (B-CMOPSO) algorithm. The optimal subset is then selected based on the

simulation accuracy of machine learning methods on the feature subset. Bai et al. [59] proposed a feature selection method based on Lasso and maximal information coefficient redundancy recognition to determine a feature subset with high correlation and low redundancy. To take advantage of both feature selection methods, several studies have presented hybrid feature selection techniques. For instance, Lan et al. [64] and Hsu et al. [65] presented hybrid feature selection by combining filters and wrappers. The present study develops a hybrid approach to gain the privilege of both filters and wrappers in the studied problem (detailed in Sections 4.2.3 and 4.4.1). The hybrid approach determines the correlation between features and the target (i.e., productivity) using filter methods and investigates the opinion of project experts using a wrapper method. The following section discusses the existing knowledge gaps to clarify the problem studied in this paper and corroborate its contribution to the state-of-the-art construction productivity estimation.

3. Problem statement

AI and specifically ML offer a new capacity for various industries such as construction. Despite efforts to utilize ML in the productivity prediction problem, its applicability is yet limitedly explored. Most studies focused on a particular operation type. In fact, this field still lacks a comprehensive, standard, and task-type-independent framework to generate and serialize productivity estimation models using the historical information of past projects. The proposed models are either not replicable to all sorts of operations/projects or require significant time and effort to implement in a new operation/project. For instance, Ok and Sinha. [56] utilized an artificial neural network to estimate the dozer's productivity. Muqeem et al. [66] developed a model to predict the production rate for the installation of beam formworks. Mohammed and Tofan [67] used ML to estimate ceramic wall construction productivity. Mady [52] and Heravi and Eslamdoost [55] modeled labor productivity in concreting activity.

In most previous studies, data gathering, data preprocessing, and productivity estimation models' training phases are performed for a particular activity in a time-consuming and labor-intensive manner. High ML knowledge is also required for civil engineers to implement the previous methodologies in other activities. Few studies have used a prevalent and inexpensive source to achieve enhanced model performance through a continuous flow of input data. DWRs provide a continuous source of information to construction projects that include labor productivity-related data for various activities in construction projects. A practical perspective necessitates developing a comprehensive, standard, and reusable methodology for the industry, which is mostly overlooked in previous research. In most cases, researchers have focused on generating ML prediction models without considering the real-world application of the developed models to address project management issues (e.g., planning and scheduling).

The paper contributes to the body of knowledge by presenting a comprehensive semi-automated framework to generate productivity estimation models using ML. The presented framework considers several affecting factors (e.g., weather and terrain conditions, number of resources, and their composition in a work crew) from DWRs through a structured data extraction procedure. To respond the gaps and improve ML models performances, the proposed framework, endeavor to (1) use DWRs as a prevalent sources to extract underlying knowledge and hidden patterns regarding WCP (2) establish a process flow to generate productivity prediction models customizable for different operation/project type, (3) facilitate model development for civil engineers and users with limited specialized knowledge of ML, (4) maximize the prediction accuracy by employing various ML models and performing intensive model enhancement methods (e.g., hyperparameter tuning and bin boundaries determination), (5) enhance applicability by embedding dedicated modules within the framework to respond to various data types, data distributions, missing data, and outliers in data any condition and running the presented process flow, (6) consider the real-world implementation by serializing productivity prediction models and storing essential dependencies in a database, and (7) analyze affecting factors on productivity (i.e., features) and determine important features by presenting a hybrid feature selection process based on filter and wrapper method.

4. Methodology

In the paper, the estimation is based on historical data, including information regarding (1) operations' progress, (2) weather conditions, (3) the number of resources and their composition in a work crew, and (4) other project specifications that affect productivity. To discover underlying knowledge and hidden patterns regarding WCP in the data, DWRs are utilized. DWRs are a prevalent source in most construction projects to be used as historical data. Once the required data is extracted from the DWRs and compiled in a standard format, data is preprocessed to obtain refined information. Next, various hyperparameter-tuned ML models are developed to be evaluated and compared. The trained models are ranked based on their prediction accuracy and computational complexity on test data. To facilitate the process and minimize errors, the above steps are performed through a semi-automated manner. The following section provides an overview of the presented framework.

4.1. Overview

To describe the framework, the methodology is divided into several parts. First, the system's general flowchart is depicted and briefly discussed in this section (Fig. 2). Next, various phases of data preprocessing, model preparation, and model training are presented orderly.

Fig. 2 demonstrates the proposed framework's flowchart for estimating productivity. The presented framework includes three phases: (1) data extraction and preprocessing, (2) model preparation, and (3) model training. DWRs are utilized as the primary source of data gathering for the ML model development to increase the flexibility and scalability of the framework. Data extraction and preprocessing ascertain how to refine raw data from DWRs. Model properties such as model type, train, test, and cross-validation data are determined in the model preparation phase. Finally, training ML models, comparing, and storing results are embedded in the model training phase. In the proposed framework, various types of supervised ML models and data analysis techniques are employed to acquire a holistic view of ML models' performances and to balance the bias-variance tradeoffs (illustrated in Fig. 2). The following section describes the framework in more depth.

4.2. Data extraction and preprocessing

4.2.1. Data extraction

In the first step, using the collected DWRs, a dataset containing project activities' data points is built. Each data point describes the quantity of the factors affecting productivity and the daily progress rate as features and the target, respectively. The desired format for gathering the data is depicted in Fig. 3. As shown in Fig. 3, extracted features are generally dividable into five categories of calendar, weather conditions, labors, equipment, and terrain conditions, which are almost common among various project types.

The collected DWRs are provided in Excel format. Each Excel file contains several sheets regarding day-to-day project data, including activity progress, number and types of machinery, number and types of labors, and weather conditions such as temperature, precipitation, and humidity. Such information is almost common among all contractors. When DWRs are collected, the DWRs are categorized based on their form's shape, and then the proposed system utilizes a two-step algorithm to facilitate the data extraction. By executing the algorithm, parameters affecting productivity that are registered in DWRs are arranged semi-automatically in a unique format with their quantities, as presented

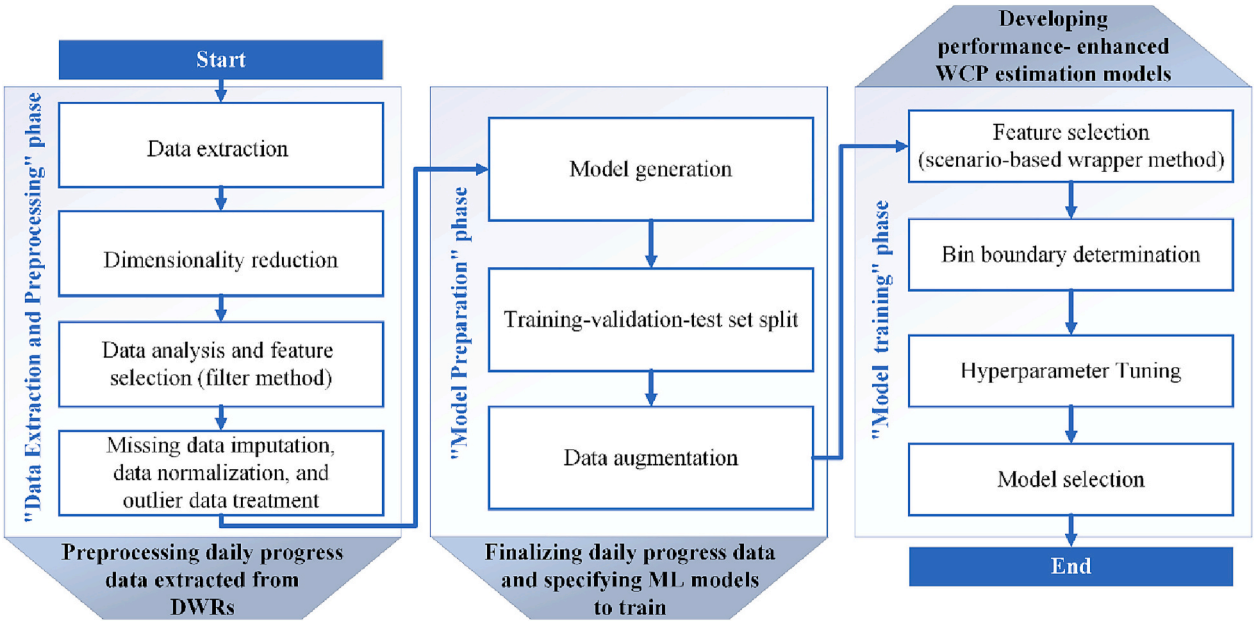

Fig. 2. The proposed framework's flowchart.

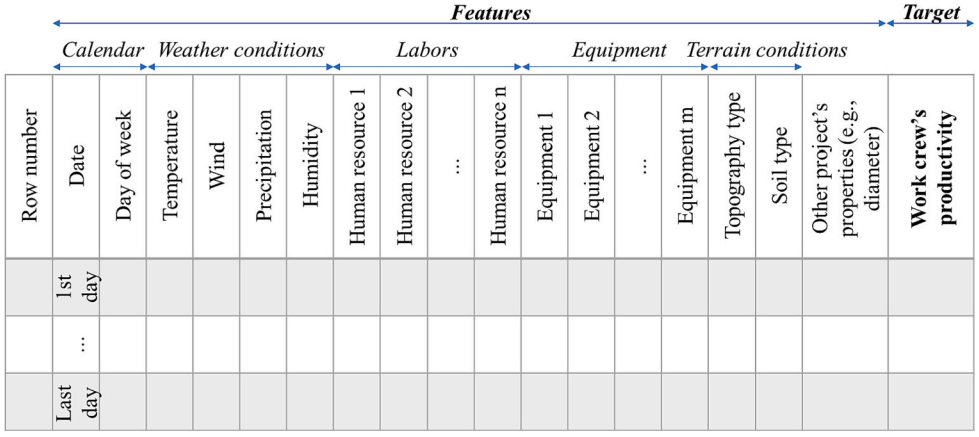

Fig. 3. Desired extracted data format from DWRs.

below. Generally, one can automatically extract any data from excel by specifying (1) sheet titles, (2) the columns/row address that includes the feature title, and (3) the columns/row address that contains the corresponding value.

4.2.1.1. Title extraction. To extract data from DWRs, it is necessary to specify the titles to extract the associated quantity in the next step. In this regard, the list of titles is stored by reviewing all DWRs automatically. Each section of the titles, including activity progress, labor, machinery resource, and weather condition, is extracted and saved separately. The saved files are examined, and the titles are modified if necessary.

4.2.1.2. Quantity extraction. After determining the desired titles, the algorithm's second step extracts the quantities corresponding to each title automatically. In this section, all DWRs are selected, and the quantity is searched for each specified title. By storing the titles' quantity in DWRs, the desired set of information is obtained in a standard format, where a data row in the output file represents one day of the project. Similarly, a column represents a title (feature or target). The described standard format is depicted in Fig. 3.

The presented data extraction flowchart with an illustrative example is depicted in Fig. 4, where the blue rectangles and arrows represent the "title extraction", and the green ones represent the "quantity extraction" step. In the example, by specifying (1) sheet "Equipment" as the target sheet, (2) columns "B-C", "H-I", "N-O", and "T-U" as the feature title columns, and (3) columns "E", "K", "Q", and "W" as the corresponding value columns, the framework automatically extracts these contents for each Excel file (equivalent to one working day). This process will be repeated automatically for every report in a category to finally gather all the required data in a single Excel file.

4.2.2. Dimensionality reduction

Some extracted features (generally machinery resources) are similar and have the same functionality, with the only difference in the equipment specifications (e.g., engine power and type). To increase the accuracy of ML models, the mentioned features/connatural resources (e.g., various types of dump trucks, bulldozers, side-booms, and loaders) are merged into a united feature/resource. It should be noted that composite features are defined according to the equipment number, power, and volume. The quantity of composite features is obtained by weighted summation of features' quantity. Other effective project parameters (e.

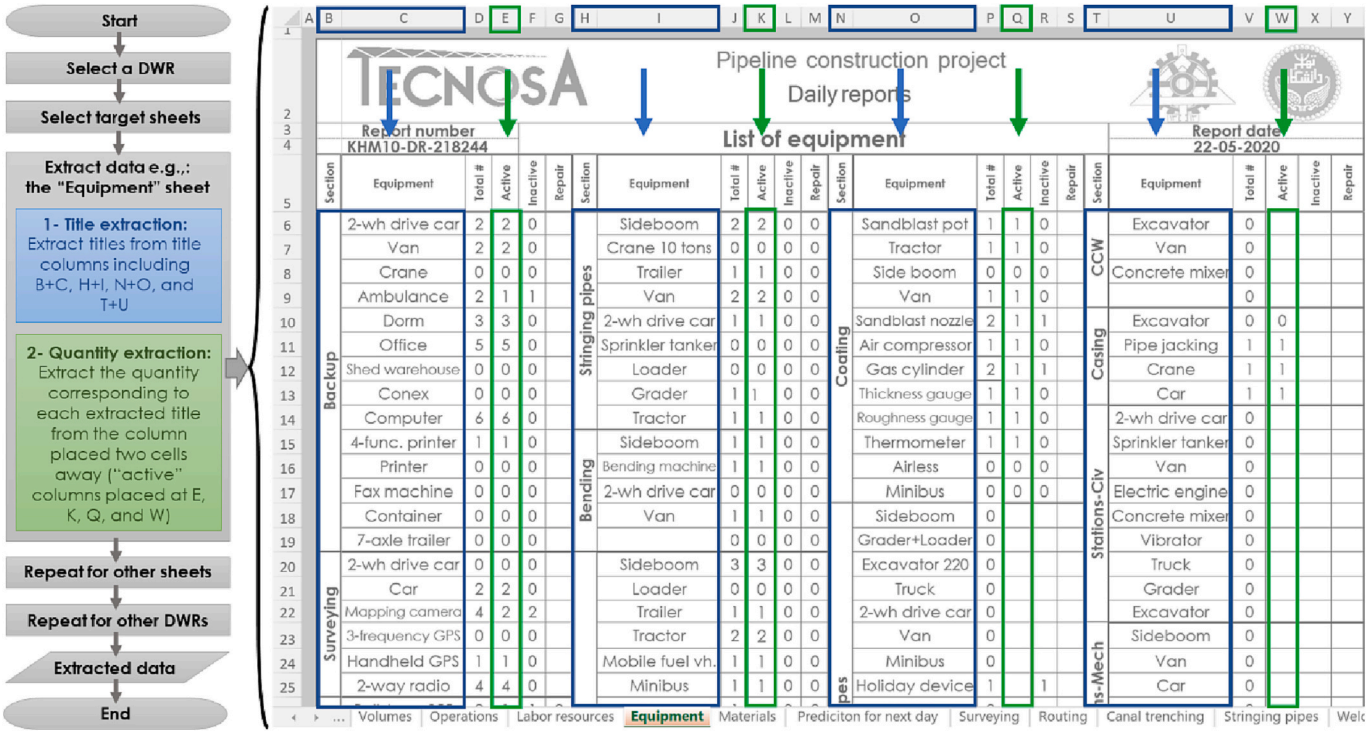

Fig. 4. The data extraction flowchart with a DWR's "Equipment" sheet.

g., slope and land type in roadway construction) on productivity are also added to the dataset.

4.2.3. *Feature selection (filter method)*

The paper proposes the feature selection in two parts. The first part is developed to eliminate irrelevant features using filter methods and identify candidate (to-be-verified) features. Candidate features here represent ones that are not correlated to productivity based on the extracted data, but on the other hand, the experts suggest keeping them according to the project's nature. Thus, the framework recommends investigating each candidate feature separately. Accordingly, the second part uses a wrapper method approach with different scenarios, each representing a combination of candidate features. By developing ML models based on the scenarios, the best combination of features is ascertained. In the first part of the proposed feature selection (i.e., filter method), various assessments are performed automatically on the data to discover embedded knowledge and relationships and identify insignificant features. The framework implements various correlation filter methods and univariate feature selection techniques such as the Pearson correlation coefficient, Spearman's rank correlation coefficient, Kendall's rank correlation coefficient, the Chi-Square test, the Analysis of Variance (ANOVA) F-value, and mutual information. Each method has been applied for a particular purpose (Table 1), and the framework maintains features correlated to productivity, at least by one measure.

Pearson measures the linear correlation between two sets of data. Pearson's correlation coefficient is denoted by "r" and takes a range of values from −1 to +1. A value of 0 is an indicator of no linear relationship between two datasets. Moving toward +1 and − 1 signifies a greater positive and negative linear relationship between the two datasets. Spearman's rank correlation coefficient assesses the relationship (linear or non-linear) based on the monotonic function. The coefficient is denoted by "ρ" and takes a range of values from −1 to +1. A value of 0 indicates no monotonic relationship between the two datasets. Values close to +1/−1 insinuate a greater monotonic increasing/decreasing relationship between the two datasets. Kendall rank correlation evaluates the ordinal association between two quantities. The correlation determines the strength of the relationship based on the concordance and discordance between the pairs (data points). Similar to Spearman's rank correlation coefficient, Kendall's rank correlation coefficient's domain is between −1 and 1, while 0 represents no correlation

Table 1
Filter methods used in the framework.

Test type	Application	Analysis type	Input variable (Numerical: N, Categorical: C)	Output variable (Numerical: N, Categorical: C)
Pearson product-moment (Pearson) correlation	Measuring the linear correlation between two sets of data	Parametric	N	N
Spearman's rank-order (Spearman) correlation	Measuring the strength and direction of association between two sets of data	Non-parametric	N	N
Kendall rank correlation	Measuring the ordinal association between two measured quantities.	Non-parametric	N/C	C/N
Chi-Square test	Measuring the independence of two events	Non-parametric	C	C
Analysis of variance (ANOVA) F-value	Measuring the equality of samples' distribution	Parametric	N/C	C/N
Mutual information	Measuring the reduction in uncertainty for one variable given a known value of the other variable.	–	N/C	C

between two quantities.

The Chi-square test determines the independence of two datasets, and the Chi-square value is the outcome of the test, which is calculated based on the difference between the observed value and the expected value. If two datasets are independent, expected values tend to the observed values, which decrease the Chi-square value. In the ANOVA f-test, ANOVA is a parametric statistical hypothesis test to determine whether the means of data samples have the same distribution, and the F-test is a class of statistical tests that measures the ratio between variance values (e.g., ANOVA). The resulting scores of the test specify the dependency of two datasets (features and the target herein); the higher the score, the greater the dependency between the feature and the target. Lastly, given a value of one variable, mutual information assesses the reduction in uncertainty for the other variable. Simply put, mutual information determines the amount of information that can be obtained from one random variable given another. Similar to the ANOVA f-test, the higher score indicates a stronger relationship.

After executing the analysis, each activity's features are divided into three branches. The first branch is the features that are important and maintained. The second branch is a set of features that are not required according to the analysis and project construction experts; thus, are omitted. The third branch is a set of features considered to-be-verified (i.e., candidate) according to the analysis or the opinion of project construction experts requiring further investigation. Thus, various scenarios are defined to achieve the best subset of features, where each scenario contains a combination of the first branch and candidate features.

4.2.4. Data imputation, normalization, and outlier treatment

Missing data in the DWRs must be identified to estimate their value. As a replacement for traditional imputation techniques, KNN is a widely applied algorithm to impute missing values. The KNN method identifies the neighboring points based on their distance. Subsequently, the algorithm predicts missing values using completed (filled) values of neighboring observations. The KNN algorithm, automatically implemented in the proposed framework, utilizes the nearest neighbors' distance to fill in the missing values. Next, the outliers in data are examined on the dataset. To this end, the target data (i.e., productivity) is sorted for each activity, and the filtered output dataset is obtained by combining visual assessment and the Inter-Quartile Range (IQR) proximity rule to conservatively detect, eliminate outliers and prevent possible outlier entry.

Data normalization is the last step of the “data extraction and preprocessing” phase. Rescaling is a technique to transform data into a common scale without distorting differences in the ranges of values. As a result, the quantitative difference between the features does not affect the significance of the features. The Rescaling technique is utilized in the framework to scale values from zero to one through Eq. (1).

$$z_i = \frac{x_i - x_{\min}}{x_{\max} - x_{\min}} \tag{1}$$

4.3. Model preparation

4.3.1. Model generation

A random search method is used to tune the hyperparameters of models and compare different model types. This method randomly takes sample points in the search space (i.e., the predefined domain of each hyperparameter). The system operates a variety of ML model types to compare various model types and improve the estimation. The models are divided into two categories of single and ensemble models. Single models (e.g., decision trees and neural networks) rely on one model to develop the prediction. While ensemble models such as bagging, stacking, and boosting improve the prediction performance by combining the estimations from multiple models. Despite the improvement, ensemble models take more time in generation and runtime than single models. Thus, the paper comprehensively compares both categories with respect to computational complexity/simplicity and accuracy.

4.3.2. Training-validation-test set split

Model generation specification is followed by Training-validation-test splitting to divide the refined data. Firstly, according to the bins' boundaries, the dataset is apportioned between bins. The train and test datasets are then obtained by randomly sharing bins' data with the train and test dataset, considering the assigned ratio. In the proposed system, the train set splits into groups using a k-fold cross-validation procedure to evaluate and modify ML models on an unseen data sample.

4.3.3. Data augmentation

Data augmentation is a technique to artificially expand the diversity of training sets and decrease models' variance by improving models' generalization (robustness). As a simple form of data augmentation technique, adding noise to inputs is an approach to make the input space smoother, easier to learn, and prevent overfitting. This approach results better in small datasets. Hence, the system adds Gaussian noise to the train dataset of each operation's dataset and forms new train datasets. Furthermore, according to the probabilistic nature of productivity, Gaussian noise maps models better by adding new train data rows near to original train data rows. For implementation, corresponding to each data row in the train set, a new data row with the same feature values is defined, and the target value (i.e., productivity) is obtained from the sum of the current target value and a random sampled number from the standard normal distribution.

Adding Gaussian noise into the training datasets faces ML models with new training sets where the prior knowledge is planted. Serialized models as the framework's main outputs are desired to be functional and practical in various project management fields and must be tested with wide-ranging input data to predict WCPs. Furthermore, the research is supposed to be applicable to different types of projects and operations and be able to handle unseen training sets (i.e., DWRs). Thus, embedding such a technique in the framework enhances the models' robustness, which is essential to preserve the framework's flexibility.

In terms of the supervised ML models' problem type, several studies (e.g., Portas and AbouRizk [68]) developed classification models using the data discretization (binning) process to estimate construction productivity ranges (classes of data points) instead of continuous values. Similarly, the proposed framework discretizes input targets into bins to get reliable estimation models and addresses uncertainties embedded in operations' progress. As a classification problem, developed ML models predict productivity ranges, probabilistic instead of deterministic values.

In the classification problem, determining bins' boundaries and subsequently dividing input data based on the defined bins are required. Since there is a possibility of an imbalance amount of data in each class (i.e., bin) and overlooking the minority class entirely, which leads to preserving a bias toward the majority class, the framework addresses the problem via the oversampling technique. This approach randomly duplicates examples in the minority class without adding additional knowledge to the system.

4.4. Model training

4.4.1. Feature selection (scenario-based wrapper method)

In the proposed framework, candidate features form scenarios. Each scenario contains important features and a combination of candidate features. The system trains corresponding ML models to the scenarios and compares the performances. The top model indicates the best combination of effective features, and the system eliminates features with less contribution to productivity estimation.

4.4.2. Bin boundary optimization

To improve the performance of models, an enhancement algorithm is

developed to investigate various bin boundaries. Traditional discretization methods, such as equal width binning and equal frequency binning, have some drawbacks. For instance, equal-width binning ignores the data distribution, and some bins may be empty. To resolve the shortcoming, Salman and Kecman [69] proposed, "The right level of a discretization should be determined by cross-validation." Thus, the developed algorithm in this paper improves the performance by investigating different boundaries through cross-validation. The algorithm creates several levels, constrained by a minimum amount of data in each bin, and each level represents a combination of boundaries. In the procedure, the first level is equivalent to the Equal frequency binning method's boundaries, and the higher the level, the more it leans toward the Equal width binning method's boundaries. Hence, improved bin boundaries are acquired by comparing the performances of each corresponding model.

4.4.3. Hyperparameter tuning

The framework generates single and ensemble models to optimize hyperparameters, compare various model types, and serialize the chosen productivity prediction model and its properties. ML models mentioned in Tables 2 and 3 are generated by examining various combinations of hyperparameters using the random search method. As a result, the proposed framework obtains single and ensemble hyperparameter-tuned prediction models.

In the proposed workflow, generating single models occur before ensemble models. To develop ensemble models, the system takes advantage of pretrained and hyperparameter-tuned single models as base estimator models (in contrast to weak learners common in ensemble models). Consequently, to train ensemble models, the system considers pretrained single models as a tuned hyperparameter, and the randomized search algorithm tunes the rest of the hyperparameters (smaller search space).

According to Tables 2 and 3, implemented ML models comprise 13 supervised algorithms of KNN, SVM, logistic regression, Multi-Layer Perceptron (MLP), ridge regression, decision tree, RF, bootstrap aggregating (bagging), adaptive boosting (AdaBoost), gradient boosting, histogram-based gradient boosting, voting, and stacking. The mentioned algorithms are briefly described as follows.

The KNN algorithm aims to find the most similar data point to the target by calculating distances (e.g., Euclidean distance [70]). SVM aims to find a hyperplane with the maximum distance between data points of classes in N-dimensional space (N refers to the number of features in the problem) to classify the data points into distinct regions [71]. Logistic regression uses the logistic function to model a binary outcome for classification problems. In contrast with linear regression, logistic regression develops models without requiring a linear relationship between input and output variables [72]. MLP is a developed version of perceptron (a binary classifier) and refers to networks composed of multiple layers of the perceptron. The model comprises input and output layers and several hidden layers, each containing several neurons. Each layer processes data and its outputs are fed to the following layer to generate a set of outputs from a set of inputs [73]. Ridge regression is a modified version of linear regression developed to omit the possible sensitivity of models to inputs and their instability by altering the loss function [74]. The decision tree is a tree-like model of decisions that only contains conditional control statements, and the output is predicted based on these learned rules [75].

The RF is an ensemble learning method that includes several individual decision trees developed on different samples at the training

Table 2
Single ML models used in the framework.

Row number	ML model title	Description
1	Logistic regression l2	Penalty: L2 penalty term
2	Logistic regression l1	Penalty: L1 penalty term
3	Logistic regression elastic net	Penalty: Both l1 and l2 penalty terms
4	K-nearest neighbor (KNN)	–
5	Decision tree	–
6	Multi-layer perceptron (MLP) SGD	Solver: stochastic gradient descent (SGD)
7	Multi-layer perceptron (MLP) adam	Solver: a stochastic gradient-based optimizer proposed by Kingma, Diederik, and Jimmy Ba (ADAM) [82]
8	Support Vector Machine (SVM)	–
9	Ridge	–

Table 3
Ensemble ML models used in the framework.

Row number	ML model	Description	Row number	ML model	Description
1	Random forest (RF)	–	13	Soft voting	Voting rule: soft
2	Bagging LR l2	Base estimator: logistic regression Penalty: l2	14	Hard voting	Voting rule: hard
3	Bagging LR l1	Base estimator: logistic regression Penalty: l1	15	Stacking LR l2	Final estimator: logistic regression Penalty: l2
4	Bagging LR Elastic net	Base estimator: logistic regression Penalty: l2 & l1	16	Stacking LR l1	Final estimator: logistic regression Penalty: l1
5	Bagging R	Base estimator: ridge	17	Stacking LR Elastic net	Final estimator: logistic regression Penalty: l2 & l1
6	Bagging SVC	Base estimator: C-support vector classification	18	Stacking R	Final estimator: ridge
7	Bagging KNN	Base estimator: K-nearest neighbor	19	Stacking SVC	Final estimator: C-support vector classification
8	Bagging RF	Base estimator: Random forest (RF)	20	Stacking KNN	Final estimator: K-nearest neighbor
9	Bagging MLP SGD	Base estimator: Multi-layer perceptron Solver: SGD	21	Stacking DT	Final estimator: Decision tree
10	Bagging MLP adam	Base estimator: Multi-layer perceptron Solver: adam	22	Stacking MLP SGD	Final estimator: Multi-layer perceptron Solver: SGD
11	Gradient boosting	–	23	Stacking MLP adam	Final estimator: Multi-layer perceptron Solver: adam
12	Histogram-based gradient boosting	–	24	AdaBoost	–

stage. Each individual tree in an RF outputs a prediction, and the final prediction of the RF is the majority vote for classification and average in the case of regression [76]. Similar to the RF, the bagging algorithm includes a number of decision trees, whereas each decision tree is fitted on a random sample of the dataset [77]. Adaboost is another ensemble algorithm that combines multiple weak classifiers to build a strong classifier. This algorithm is called Adaptive Boosting since the algorithm recorrects weights and assigns higher weights to incorrectly classified instances, which results in bias and variance reduction [78]. Gradient boosting is similar to Adaboost, except that AdaBoost utilizes an exponential loss function that is sensitive to outliers, while Gradient Boosting can use any differentiable loss function. Consequently, Gradient Boosting is more robust to outliers than AdaBoost [79]. The slow training phase is a major drawback of the gradient boosting method, which is challenging for problems with a large training dataset. Histogram-based gradient boosting is a modified gradient boosting method that improves the efficiency of gradient boosting's training phase by discretizing the continuous input variables to a few hundred unique values. The voting classifier trains various base models or estimators and predicts the output based on aggregating the prediction of each base model. Finally, stacking combines outputs from individual models by providing a weighted sum prediction. During the training phase, the base model that outperforms gets a higher weight for the prediction [80,81].

4.4.4. Model selection

The best productivity estimation model is selected from all trained models by comparing the confusion matrices and numerical outputs described below. Confusion matrices are as important as accuracy, recall and precision to compare and determine the best models. For summarizing purposes, the multi-criteria assessment result is called performance in the paper. The chosen productivity estimation model, other specifications, and information are serialized to be utilized in project planning and scheduling. In addition to serializing and storing trained ML models, other automatically generated and stored outputs include:

4.4.4.1. ML model outputs. This output file is defined to store the numerical results of the system, comprising the execution time in seconds, the accuracy of training data, the accuracy of test data, the average precision, the average recall, and the best combination of hyperparameters for each type of model.

Accuracy is calculated by dividing the number of correct predictions by the total number of predictions (Eq. (2)). In an imbalanced classification problem with more than two classes, recall is calculated as the sum of true positives across all classes divided by the sum of true positives and false negatives across all classes (Eq. (3)). Also, precision is calculated as the sum of true positives across all classes divided by the sum of true positives and false positives across all classes (Eq. (4)). F1 index as the harmonic mean of precision and recall can evaluate results by considering the precision and recall indices simultaneously with equal weight (Eq. (5)). where TP, TN, FP, and FN are the abbreviation of true positive, true negative, false positive and false negative, respectively.

$$Accuracy = \frac{(TP + TN)}{(TP + FN + TN + FP)} \tag{2}$$

$$Sensitivity = \frac{TP}{(TP + FN)} \tag{3}$$

$$Precision = \frac{TP}{(TP + FP)} \tag{4}$$

$$F_1 = \frac{2\,TP}{(2\,TP + FP + FN)} = 2 \times \frac{Precision \times Recall}{Precision + Recall} \tag{5}$$

In addition to the abovementioned metrics, Mean Squared Error (MSE) and Mean Absolute Error (MAE) have also been utilized. According to the research conducted by Gaudette and Japkowicz [83], these metrics demonstrated exemplary performance in ordinal classification problems. As an advantage, MSE and MAE measure the variance between the predicted and observed categories which is beneficial. Because the more significant the gap between the misclassified category and the observed category, the more penalty is accounted for. MAE metric evaluates the absolute distance of the observations (the entries of the dataset) to the predictions on a regression (Eq. (6)), whereas MSE is the average squared distance between observed and predicted values (Eq. (7)). where N, y_i, and $\hat{y}$ are the size of the test set, the actual category number, and the predicted category number, respectively.

$$MAE = \frac{1}{N}\sum_{i=1}^{n}|y_i - \hat{y}| \tag{6}$$

$$MSE = \frac{1}{N}\sum_{i=1}^{n}(y_i - \hat{y})^2 \tag{7}$$

The confusion matrix, as the subsequent output, is used to evaluate the performance of the ML models in classification problems. The horizontal axis of the matrix represents the predicted category, and the vertical axis represents the actual category.

4.4.4.2. ML model saving and reusing. Required data to utilize the developed ML models and predict productivity are gathered in several tables within the database. The database facilitates the reusability and interpretability of the stored models comprehensively, as described in Table 4 and Fig. 5. Since the presented framework is planned to be used in project planning and scheduling phases, a web-based GUI is developed. The GUI contains a main page to analyze data through several planning modules and several data pages to illustrate/modify data embedded in data tables for users.

5. Implementation environment

The proposed framework's implementation architecture is investigated by specifying programming language, libraries, and I/O formats (Fig. 6). The field data is provided in the Excel file format and required to be refined (i.e., data cleaning) to address the shortcomings and embedded issues of raw data. ML models are then developed on the cleaned data of each activity separately. The data analysis and development of ML models are carried out in the Python programming language. Scikit-learn [84], Pandas [85], Imblearn [86], and Scipy [87] libraries are utilized to analyze and refine the data as well as to build ML models.

As shown in Fig. 6, four main modules are defined to predict WCP. In the first module (i.e., data extraction module), the system extracts the input data, which is a collection of projects' DWRs, into a predefined standard format. The second module (i.e., Preprocessing module) analyzes and cleans the extracted data to obtain the refined data.

In the single models' module, using libraries specific to data mining, various types of ML productivity estimation models are implemented to compare productivity estimation models in addition to generating supervised ML models. To enhance the performance of the developed models (baseline models), hyperparameter tuning is also carried out in this module. The module's outputs are productivity prediction models, modeling results (e.g., confusion matrix, accuracy, precision, recall, tuned hyperparameters, and execution time), models' information (e.g., features' range, historical data, and ML model developed operations), and estimator models. The estimators are ML models considered the base model (i.e., weak learners) in several ensemble learning models. In this regard, the module stores the single model's specifications with tuned hyperparameters as estimator models in a serialization format (i.e., PKL). The PKL format enables the system to (1) serialize objects (a byte stream that represents the objects), (2) write serialized objects into files on local storage, (3) read serialized files, and (4) deserialize objects

Table 4
The database's tables' objectives and description.

Table title	Objective(s)	Description
Model properties	1. Fitting probability distributions/ Defining empirical distribution 2. Connecting ML models' serialized files with the database	"Model properties" is the main table that specifies information, such as (I) ID, (II) models' type, and (III) bin boundaries.
Feature range	1. Determining feature lists 2. Normalizing input data row 3. Denormalizing models outputs	To utilize models, normalized feature values are required, and to interpret model outputs, denormalization is essential. Thus "Feature range" is defined to enable quantities' normalization and denormalization through their minimum and maximum value. Moreover, the table determines the models' demanded features to get their value in order.
Historical data	1. Fitting probability distributions/ Defining empirical distribution	"Historical data" authorizes systems to estimate productivity probabilistic using empirical distributions or fitted probability distributions on historical data.
Operation	1. Creating a link between (I) models' properties and (II) operations and their specifications	The "Operation" table connects ML models' specifications to operations and their properties. Subsequently, each operation's corresponding models' serialized file is determined.

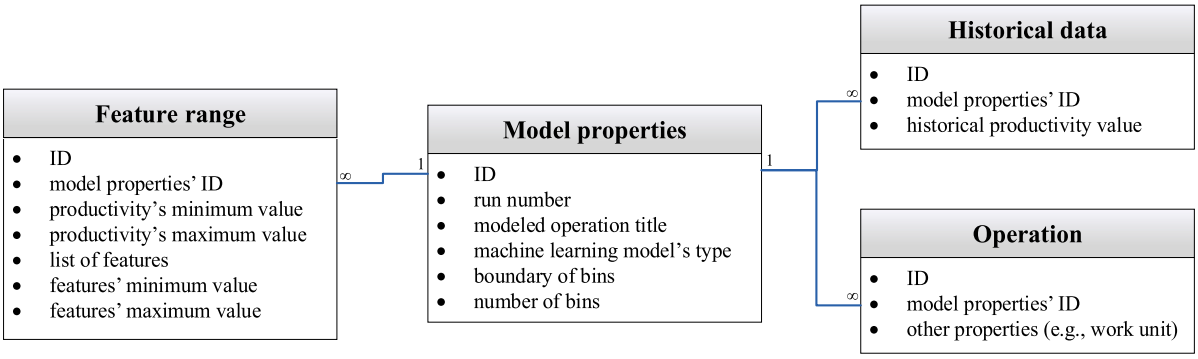

Fig. 5. The database's tables' columns and relationships.

back into the program at runtime. Also, the top trained models in this module are serialized in the Open Neural Network Exchange (ONNX) format, which is an open format built to represent ML models. In the proposed research, the difference between serialized models with PKL and ONNX formats is in their application. The PKL format is used to serialize the model and its hyperparameters in an untrained state and applied in the ensemble models' module. In contrast, the ONNX format is used to serialize the model and its hyperparameters in a trained state to estimate productivity.

In the ensemble models module, previously serialized estimator models and stored information are used as input data to develop ensemble learning models. In this module, the goal is to combine single ML models and create various ensemble models. Similar to the single models' module, the models' information and structure, productivity prediction models, test results, and tuned hyperparameters are stored for future applications.

6. Case study

In this section, the proposed system is investigated and implemented using actual DWRs of four projects. Since pipe diameter is a critical factor for productivity estimation in pipeline projects, DWRs are gathered from projects with different pipe diameters (Table 5). The required data is extracted according to the proposed framework, and the data preprocessing is performed. During the extraction process, the data associated with nine main activities, including daily progress, labors (type and number), equipment (type and number), and weather conditions, are obtained. The implemented phases of the system are described in the case study.

6.1. Data extraction and preprocessing

In this study, the project progress can be evaluated based on DWRs, collected from the site using Excel worksheets. The "Main" worksheet quantifies general information such as the day of the week, precipitation, wind, relative humidity, and temperature. The "Work volumes" worksheet describes each operation's daily work volume and measurement unit. Eventually, in the "Labros" and "Machinery" worksheets, (1) various types of labors with their quantity and (2) various types of construction machinery with their quantity are clarified, respectively.

At the first stage of the extraction algorithm, titles such as "precipitation", "temperature", and "resource types" are extracted. Due to the placement of the "quantity" cells next to the "title" cells, the second stage of the algorithm is activated, and corresponding quantities for each title are extracted. Fig. 7. illustrates a part of the extracted data in the predefined standard format. In Fig. 8, the variation in the number of welders, welder assistants, and grinding responsible for the 10″ pipe diameter project is illustrated.

Some features are combined according to the functionality or affective factors, such as engine power and volumetric capacity. Table 6 lists resources before and after the combination. As shown in the table, various types of equipment with similar functionality are merged, defining general features to be expendable with new DWRs in the future. Several labors are also merged, given the expert judgment.

To obtain the correlation and relationships between features and target (i.e., productivity), the data are analyzed and illustrated in the form of bar charts, matrices, and graphs. For instance, Figs. 9-11 show the results of the automated performed analysis on routing operation. Thereby, all features are categorized into irrelevant features, candidate features, and important features.

Due to human errors, a slight amount of data is missing, and their corresponding quantity worksheet cells are empty in the DWRs. According to the framework, missing data are imputed using the KNN algorithm to fill the empty cells. As a sample, Fig. 12 depicts 13 data rows of routing operation before and after the imputation. Outliers in data are also eliminated, and data are normalized based on the described methodology. Consequently, refined data are acquired, finalizing the framework's first phase.

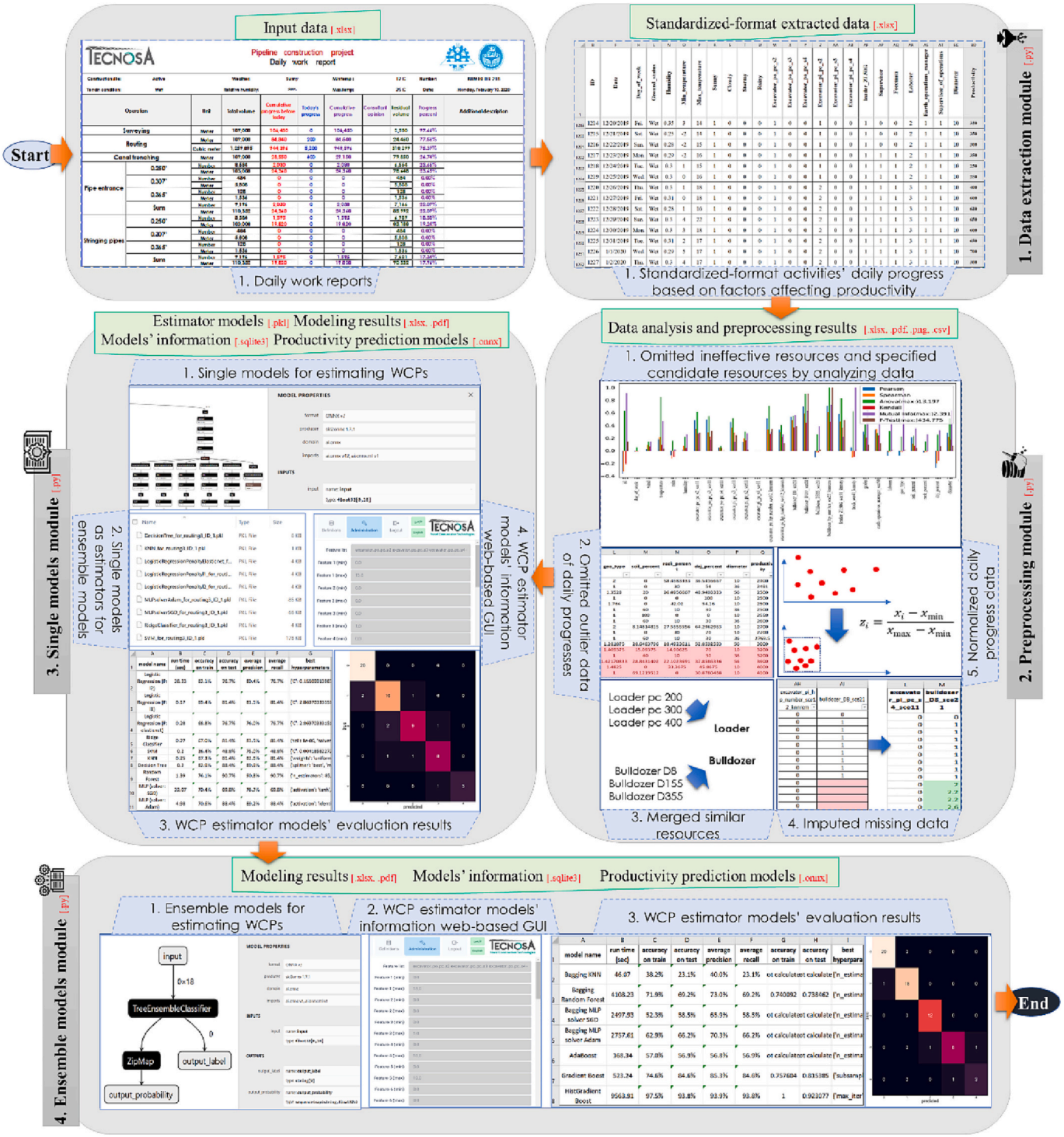

Fig. 6. The proposed framework's general architecture.

6.2. Model preparation

In this phase, the required properties to generate ML models are determined. The main indispensable specifications are listed in the following.

1) Model types: Desired classification models are assigned to compare and examine productivity prediction models. The embedded ML models are shown in Tables 2 and 3.

2) Hyperparameters' domain: Since the framework utilizes the randomized search algorithm to enhance the performance of models' prediction, this phase allocates a particular range to each hyperparameter.

After the specifications, according to the methodology, input data are divided into (1) training, (2) validation, and (3) test datasets. Adding Gaussian noise and oversampling are the last steps that took place before training the model.

Table 5
Details of DWRs received from consultants and contractors of pipeline construction projects.

Row number	Project area	Diameter (inch)	DWRs' start date	DWRs' finish date
1	South-southwest	56	2018-10-23	2019-12-16
2	South-southwest	8	2018-10-23	2019-12-16
3	Southeast	10	2019-07-08	2020-06-07
4	South-southwest	36	2018-05-30	2020-06-24

6.3. Model training

Given the candidate features obtained from the "Data extraction and preprocessing" phase, feature selection scenarios are defined for each operation as presented in Table 7. The best scenario (i.e., the combination of candidate features) is determined by training the model using each scenario and comparing their performance. For instance, Fig. 13 shows the average accuracy of each scenario in the models for canal trenching activity. According to the results, the second scenario is selected as the best scenario, and the same process is followed for other activities. The outcome of the feature selection procedure is demonstrated in Table 8.

Next, bin boundaries are investigated for several operations. The outcome is illustrated in Table 9.

A variety of single and ensemble ML models are implemented for each activity. The system optimizes each model using the randomized search as a hyperparameter tuning algorithm. Afterward, the optimized trained models, along with their properties (tunned hyperparameters), are stored automatically. For example, the optimization results for routing activity are presented as follows. Table 10 indicates the result for single models, whereas Tables 11 and 12 represent the ensemble models' results.

Confusion matrices are also generated to interpret and compare the performance of the prediction models. For instance, the confusion matrix of MLP (solver: adam), Histogram-based gradient boosting, and Stacking (final estimator: MLP, solver: SGD) models for "routing" activity are depicted in Fig. 14. The values on the main diagonal indicate the number of data predicted in the correct bin, and the other values are the number of data that the estimated bin value differs from the actual bin number. For example, a value of 2 in the cell (1,0) in Fig. 14(a) demonstrates that two test data were misclassified. Based on the results, MLP in single learning models, stacking in ensemble learning models, and overall, the stacking model with final estimator as MLP (solver: SGD) are found to be the best models to estimate routing operations' productivity. Table 13 lists the best-identified models for each activity. Data such as model specifications, historical productivity data, and feature ranges are collected in various data tables. These data tables are accessible through the web-based GUI for future reuse. For instance, the "Feature range" table (Fig. 15) contains information about features with their minimum and maximum values, which is required for normalization and denormalization in the future.

7. Discussion

7.1. Feature selection

Although some efforts to present a new hybrid feature selection method (e.g., Hsu et al. [65], and Lan et al. [64]), special circumstances and problems' inherent characteristics (e.g., operations' inherent characteristics in this paper) are not often considered in feature selection procedures. The mentioned deficiency is indicated in cases where the data are interpretable, and experts with a level of confidence can claim whether the features are effective. In this regard, the proposed framework customized the wrapper method. Instead of training data with random feature sets, feature sets were defined via candidate features that originated from filter methods' results and expert knowledge. Expert knowledge entailed the effectiveness of features such as weather and terrain conditions, labors, and equipment on WCP. As a side task, the effectiveness of the day-of-week on operations' progress was

	B	F	H	L	N	O	P	R	S	T	U	W	X	Y	Z	AA	AB	AF	AP	AQ	AR	AS	AT	BC	BD
1	ID	Date	Day_of_week	Ground_status	Humidity	Min_temperature	Max_temperature	Sunny	Cloudy	Stormy	Rainy	Excavator_po_pc_s2	Excavator_po_pc_s3	Excavator_po_pc_s4	Excavator_pi_pc_s2	Excavator_pi_pc_s3	Excavator_pi_pc_s4	loader_ZL50G	Supervisor	Foreman	Laborer	Earth_operations_manager	Supervisor_of_operations	Diameter	Productivity
1219	1214	12/20/2019	Fri.	Wet	0.35	3	14	1	0	0	0	1	0	0	1	0	0	1	0	0	2	1	1	10	320
1220	1215	12/21/2019	Sat.	Wet	0.25	-2	14	1	0	0	0	1	0	0	1	0	0	1	0	0	2	1	1	10	350
1221	1216	12/22/2019	Sun.	Wet	0.28	-2	15	1	0	0	0	1	0	0	1	0	0	1	0	0	2	1	1	10	300
1222	1217	12/23/2019	Mon.	Wet	0.29	-2	16	1	0	0	0	1	0	0	1	0	0	1	1	1	2	1	1	10	350
1223	1218	12/24/2019	Tue.	Wet	0.3	1	15	1	0	0	0	1	0	0	1	0	0	1	1	1	2	1	1	10	250
1224	1219	12/25/2019	Wed.	Wet	0.3	0	16	1	0	0	0	1	0	0	1	0	0	1	1	1	2	1	1	10	250
1225	1220	12/26/2019	Thu.	Wet	0.3	1	18	1	0	0	0	1	0	0	2	0	0	1	1	1	3	1	1	10	400
1226	1221	12/27/2019	Fri.	Wet	0.31	0	18	1	0	0	0	1	0	0	2	0	0	1	1	1	3	1	1	10	600
1227	1222	12/28/2019	Sat.	Wet	0.28	1	16	1	0	0	0	1	0	0	2	0	0	1	1	1	3	1	1	10	620
1228	1223	12/29/2019	Sun.	Wet	0.3	4	22	1	0	0	0	1	0	0	2	0	0	1	1	1	3	1	1	10	650
1229	1224	12/30/2019	Mon.	Wet	0.3	3	18	1	0	0	0	1	0	0	2	0	0	1	1	1	3	1	1	10	600
1230	1225	12/31/2019	Tue.	Wet	0.31	2	17	1	0	0	0	1	0	0	2	0	0	1	1	1	3	1	1	10	650
1231	1226	1/1/2020	Wed.	Wet	0.29	5	17	1	0	0	0	1	0	0	2	0	0	1	1	1	3	1	1	10	700
1232	1227	1/2/2020	Thu.	Wet	0.3	4	17	1	0	0	0	1	0	0	2	0	0	1	1	1	3	1	1	10	300

Fig. 7. A part of the extracted raw data (canal trenching operation).

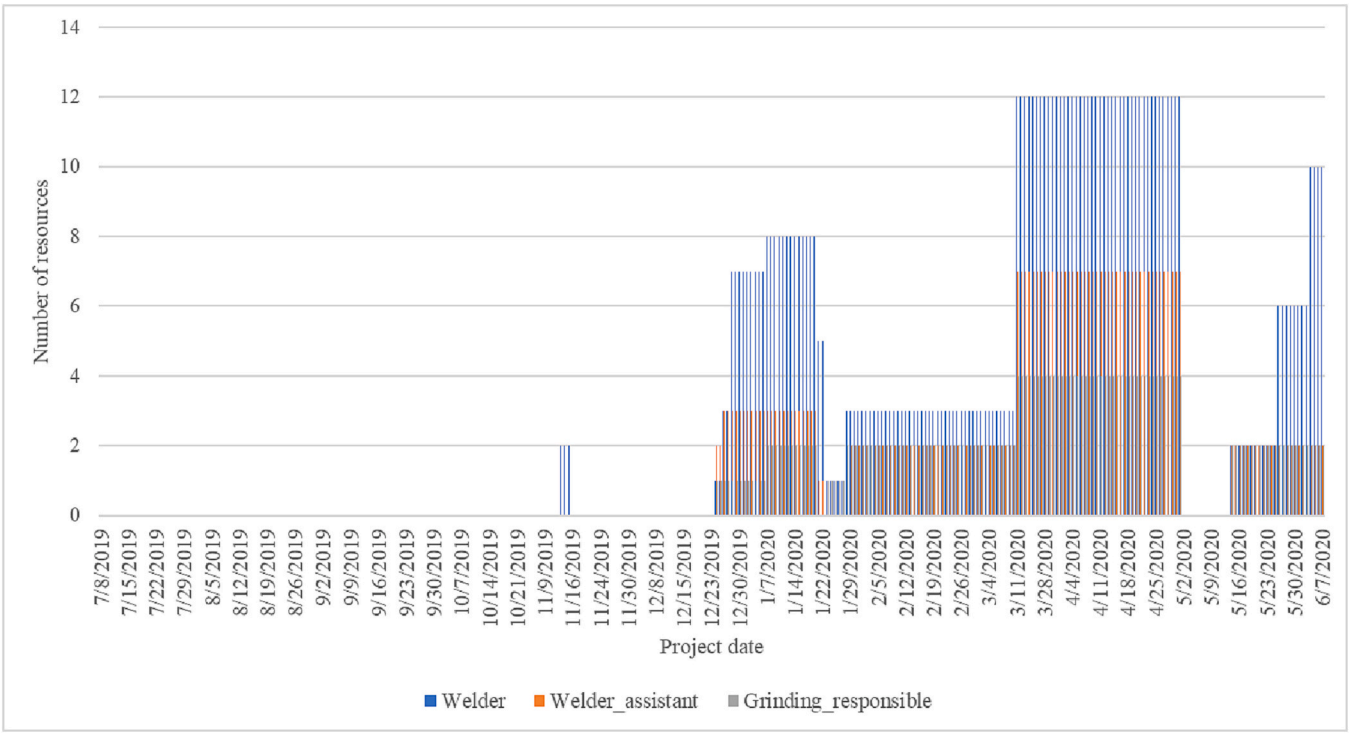

Fig. 8. Welder, welder assistant, and grinding responsible changes in the 10″ project (extracted from DWRs).

Table 6
Case study projects' resources before and after dimensionality reduction.

Row number	Operation title	Resources (before combination)	Resources (after combination)	Row number	Operation title	Resources (before combination)	Resources (after combination)
1	Surveying	Surveyor supervisor	Surveyor supervisor and surveyor	20	Welding	D9 sideboom	Side-boom and crane
2	Surveying	Surveyor	Surveyor supervisor and surveyor	21	Welding	25_ton crane	Side-boom and crane
3	Routing	Excavator po_pc_s2	Bucket excavator	22	Welding	ZL50 Loader	Loader
4	Routing	Excavator po_pc_s3	Bucket excavator	23	Welding	Fitter	Fitter and assistant
5	Routing	Excavator po_pc_s4	Bucket excavator	24	Welding	Fitter assistant	Fitter and assistant
6	Routing	Excavator pi_pc_s2	Breaker excavator	25	Welding	Electrician responsible	Electrician and mechanical responsible
7	Routing	Excavator pi_pc_s3	Breaker excavator	26	Welding	Mechanical responsible	Electrician and mechanical responsible
8	Routing	Excavator pi_pc_s4	Breaker excavator	27	Welding	Bender	Bender and assistant
9	Routing	D8 bulldozer	Bulldozer	28	Welding	Bender assistant	Bender and assistant
10	Routing	D155 bulldozer	Bulldozer	29	Canal trenching	Excavator po_pc_s2	Bucket excavator
11	Routing	D355 bulldozer	Bulldozer	30	Canal trenching	Excavator po_pc_s3	Bucket excavator
12	Routing	ZL50 Loader	Loader	31	Canal trenching	Excavator po_pc_s4	Bucket excavator
13	Stringing pipes	25_ton crane	Crane	32	Canal trenching	Excavator pi_pc_s2	Breaker excavator
14	Stringing pipes	10_ton crane	Crane	33	Canal trenching	Excavator pi_pc_s3	Breaker excavator
15	Stringing pipes	ZL50 Loader	Loader	34	Canal trenching	Excavator pi_pc_s4	Breaker excavator
16	Stringing pipes	Rigger	Rigger and assistant	35	Canal trenching	ZL50 Loader	Loader
17	Stringing pipes	Rigger assistant	Rigger and assistant	36	Canal trenching	Earth operations manager	Earth operations manager and supervisor
18	Lowering pipes	ZL50 Loader	Loader	37	Canal trenching	Supervisor of operations	Earth operations manager and supervisor
19	Backfilling	ZL50 Loader	Loader				

investigated. Accordingly, for each operation in the case study, a particular scenario was added to assess whether the day of the week feature is influential on productivity. Thus, according to the DWRs, the day-of-week feature affects the productivity of surveying, routing, canal trenching, and stringing pipes operations (Table 8).

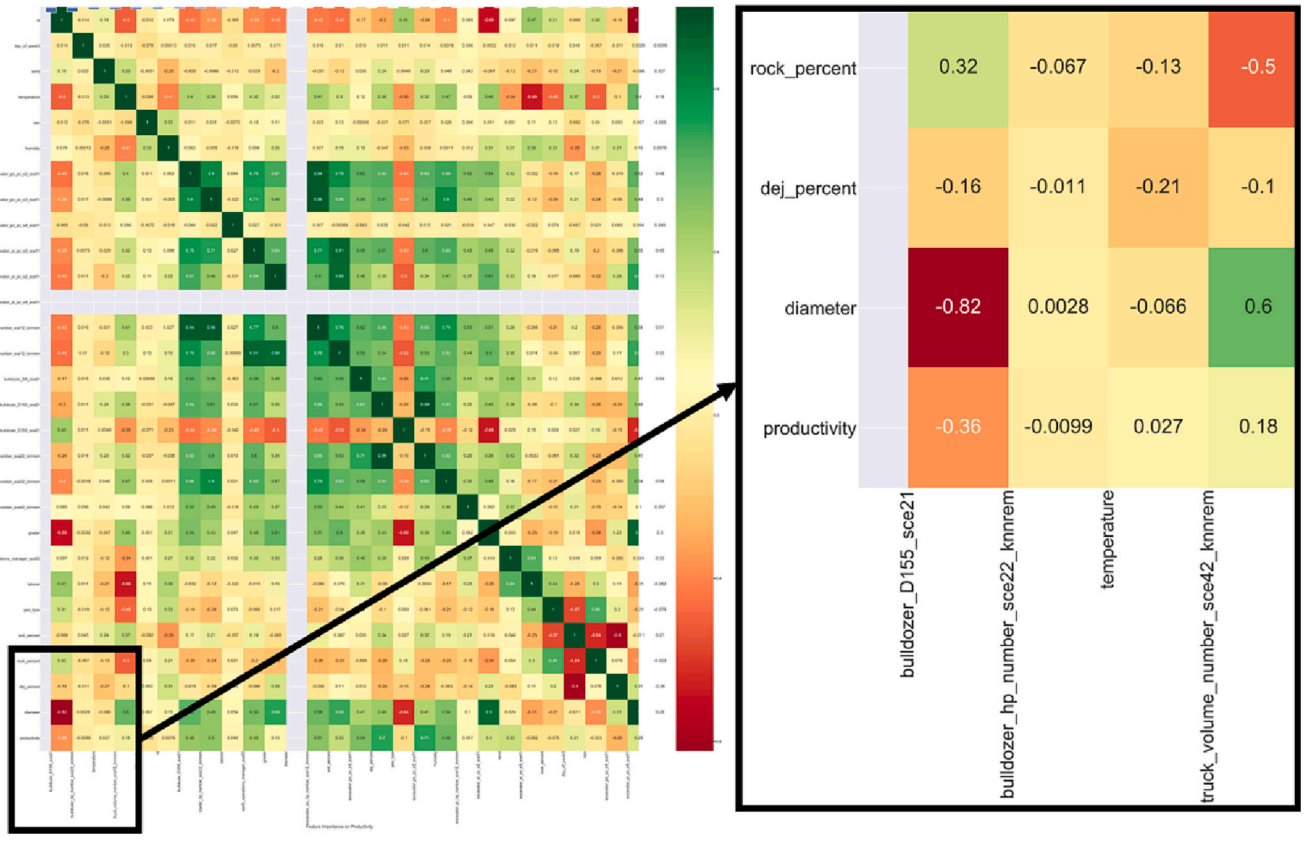

Fig. 9. Heat map for routing operation based on the case study.

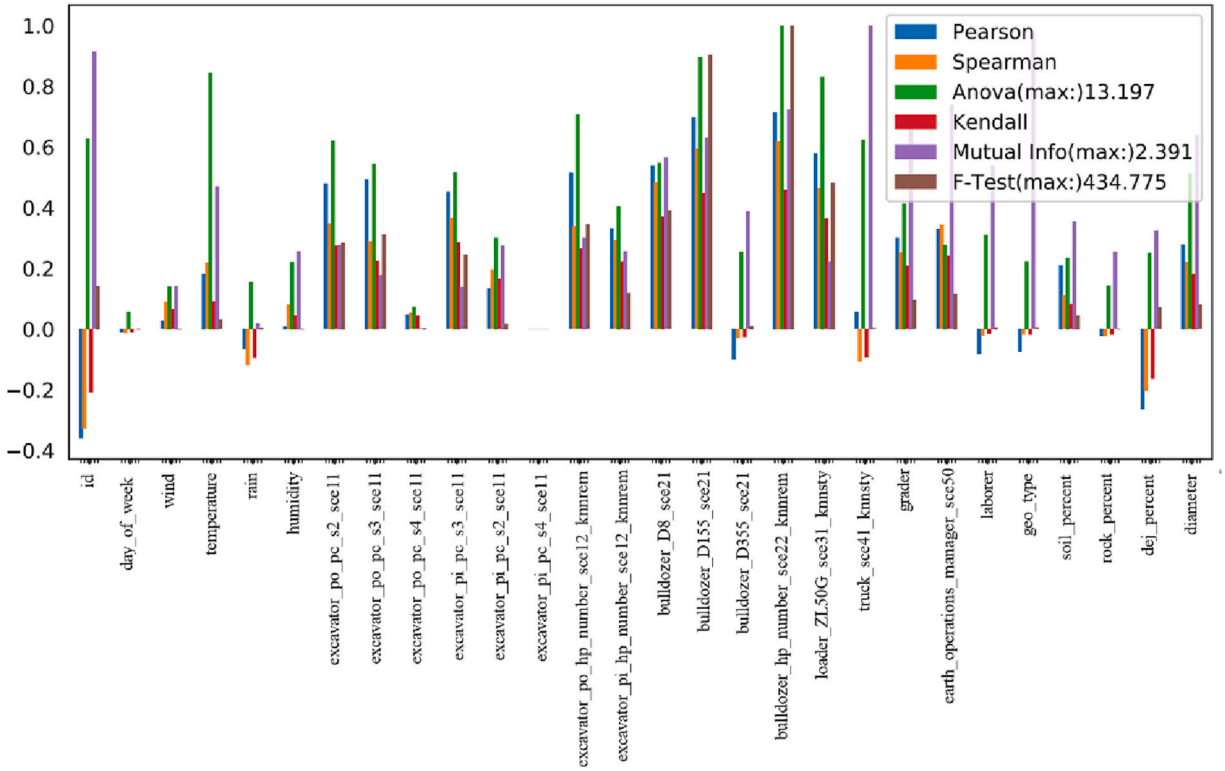

Fig. 10. Bar chart of analyzes performed on the relationship between features and productivity (routing operation).

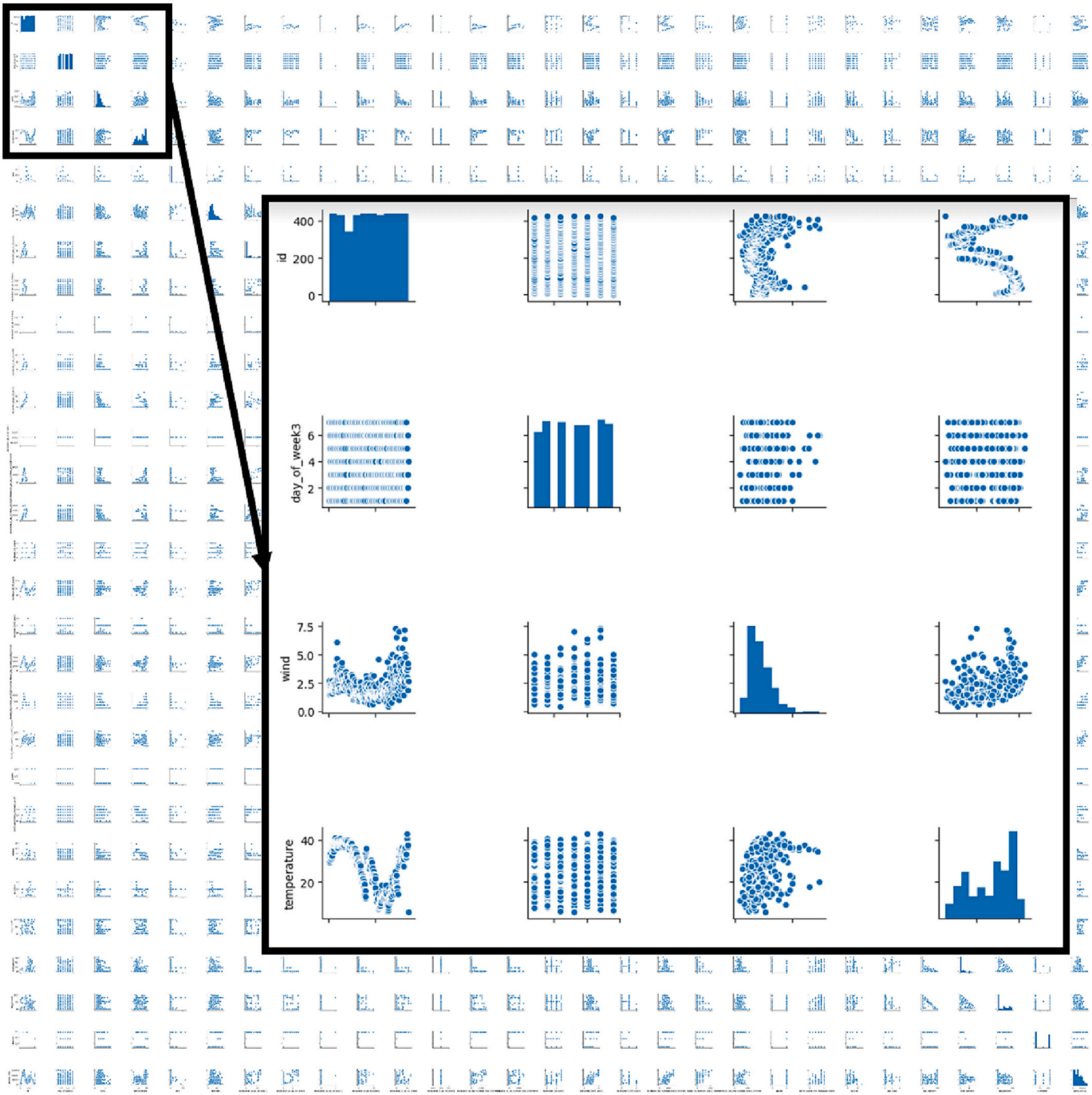

Fig. 11. Pair plot for extracted data (i.e., features and productivity) from DWRs (routing operation).

7.2. Bin boundary determination

By performing the bin boundary determination algorithm on the case study, each bin's boundaries were determined (Table 9). The common point among all operations was the density of bins. Bin boundaries of all operations were obtained at low values, indicating high data resolution at low values. This point illustrates that considering the productivity ranges extracted from DWRs, activities are mostly operated in low ranges, and fitted distributions on productivity values are right-skewed.

7.3. Estimation models enhancement

One of the primary purposes of the framework is to employ various techniques to enhance productivity prediction models. In this regard, comparing different prediction models plays an important role. In every step in the model training phase, including (1) feature selection scenarios' comparison, (2) bin boundaries determination, (3) hyperparameter-tuned models' generation, and (4) comparing and storing final outputs, comparisons were performed with respect to computational complexity and performance, and the best models were distinguished. The system evaluated different feature selection scenarios, boundaries, hyperparameters, and model types through a semi-automatic approach to acquire the top model.

Since it is desired to develop a replicable workflow, the number of executing prediction models in the runtime mode may exceed thousands. For instance, to schedule 10 main operations of the railway

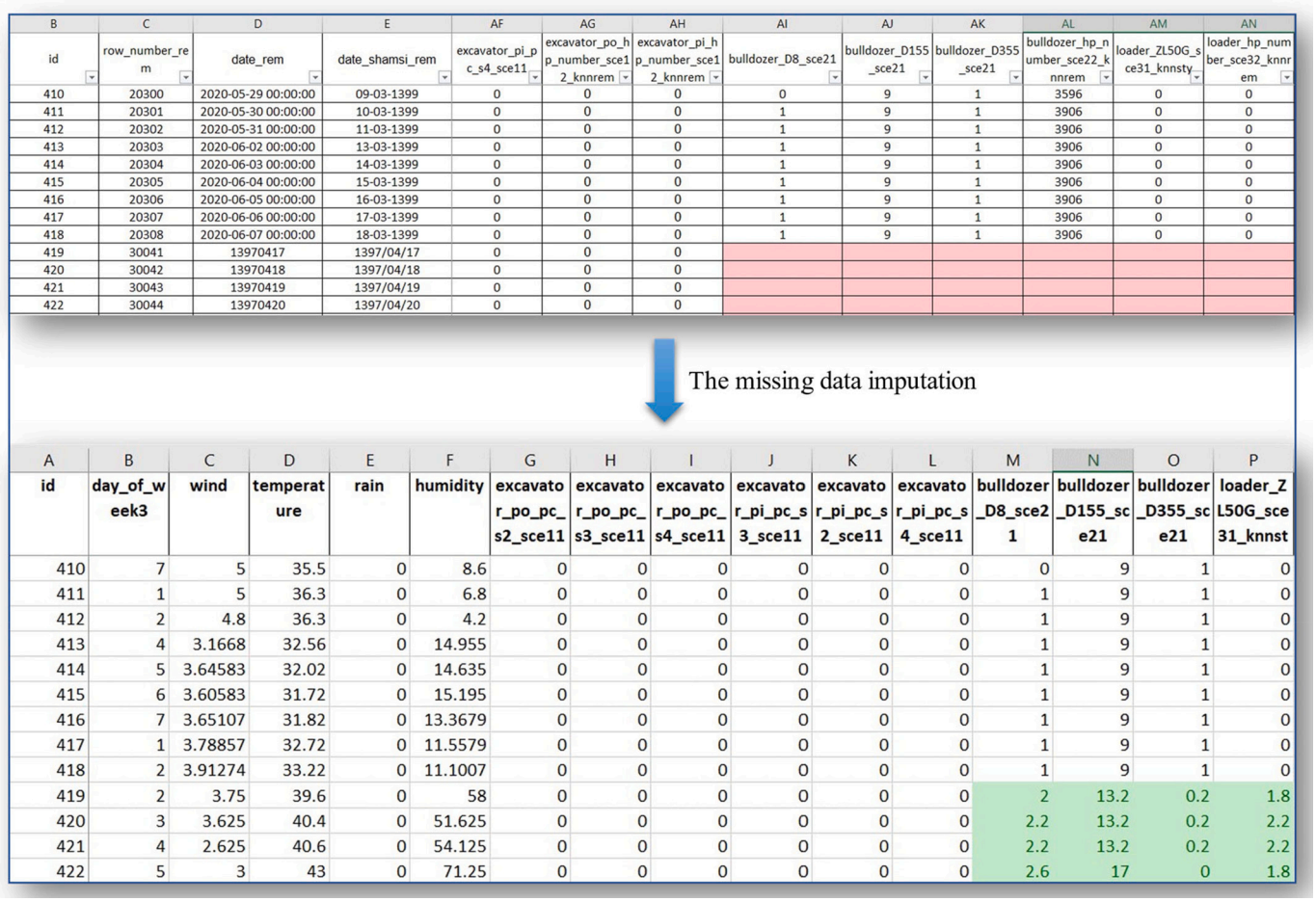

Fig. 12. A part of the routing operation dataset under the process of missing data imputation.

Table 7
Feature selection scenarios for the case study.

Operation title	Scenario number	Candidate feature(s)	Operation title	Scenario number	Candidate feature(s)
Surveying	1	Nothing (base scenario)	Welding	1	Nothing (base scenario)
Surveying	2	Day of the week	Welding	2	Foreman
Routing	1	Nothing (base scenario)	Welding	3	Bender assistant
Routing	2	Excavator	Welding	4	Rigger
Routing	3	Bulldozer	Welding	5	Electrician and mechanical responsible
Routing	4	Earth operations manager	Welding	6	Day of the week
Routing	4	Earth operations manager	Welding	7	Day of the week and foreman
Routing	5	Day of the week	Radiography	1	Nothing (base scenario)
Canal trenching	1	Nothing (base scenario)	Radiography	2	Day of the week
Canal trenching	2	Excavator	Coating	1	Nothing (base scenario)
Canal trenching	3	Foreman, earth operations manager and supervisor	Coating	2	Day of the week
Canal trenching	4	Labor	Lowering pipes	1	Nothing (base scenario)
Canal trenching	5	Day of the week	Lowering pipes	2	Day of the week
Stringing pipes	1	Nothing (base scenario)	Backfilling	1	Nothing (base scenario)
Stringing pipes	2	Stringing pipes responsible	Backfilling	2	Day of the week
Stringing pipes	3	Date of the week			

projects with daily execution of prediction models, given each operation's duration approximately equals 500 days, the processor runs the prediction models over 5000 times. The complexity of a model results in a long duration, whether to train models or evaluate them, undermining the framework's applicability. In addition to accuracy and other ML metrics, execution time with a lower weight is an integral criterion in the proposed system. Thus, it is possible to select single ML models over ensemble models with similar performances and shorter execution times (e.g., the KNN model for radiography activity in Table 13). As it is illustrated in Table 13, in most cases (i.e., operations), ensemble models performed better than single models due to the mentioned criteria.

Among ensemble models, RF and histogram-based gradient boosting were more suitable models for the problem studied herein (i.e., WCP prediction using DWRs). In 7 out of 9 activities, these algorithms were selected as the best productivity prediction models. The most obtained accuracy on the test set for RF belonged to coating activity with 92.50% accuracy, and similarly for histogram-based gradient boosting belonged to routing activity with 93.80% accuracy. For datasets with a high number of features, the random forest performs appropriately. Because each tree in a random forest selects a random subset of features, leading

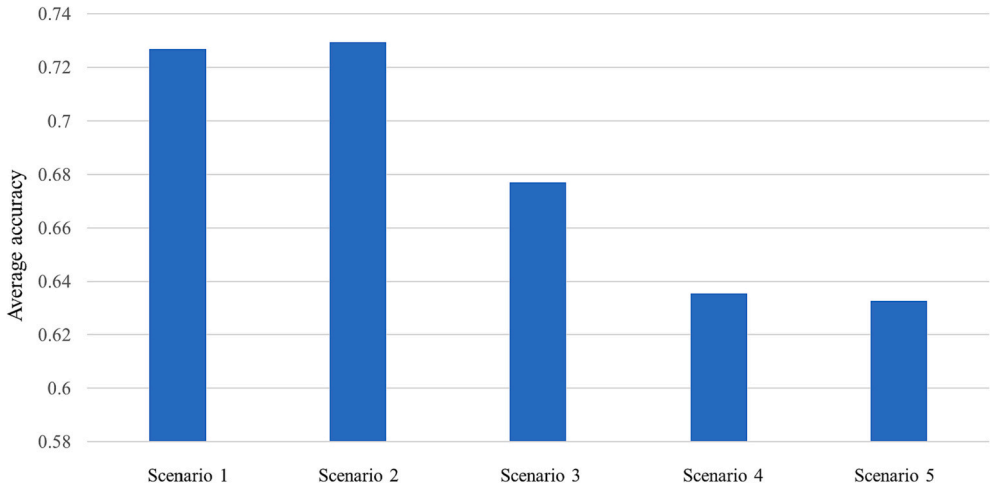

Fig. 13. Average accuracy of defined scenarios for canal trenching operation.

Table 8
The outcome of feature selection scenarios (all equipment are considered with their driver).

Operation title	Effective features (resources)	Other effective features
Surveying	(I) Surveyor supervisor, (II) surveyor, and (III) surveyor assistant	
Routing	(I) Bucket excavator, (II) breaker excavator, (III) bulldozer, (IV) loader, (V) dump truck, (VI) grader, (VII) earth operations manager, and (VIII) labor	
Canal trenching	(I) Bucket excavator, (II) breaker excavator, (III) loader, (IV) foreman, (V) labor, (VI) earth operations manager, and (VII) supervisor of operations	(I) Day of the week* (II) wind (III) temperature (IV) precipitation (V) relative humidity (VI) topography type (VII) soil percentage (VIII) rock percentage (IX) hard rock percentage (X) pipe's diameter
Stringing pipes	(I) Sideboom, (II) crane, (III) trailer truck, (IV) loader, (V) rigger, (VI) rigger assistant, and (VII) labor	
Welding	(I) Sideboom, (II) crane, (III) loader, (IV) bending machine, (V) rigger, (VI) supervisor, (VII) welder, (VIII) welder assistant, (IX) grinder, (X) fitter, (XI) fitter assistant, (XII) service labor. (XIII) labor, (XIV) electrician responsible, (XV) mechanical responsible, (XVI) bender, and (XVII) bender assistant	
Radiography	(I) Radiographer, (II) radiographer assistant, and (III) interpreter	
Coating	(I) Supervisor, (II) foreman, (III) coating responsible, (IV) coating responsible assistant, (V) sandblasting responsible, and (VI) sandblasting responsible assistant	
Lowering pipes	(I) Sideboom and (II) loader	
Backfilling	(I) Loader, (II) dump truck	

* The "day of the week" feature only affects surveying, routing, canal trenching, stringing pipes, and welding operations.

to lower dimensional data. In this study, random forest outperforms since it is common for the datasets to possess a greater quantity of features relative to the number of records. Additionally, due to the manual nature of the data collection for DWRs which are prone to error, outliers exist in the dataset. Histogram-based gradient boosting is a robust method for outliers that tackles them, ultimately resulting in improved performance. Moreover, several prior studies in other fields have achieved the same results. For example, Olson et al. [88] compared the performance of various machine learning algorithms in bioinformatics science. In this regard, the 13 most common algorithms were trained on 165 publicly available datasets for classification problems. By training the models, tree-based ensemble algorithms, including gradient boosting (top model) and random forest (second top model), showed the best prediction performance.

8. Conclusion

This study presented a data-driven framework for estimating the WCP (i.e., crew-produced outputs over the crew working hours), which can be used in various operations and project types. This comprehensive, standard, and task-type-independent framework utilized historical data obtained from DWRs to develop productivity estimation models. This process has been conducted in three phases, including (1) data extraction and preprocessing, (2) model preparation, and (3) model training.

DWRs are the major inputs of the system and are comprised of information about each operation, the number and type of resources (crew composition), weather and terrain conditions, and other attributes. The proposed semi-automatic framework is able to search among entire DWRs and extract all the data corresponding to the factors affecting WCP. In order to select the most important and effective features and eliminate the redundant ones, a hybrid feature selection process based on the filter and wrapper method has been implemented. Pearson correlation coefficient, Spearman's rank correlation coefficient, Kendall's rank correlation coefficient, ANOVA F-Value, Mutual information, and F-test are methods that have been utilized to perform the filter method feature selection. By performing the filter methods, features with low correlation to the target (i.e., WCP) are identified. According to the expert judgment, several low-correlated features were effective on the WCP (i.e., candidate features), which are analyzed through the developed scenario-based wrapper method. Feature engineering in the developed framework also contains a step for the concatenation of

Table 9
The results of bin boundaries determination for surveying, routing, canal trenching and welding activities.

Model title (activity title + chosen scenario number)	1st boundary	2nd boundary	3rd boundary	4th boundary	5th boundary	6th boundary
Surveying 1	0.0000	0.0431	0.1297	0.1873	0.3902	1.0000
Routing 3	0.0000	0.1838	0.3729	0.5567	0.7584	1.0000
Canal trenching 2	0.0000	0.1272	0.2618	0.3964	0.6054	1.0000
Welding 7	0.0000	0.0797	0.1842	0.3012	0.4367	1.0000

Table 10
The results of the routing activity's best single models.

ML model title	Execution time (sec)	Accuracy on the training dataset (%)	Accuracy on the test dataset (%)	Average precision (%)	Average recall (%)	F1-Score (%)	MAE	MSE	The chosen combination of hyperparameters
Logistic regression l2	210.72	76.20	75.40	77.90	75.40	76.63	0.277	0.338	{'C': 93.5017, 'max_iter': 615, 'penalty': 'l2', 'solver': 'lbfgs', 'tol': 8.3370e-05}c
Logistic regression l1	36.84	73.80	75.40	78.20	75.40	76.77	0.308	0.431	{'C': 76.5530, 'max_iter': 610, 'penalty': 'l1', 'solver': 'saga', 'tol': 0.0227}
Logistic regression elastic net	39.78	75.20	76.90	78.80	76.90	77.84	0.262	0.323	{'C': 36.0670, 'l1_ratio': 0.6406, 'max_iter': 878, 'penalty': 'elastic net', 'solver': 'saga', 'tol': 0.0082}
Ridge	16.7	68.60	70.80	72.00	70.80	71.39	0.415	0.692	{'tol': 1e-06, 'solver': 'auto', 'normalize': True, 'alpha': 1e-06}
SVM	268.96	98.70	81.50	81.50	81.50	81.50	0.231	0.323	{'C': 37.1423, 'cache_size': 2533, 'coef0': 4.01, 'decision_function_shape': 'ovo', 'degree': 4, 'kernel': 'poly', 'tol': 1e-06}
KNN	40.17	99.50	81.50	81.30	81.50	81.40	0.2	0.277	{'weights': 'distance', 'p': 2, 'n_neighbors': 2, 'metric': 'minkowski', 'leaf_size': 77, 'algorithm': 'kd_tree'}
Decision tree	41.5	94.50	84.60	85.30	84.60	84.95	0.2	0.323	{'splitter': 'random', 'min_weight_fraction_leaf': 0.0, 'min_samples_split': 3, 'max_depth': 275, 'criterion': 'gini'}
(MLP) SGD	13,037.93	94.20	81.50	82.10	81.50	81.80	0.2	0.231	{'activation': 'tanh', 'alpha': 1e-06, 'batch_size': 14, 'learning_rate_init': 0.5072, 'max_iter': 1000, 'momentum': 0, 'power_t': 0.01, 'solver': 'sgd', 'tol': 1e-06}
(MLP) adam	**7513.04**	**97.10**	**86.20**	**86.20**	**86.20**	**86.20**	**0.154**	**0.185**	**{'activation': 'logistic', 'alpha': 1e-06, 'batch_size': 85, 'learning_rate_init': 0.0565, 'max_iter': 1000, 'momentum': 0, 'power_t': 3.01, 'solver': 'adam', 'tol': 1e-06}**

Table 11
The results of the routing activity's best ensemble models (first part).

ML model title	Execution time (sec)	Accuracy on the training dataset (%)	Accuracy on the test dataset (%)	Average precision (%)	Average recall (%)	F1-Score (%)	MAE	MSE	The chosen combination of hyperparameters
RF	556.45	96.90	92.30	92.40	92.30	92.35	0.092	0.123	{'n_estimators': 97, 'min_weight_fraction_leaf': 0.0, 'min_samples_split': 3, 'max_features': 'auto', 'max_depth': 292, 'criterion': 'entropy'}
Bagging KNN	106.6	38.00	24.60	69.90	24.60	36.39	1.308	2.692	{'n_estimators': 200, 'max_samples': 0.01}
Bagging LR l2	1093.41	58.50	53.80	55.50	53.80	54.64	0.662	1.108	{'n_estimators': 750, 'max_samples': 0.01}
Bagging LR l1	1332.22	72.40	73.80	77.80	73.80	75.75	0.323	0.446	{'n_estimators': 10}
Bagging LR Elastic net	2809.38	44.60	30.80	73.90	30.80	43.48	1.231	2.585	{'n_estimators': 200, 'max_samples': 0.01}
Bagging R	389.63	61.00	63.10	69.10	63.10	65.96	0.477	0.723	{'n_estimators': 1000, 'max_samples': 0.01}
Bagging SVC	347.56	82.50	83.10	83.90	83.10	83.50	0.215	0.338	{'n_estimators': 750
Bagging RF	4108.23	71.90	69.20	73.00	69.20	71.05	0.354	0.446	{'n_estimators': 750, 'max_samples': 0.01}
Bagging MLP SGD	2497.93	52.30	58.50	65.90	58.50	61.98	0.692	1.431	{'n_estimators': 500, 'max_samples': 0.01}
Bagging MLP adam	2757.61	62.90	66.20	70.30	66.20	68.19	0.523	0.985	{'n_estimators': 350, 'max_samples': 0.01}
AdaBoost	168.34	57.00	56.90	56.80	56.90	56.85	0.554	0.8	{'n_estimators': 98}
Gradient boosting	523.24	74.60	84.60	85.30	84.60	84.95	0.169	0.2	{'subsample': 0.5, 'n_estimators': 200, 'min_samples_split': 0.4273, 'min_samples_leaf': 0.1, 'max_features': 'sqrt', 'max_depth': 3, 'loss': 'deviance', 'learning_rate': 0.01, 'criterion': 'friedman_mse'}
Histogram-based gradient boosting	**9563.91**	**97.50**	**93.80**	**93.90**	**93.80**	**93.85**	**0.062**	**0.062**	**{'max_iter': 289, 'learning_rate': 0.1, 'l2_regularization': 0}**

similar features, such as various types of dump trucks, bulldozers, side booms, and loaders which have been merged into a unique feature as a whole. Other preprocessing steps, including handling missing values using the KNN algorithm, outlier detection and removal by visual assessment and IQR proximity rule, utilizing noise addition with Gaussian noise technique, and eventually scaling data values using Eq. (1), are all properly performed in the modules of the developed framework. Since the regression problem is transformed into a classification problem, it was necessary to determine bins to categorize the continuous range of data. A heuristic algorithm was developed to determine bin boundaries and split the data points, and assign them to each bin. As some categories suffer from a small amount of data, an oversampling technique is used, which duplicates data points of minor classes to reach an equal amount of data for each class.

Different ML models are trained and evaluated to find the model with the highest accuracy for each activity. A random search technique was

Table 12
The results of the routing activity's best ensemble models (second part).

ML model title	Execution time (sec)	Accuracy on the training dataset (%)	Accuracy on the test dataset (%)	Average precision (%)	Average recall (%)	F1-Score (%)	MAE	MSE
Soft voting	306.41	99.50	89.20	89.80	89.20	89.50	0.123	0.154
Hard voting	354.8	88.30	84.60	84.50	84.60	84.55	0.169	0.2
Stacking LR l2	1299.13	100.00	89.20	89.70	89.20	89.45	0.108	0.123
Stacking LR l1	1298.38	100.00	83.10	82.90	83.10	83.00	0.185	0.215
Stacking LR Elastic net	1303.13	100.00	86.20	86.40	86.20	86.30	0.154	0.185
Stacking R	1296.52	100.00	83.10	83.10	83.10	83.10	0.185	0.215
Stacking SVC	1295.48	100.00	86.20	86.40	86.20	86.30	0.154	0.185
Stacking KNN	1295.95	100.00	86.20	86.40	86.20	86.30	0.154	0.185
Stacking DT	1290.51	100.00	83.10	82.90	83.10	83.00	0.185	0.215
Stacking MLP SGD	**1295.28**	**99.90**	**90.80**	**91.10**	**90.80**	**90.95**	**0.108**	**0.138**
Stacking MLP adam	1296.43	100.00	86.20	86.00	86.20	86.10	0.154	0.185

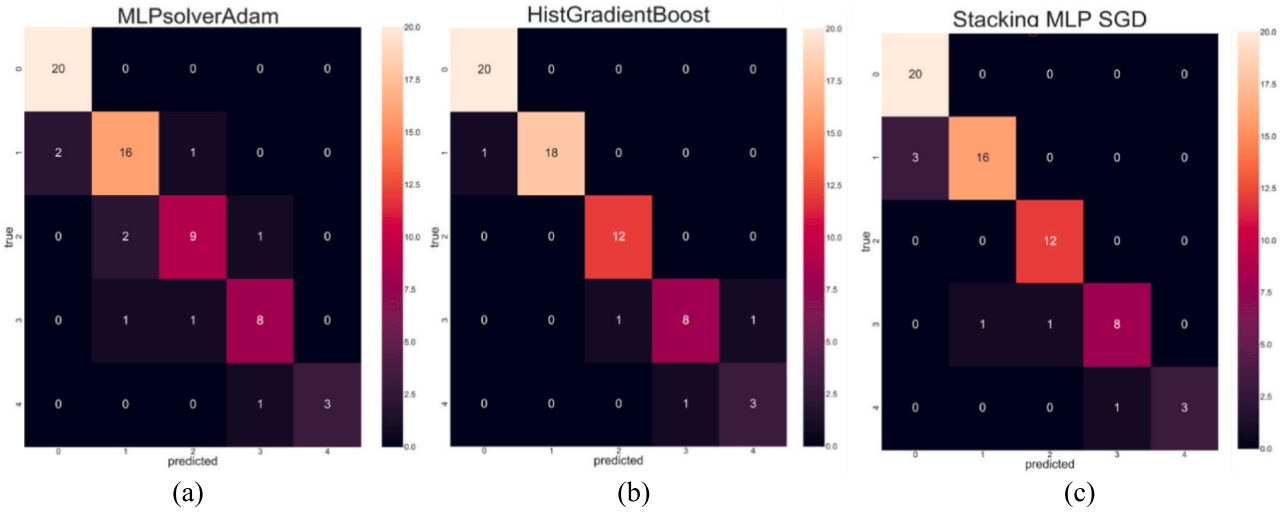

Fig. 14. Confusion matrices of (a) MLP, (b) Histogram-based gradient boosting, and (c) Stacking models for routing activity.

Table 13
The chosen models assigned to each activity.

Model title (activity title + chosen scenario number)	The chosen model type	Model title (activity title + chosen scenario number)	The chosen model type	Model title (activity title + chosen scenario number)	The chosen model type
Routing 3	Histogram-based gradient boosting	Stringing pipes 2	Histogram-based gradient boosting	Coating 2	RF
Surveying 1	Histogram-based gradient boosting	Welding 7	RF	Lowering pipes 2	Stacking MLP adam
Canal trenching 2	RF	Radiography 2	(KNN)	Backfilling 2	RF

utilized to tune the models' hyperparameters to improve the developed models' results. Ensemble ML methods are also implemented to mix and boost the power of single ML models. Finally, all the developed models are evaluated by a k-fold cross-validation procedure to find the model with the best performance. Test results of each model, in addition to structure and hyperparameters, are stored, and these productivity estimation models could be used in future works such as in project planning and scheduling. Since the framework's process flow has developed semi-automatically, civil engineers, project managers, and other project experts with limited knowledge of ML have been able to communicate with the framework and use its capabilities, including data extraction from DWRs, selection of effective factors on WCP, and generation of productivity estimation models.

Potentially, the framework is applicable to any activity in construction projects that collect and use DWRs. In practice, DWRs in some small projects are either not registered, which makes the framework inapplicable, or recorded on paper which requires a significant time and cost to gather the input data. Furthermore, inferior quality of DWRs results in lower accuracy of the machine learning models. The quality of DWRs depends on accurately recording a comprehensive list of factors affecting productivity. Therefore, although the framework can be implemented on projects with low-quality DWRs, it will not be practical due to unfavorable prediction performance.

Although DWRs provide sufficient and massive amounts of data, errors are inevitable. Therefore, ML models are not perfectly precise, leading to some inaccuracies in productivity estimation. ML models work with data as their fuel, so a huge amount of data could enhance their accuracy. In addition, the required time for computations in the framework could be significant because of the numerous models that the system runs. In searching for the best models, the search counter is

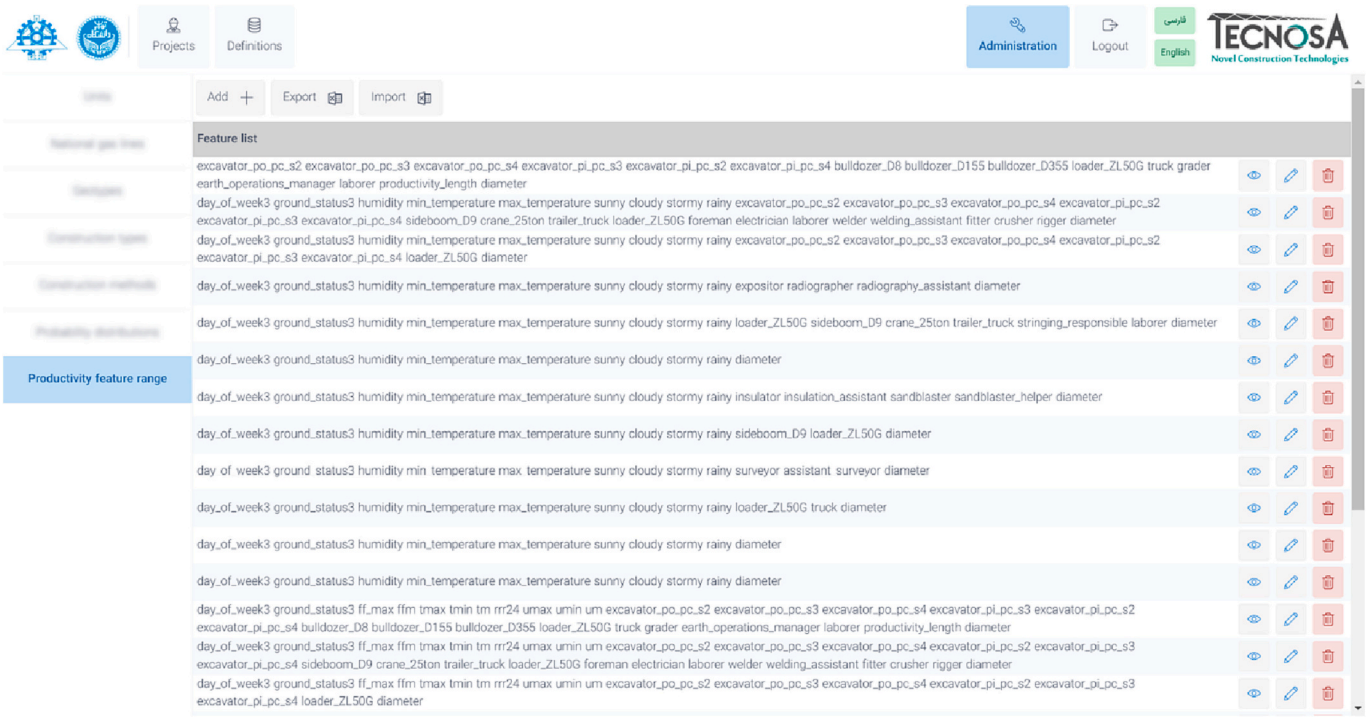

Fig. 15. "Feature range" table in the GUI.

restricted to a specific number to control the time complexity of the system. This restriction decreases the likelihood of reaching a prediction model with the best possible performance (regarding prediction accuracy and computational complexity/simplicity), leading to a limitation in the proposed framework. Furthermore, DWRs contain information, such as weather and terrain conditions, the number of resources, and their composition in a work crew. Other affecting factors, such as labors' age, experience, skill, and unclear instructions to laborers, have not been included in the study. Since the accuracy of the models depends on the completeness of the factors embedded in DWRs, it is suggested to add easily measurable factors (e.g., the experience and age of labor) to DWRs to improve the models' performance.

As a practical recommendation, it is beneficial to standardize the daily work reports' format. If the project management team, contractors, and other stakeholders responsible for providing daily work reports establish a common and standardized format for each type of construction project, the information in the reports will be more valuable. This is because the embedded labels and data will possess a logical structure, thereby facilitating comparison between various projects and effortless analysis of information.

For future research, it is recommended to gather more DWRs and enhance the generated prediction models with a little effort. DWRs can be considered as time-series data. Thus, implementing several deep learning algorithms, such as long short-term memory, enables the framework to deal with sequential data. As it is mentioned earlier, DWRs possibly contain inevitable human errors. Instead of manual data collection, other novel data collection methods, such as sensors and cameras, are useful to overcome the limitation. More specifically, recording activities' progress with cameras and other computer technologies which can extract information from the project environment is another procedure to gather data and reduce human errors. It is worth pointing out that these procedures are costly, unlike prevalent and inexpensive sources (i.e., DWRs), and the economic justification is essential to implement in the system.

Improving the methods of evaluating features influencing the target variable (i.e., WCP) is of great significance. For future works, the developed method can be compared with other common methods (e.g., the wrapper methods) in terms of the performance of models and computational cost.

Finally, if the system handles computational complexities in a reasonable time, using a grid search algorithm instead of a random search would be appropriate to increase the probability of achieving a superior predictive model by examining the hyperparameters' entire search space.

Declaration of Competing Interest

None.

Data availability

Some or all data, models, or code that support the findings of this study are available from the corresponding author upon reasonable request.

Acknowledgment

The support of the Tecnosa R&D center at the University of Tehran to accomplish this research is much appreciated.

References

[1] B.R. Sarker, P.J. Egbelu, T.W. Liao, J. Yu, Planning and design models for construction industry: a critical survey, Autom. Constr. 22 (2012) 123–134, https://doi.org/10.1016/j.autcon.2011.09.011.
[2] B. Vahdani, S.M. Mousavi, M. Mousakhani, H. Hashemi, Time prediction using a neuro-fuzzy model for projects in the construction industry, J. Optim. Ind. Eng. 9 (19) (2016) 97–103, https://doi.org/10.22094/joie.2016.231.
[3] A.N. Tak, H. Taghaddos, A. Mousaei, U.R. Hermann, Evaluating industrial modularization strategies: local vs. overseas fabrication, Autom. Constr. 114 (2020) 103175, https://doi.org/10.1016/j.autcon.2020.103175.
[4] A.A. Tsehayae, Developing and Optimizing Context-Specific and Universal Construction Labour Productivity Models. (PhD Thesis), Department of Civil and Environmental Engineering, University of Alberta, 2015, https://doi.org/10.7939/r3154dt9g.
[5] M. Abdel-Hamid, H. Mohamed Abdelhaleem, Impact of poor labor productivity on construction project cost, Int. J. Constr. Manag. 22 (12) (2022) 2356–2363, https://doi.org/10.1080/15623599.2020.1788757.

[6] M.-Y. Cheng, M.-T. Cao, A.Y.J. Mendrofa, Dynamic feature selection for accurately predicting construction productivity using symbiotic organisms search-optimized least square support vector machine, J. Build. Eng. 35 (2021), 101973, https://doi.org/10.1016/j.jobe.2020.101973.
[7] K.M. El-Gohary, R.F. Aziz, H.A. Abdel-Khalek, Engineering approach using ANN to improve and predict construction labor productivity under different influences, J. Constr. Eng. Manag. 143 (8) (2017), https://doi.org/10.1061/(asce)co.1943-7862.0001340, 04017045.
[8] H.D. Khanh, S.-Y. Kim, N.V. Khoa, N.T. Tu, The relationship between workers' experience and productivity: a case study of brick masonry construction, Int. J. Constr. Manag. (2021) 1–10, https://doi.org/10.1080/15623599.2021.1899593.
[9] J. Dai, P.M. Goodrum, W.F. Maloney, Construction craft workers' perceptions of the factors affecting their productivity, J. Constr. Eng. Manag. 135 (3) (2009) 217–226, https://doi.org/10.1061/(asce)0733-9364(2009)135:3(217).
[10] R. Assaad, I.H. El-adaway, Impact of dynamic workforce and workplace variables on the productivity of the construction industry: new gross construction productivity indicator, J. Manag. Eng. 37 (1) (2021), https://doi.org/10.1061/(asce)me.1943-5479.0000862, 04020092.
[11] M. Hamza, S. Shahid, M.R. Bin Hainin, M.S. Nashwan, Construction labour productivity: review of factors identified, Int. J. Constr. Manag. 22 (3) (2022) 413–425, https://doi.org/10.1080/15623599.2019.1627503.
[12] H.-S. Park, Conceptual framework of construction productivity estimation, KSCE J. Civ. Eng. 10 (5) (2006) 311–317, https://doi.org/10.1007/bf02830084.
[13] L. Song, S.M. AbouRizk, Measuring and modeling labor productivity using historical data, J. Constr. Eng. Manag. 134 (10) (2008) 786–794, https://doi.org/10.1061/(asce)0733-9364(2008)134:10(786).
[14] J. Zhao, N. Zhu, S. Lu, Productivity model in hot and humid environment based on heat tolerance time analysis, Build. Environ. 44 (11) (2009) 2202–2207, https://doi.org/10.1016/j.buildenv.2009.01.003.
[15] A.M. Jarkas, C.G. Bitar, Factors affecting construction labor productivity in Kuwait, J. Constr. Eng. Manag. 138 (7) (2012) 811–820, https://doi.org/10.1061/(asce)co.1943-7862.0000501.
[16] M. Soham, B. Rajiv, Critical factors affecting labour productivity in construction projects: case study of South Gujarat region of India, Int. J. Eng. Adv. Technol. 2 (4) (2013) 583–591. https://www.academia.edu/download/37484099/D1479042413.pdf.
[17] J. Malara, E. Plebankiewicz, M. Juszczyk, Formula for determining the construction workers productivity including environmental factors, Buildings 9 (12) (2019) 240, https://doi.org/10.3390/buildings9120240.
[18] J. Motwani, A. Kumar, M. Novakoski, Measuring construction productivity: a practical approach, Work Study 44 (8) (1995) 18–20, https://doi.org/10.1108/00438029510103310.
[19] A. Woldesenbet, D.H.S. Jeong, G.D. Oberlender, Daily work reports–based production rate estimation for highway projects, J. Constr. Eng. Manag. 138 (4) (2012) 481–490, https://doi.org/10.1061/(asce)co.1943-7862.0000442.
[20] J. Kim, A. Golabchi, S. Han, D.-E. Lee, Manual operation simulation using motion-time analysis toward labor productivity estimation: a case study of concrete pouring operations, Autom. Constr. 126 (2021), 103669, https://doi.org/10.1016/j.autcon.2021.103669.
[21] K.J. Shrestha, H.D. Jeong, Computational algorithm to automate as-built schedule development using digital daily work reports, Autom. Constr. 84 (2017) 315–322, https://doi.org/10.1016/j.autcon.2017.09.008.
[22] H.D. Jeong, D. Gransberg, K.J. Shrestha, Framework for Advanced Daily Work Report System, I.S.U.A. Institute for Transportation, IA, USA, 2015. https://intrans.iastate.edu/app/uploads/2018/03/framework_for_advanced_daily_work_report_system_w_cvr.pdf.
[23] J. Fu, H. Tian, L. Song, M. Li, S. Bai, Q. Ren, Productivity estimation of cutter suction dredger operation through data mining and learning from real-time big data, Eng. Constr. Archit. Manag. 28 (7) (2021) 2023–2041, https://doi.org/10.1108/ecam-05-2020-0357.
[24] M. Saleem, Assessing the load carrying capacity of concrete anchor bolts using non-destructive tests and artificial multilayer neural network, J. Build. Eng. 30 (2020), 101260, https://doi.org/10.1016/j.jobe.2020.101260.
[25] M.H. Momade, S. Shahid, M.R.B. Hainin, M.S. Nashwan, A. Tahir Umar, Modelling labour productivity using SVM and RF: a comparative study on classifiers performance, Int. J. Constr. Manag. 22 (10) (2022) 1924–1934, https://doi.org/10.1080/15623599.2020.1744799.
[26] A. Bolourani, M. Bitaraf, A.N. Tak, Structural health monitoring of harbor caissons using support vector machine and principal component analysis, Structures 33 (2021) 4501–4513, https://doi.org/10.1016/j.istruc.2021.07.032.
[27] K. Nelsia Priya Dharsini, M. Sashikkumar, Probabilistic model development for estimating construction labor productivity optimization integrating with fuzzy logic approach systems, Iran. J. Fuzzy Syst. 17 (6) (2020) 193–201, https://doi.org/10.22111/ijfs.2020.5611.
[28] L. Florez, J.C. Cortissoz, Defining a mathematical function for labor productivity in masonry construction: a case study, Procedia Eng. 164 (2016) 42–48, https://doi.org/10.1016/j.proeng.2016.11.590.
[29] F. Al-Zwainy, A. Eiada, T. Khaleel, Application intelligent predicting technologies in construction productivity, Am. J. Eng. Technol. Manag. 1 (3) (2016) 39–48. https://www.researchgate.net/profile/Faiq-M-Al-Zwainy/project/Neural-Network-in-Project-Management/attachment/59a466454cde26e1c1d7b97e/AS:532403564105728@1503946309283/download/10.11648.j.ajetm.20160103.13_2.pdf.
[30] H.-S. Park, S.R. Thomas, R.L. Tucker, Benchmarking of construction productivity, J. Constr. Eng. Manag. 131 (7) (2005) 772–778, https://doi.org/10.1061/(asce)0733-9364(2005)131:7(772).
[31] R.D. Ellis Jr., S.-H. Lee, Measuring project level productivity on transportation projects, J. Constr. Eng. Manag. 132 (3) (2006) 314–320, https://doi.org/10.1061/(asce)0733-9364(2006)132:3(314).
[32] B.-G. Hwang, C.K. Soh, Trade-level productivity measurement: critical challenges and solutions, J. Constr. Eng. Manag. 139 (11) (2013) 04013013, https://doi.org/10.1061/(asce)co.1943-7862.0000761.
[33] D. Graham, S.D. Smith, Estimating the productivity of cyclic construction operations using case-based reasoning, Adv. Eng. Inform. 18 (1) (2004) 17–28, https://doi.org/10.1016/j.aei.2004.03.001.
[34] F. Mirahadi, T. Zayed, Simulation-based construction productivity forecast using neural-network-driven fuzzy reasoning, Autom. Constr. 65 (2016) 102–115, https://doi.org/10.1016/j.autcon.2015.12.021.
[35] A.T. Gurmu, C.S. Ongkowijoyo, Predicting construction labor productivity based on implementation levels of human resource management practices, J. Constr. Eng. Manag. 146 (3) (2020), https://doi.org/10.1061/(asce)co.1943-7862.0001775, 04019115.
[36] W. Yi, A.P. Chan, Critical review of labor productivity research in construction journals, J. Manag. Eng. 30 (2) (2014) 214–225, https://doi.org/10.1061/(asce)me.1943-5479.0000194.
[37] O. Okyanus, E.L. Oral, M.S. Andaç, Comparison of the performance of K-nearest neighbours and generalized neural network in construction crew productivity prediction, Çukurova Üniversitesi Mühendislik Fakültesi Dergisi 36 (1) (2021) 131–140, https://doi.org/10.21605/cukurovaumfd.933867.
[38] M. Khanzadi, A. Kaveh, M. Alipour, R. Khanmohammadi, Assessment of labor productivity in construction projects using system dynamic approach, Sci. Iran. 24 (6) (2017) 2684–2695, https://doi.org/10.24200/sci.2017.4164.
[39] A.R. Fayek, A. Oduba, Predicting industrial construction labor productivity using fuzzy expert systems, J. Constr. Eng. Manag. 131 (8) (2005) 938–941, https://doi.org/10.1061/(asce)0733-9364(2005)131:8(938).
[40] A.A. Tsehayae, A.R. Fayek, System model for analysing construction labour productivity, Constr. Innov. 16 (2) (2016) 203–228, https://doi.org/10.1108/ci-07-2015-0040.
[41] K. Srinavin, S. Mohamed, Thermal environment and construction workers' productivity: some evidence from Thailand, Build. Environ. 38 (2) (2003) 339–345, https://doi.org/10.1016/S0360-1323(02)00067-7.
[42] M. Bilal, L.O. Oyedele, J. Qadir, K. Munir, S.O. Ajayi, O.O. Akinade, H.A. Owolabi, H.A. Alaka, M. Pasha, Big data in the construction industry: a review of present status, opportunities, and future trends, Adv. Eng. Inform. 30 (3) (2016) 500–521, https://doi.org/10.1016/j.aei.2016.07.001.
[43] Z. You, C. Wu, A framework for data-driven informatization of the construction company, Adv. Eng. Inform. 39 (2019) 269–277, https://doi.org/10.1016/j.aei.2019.02.002.
[44] Y. Pan, L. Zhang, Roles of artificial intelligence in construction engineering and management: a critical review and future trends, Autom. Constr. 122 (2021), 103517, https://doi.org/10.1016/j.autcon.2020.103517.
[45] D. Bzdok, N. Altman, M. Krzywinski, Points of significance: statistics versus machine learning, Nat. Methods 15 (04) (2018) 233–234, https://doi.org/10.1038/nmeth.4642.
[46] T. Han, D. Jiang, Q. Zhao, L. Wang, K. Yin, Comparison of random forest, artificial neural networks and support vector machine for intelligent diagnosis of rotating machinery, Trans. Inst. Meas. Control. 40 (8) (2018) 2681–2693, https://doi.org/10.1177/0142331217708242.
[47] A.S. Ezeldin, L.M. Sharara, Neural networks for estimating the productivity of concreting activities, J. Constr. Eng. Manag. 132 (6) (2006) 650–656, https://doi.org/10.1061/(asce)0733-9364(2006)132:6(650).
[48] M. Kassem, E. Mahamedi, K. Rogage, K. Duffy, J. Huntingdon, Measuring and benchmarking the productivity of excavators in infrastructure projects: a deep neural network approach, Autom. Constr. 124 (2021), 103532, https://doi.org/10.1016/j.autcon.2020.103532.
[49] J. Jeong, J. Jeong, J. Lee, D. Kim, J. Son, Learning-driven construction productivity prediction for prefabricated external insulation wall system, Autom. Constr. 141 (2022), 104441, https://doi.org/10.1016/j.autcon.2022.104441.
[50] F.M.S. Al-Zwainy, H.A. Rasheed, H.F. Ibraheem, Development of the construction productivity estimation model using artificial neural network for finishing works for floors with marble, ARPN J. Eng. Appl. Sci. 7 (6) (2012) 714–722. https://www.researchgate.net/profile/Faiq-M-Al-Zwainy/publication/288719846_Development_of_the_construction_productivity_estimation_model_using_artificial_neural_network_for_finishing_works_for_floors_with_marble/links/56c0626308ae2f498ef94a4d/Development-of-the-construction-productivity-estimation-model-using-artificial-neural-network-for-finishing-works-for-floors-with-marble.pdf.
[51] T. Mahfouz, A productivity decision support system for construction projects through machine learning (ML), in: Proceedings of the CIB W78 2012: 29th International Conference–Beirut, Lebanon, 17–19 October, 2012. https://itc.scix.net/pdfs/w78-2012-Paper-54.pdf.
[52] M. Mady, Prediction Model of Construction Labor Production Rates in Gaza Strip using Artificial Neural Networks. (MSc Thesis) Civil Engineering Department Project Management, The Islamic University of Gaza, 2013. https://search.emarefa.net/en/detail/BIM-531684-prediction-model-of-construction-labor-production-rates-in-g.
[53] F.M.S. Al-Zwainy, M.H. Abdulmajeed, H.S.M. Aljumaily, Using multivariable linear regression technique for modeling productivity construction in Iraq, Open J. Civ. Eng. 3 (3) (2013) 127–135, https://doi.org/10.4236/ojce.2013.33015.
[54] M. Kaya, A.E. Keleş, E.L. Oral, Construction crew productivity prediction by using data mining methods, Procedia Soc. Behav. Sci. 141 (2014) 1249–1253, https://doi.org/10.1016/j.sbspro.2014.05.215.

[55] G. Heravi, E. Eslamdoost, Applying artificial neural networks for measuring and predicting construction-labor productivity, J. Constr. Eng. Manag. 141 (10) (2015) 04015032, https://doi.org/10.1061/(asce)co.1943-7862.0001006.
[56] S.C. Ok, S.K. Sinha, Construction equipment productivity estimation using artificial neural network model, Constr. Manag. Econ. 24 (10) (2006) 1029–1044, https://doi.org/10.1080/01446190600851033.
[57] F. Nasirzadeh, H.D. Kabir, M. Akbari, A. Khosravi, S. Nahavandi, D.G. Carmichael, ANN-based prediction intervals to forecast labour productivity, Eng. Constr. Archit. Manag. 27 (9) (2020) 2335–2351, https://doi.org/10.1108/ecam-08-2019-0406.
[58] S. Golnaraghi, Z. Zangenehmadar, O. Moselhi, S. Alkass, Application of artificial neural network (s) in predicting formwork labour productivity, Adv. Civ. Eng. 2019 (2019) 5972620, https://doi.org/10.1155/2019/5972620.
[59] S. Bai, M. Li, R. Kong, S. Han, H. Li, L. Qin, Data mining approach to construction productivity prediction for cutter suction dredgers, Autom. Constr. 105 (2019), 102833, https://doi.org/10.1016/j.autcon.2019.102833.
[60] H.-Y. Lee, F.-J. Shiue, M.-C. Zheng, Y.-C. Chang, Integrating value estimation and simulation for contractor selection, Autom. Constr. 119 (2020), 103340, https://doi.org/10.1016/j.autcon.2020.103340.
[61] M. Kuhn, K. Johnson, Applied Predictive Modeling vol. 26, Springer, New York, 2013, https://doi.org/10.1007/978-1-4614-6849-3.
[62] H. Liu, L. Yu, Toward integrating feature selection algorithms for classification and clustering, IEEE Trans. Knowl. Data Eng. 17 (4) (2005) 491–502, https://doi.org/10.1109/tkde.2005.66.
[63] S. Yu, W. Tan, C. Zhang, Y. Fang, C. Tang, D. Hu, Research on hybrid feature selection method of power transformer based on fuzzy information entropy, Adv. Eng. Inform. 50 (2021), 101433, https://doi.org/10.1016/j.aei.2021.101433.
[64] Y. Lan, H. Ren, Y. Zhang, H. Yu, X. Zhao, A hybrid feature selection method using both filter and wrapper in mammography CAD, in: 2011 International Conference on Image Analysis and Signal Processing, IEEE, 2011, pp. 378–382, https://doi.org/10.1109/iasp.2011.6109067.
[65] H.-H. Hsu, C.-W. Hsieh, M.-D. Lu, Hybrid feature selection by combining filters and wrappers, Expert Syst. Appl. 38 (7) (2011) 8144–8150, https://doi.org/10.1016/j.eswa.2010.12.156.
[66] S. Muqeem, M.F. Khamidi, A. Idrus, S.B. Zakaria, Development of construction labor productivity estimation model using artificial neural network, in: 2011 National Postgraduate Conference, IEEE, 2011, pp. 1–6, https://doi.org/10.1109/NatPC.2011.6136353.
[67] S.R. Mohammed, A.S. Tofan, Neural networks for estimating the ceramic productivity of walls, J. Eng. Des. 17 (2) (2011) 200–217. https://www.iasj.net/iasj/download/01c66857cb6485cc.
[68] J. Portas, S. AbouRizk, Neural network model for estimating construction productivity, J. Constr. Eng. Manag. 123 (4) (1997) 399–410, https://doi.org/10.1061/(asce)0733-9364(1997)123:4(399).
[69] R. Salman, V. Kecman, Regression as classification, in: 2012 Proceedings of IEEE Southeastcon, IEEE, 2012, pp. 1–6, https://doi.org/10.1109/SECon.2012.6196887.
[70] N.S. Altman, An introduction to kernel and nearest-neighbor nonparametric regression, Am. Stat. 46 (3) (1992) 175–185, https://doi.org/10.1080/00031305.1992.10475879.
[71] C. Cortes, V. Vapnik, Support-vector networks, Mach. Learn. 20 (3) (1995) 273–297, https://doi.org/10.1007/bf00994018.
[72] J. Berkson, Application of the logistic function to bio-assay, J. Am. Stat. Assoc. 39 (227) (1944) 357–365, https://doi.org/10.1080/01621459.1944.10500699.
[73] F. Rosenblatt, Principles of Neurodynamics. Perceptrons and the Theory of Brain Mechanisms. https://safari.ethz.ch/digitaltechnik/spring2018/lib/exe/fetch.php?media=neurodynamics1962rosenblatt.pdf, 1961.
[74] A.E. Hoerl, R.W. Kennard, Ridge regression: biased estimation for nonorthogonal problems, Technometrics 12 (1) (1970) 55–67, https://doi.org/10.1080/00401706.1970.10488634.
[75] E.B. Hunt, J. Marin, P.J. Stone, Experiments in Induction, Academic Press, Oxford, England, 1966. https://psycnet.apa.org/record/1966-08232-000.
[76] T.K. Ho, Random decision forests, in: Proceedings of 3rd International Conference on Document Analysis and Recognition 1, IEEE, 1995, pp. 278–282, https://doi.org/10.1109/icdar.1995.598994.
[77] L. Breiman, Bagging predictors, Mach. Learn. 24 (2) (1996) 123–140, https://doi.org/10.1007/bf00058655.
[78] Y. Freund, R.E. Schapire, A decision-theoretic generalization of on-line learning and an application to boosting, J. Comput. Syst. Sci. 55 (1) (1997) 119–139, https://doi.org/10.1006/jcss.1997.1504.
[79] J. Friedman, Greedy function approximation: a gradient boosting machine, Ann. Stat. 29 (5) (2001) 1189–1232. https://www.jstor.org/stable/2699986.
[80] Z.-H. Zhou, Ensemble Methods: Foundations and Algorithms, CRC Press, Boca Raton, 2012. https://books.google.com/books?hl=en&lr=&id=BDB50Ev2ur4C&oi=fnd&pg=PP1&dq=1.%09Zhou,+Z.-H.,+Ensemble+methods:+foundations+and+algorithms.+2012:+CRC+press.&ots=OyGIBngUOG&sig=G4nThIMKThEhlDcpO7vYlQoOQ9Q.
[81] L. Rokach, Pattern classification using ensemble methods, in: World Scientific 75, 2010. https://books.google.com/books?hl=en&lr=&id=4qnUwdoaVbsC&oi=fnd&pg=PR7&dq=L.+Rokach,+Pattern+classification+using+ensemble+methods.+World+Scientific,+vol.+55,+2010,+ISBN:+978-981-4271-06-6&ots=ioHSWS0TqN&sig=y_8VKthkWnw5VESAxVB-Zkxd1PQ.
[82] D.P. Kingma, J. Ba, Adam: a method for stochastic optimization, in: 3rd International Conference for Learning Representations, San Diego, 2015, https://doi.org/10.48550/arXiv.1412.6980.
[83] L. Gaudette, N. Japkowicz, Evaluation methods for ordinal classification, in: Advances in Artificial Intelligence: 22nd Canadian Conference on Artificial Intelligence, Canadian AI 2009 Kelowna, Canada, May 25–27, 2009 Proceedings 22, Springer, 2009, pp. 207–210, https://doi.org/10.1007/978-3-642-01818-3_25.
[84] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, Scikit-learn: machine learning in Python, J. Mach. Learn. Res. 12 (2011) 2825–2830. https://www.jmlr.org/papers/volume12/pedregosa11a/pedregosa11a.pdf?ref=https:/.
[85] J. Reback, W. McKinney, J. Van Den Bossche, T. Augspurger, P. Cloud, A. Klein, S. Hawkins, M. Roeschke, J. Tratner, C. She, Pandas-Dev/Pandas: Pandas 1.0. 5, Zenodo, 2020, https://doi.org/10.5281/zenodo.3898987.
[86] G. Lemaître, F. Nogueira, C.K. Aridas, Imbalanced-learn: a python toolbox to tackle the curse of imbalanced datasets in machine learning, J. Mach. Learn. Res. 18 (1) (2017) 559–563. https://www.jmlr.org/papers/volume18/16-365/16-365.pdf.
[87] P. Virtanen, R. Gommers, T.E. Oliphant, M. Haberland, T. Reddy, D. Cournapeau, E. Burovski, P. Peterson, W. Weckesser, J. Bright, SciPy 1.0: fundamental algorithms for scientific computing in Python, Nat. Methods 17 (3) (2020) 261–272, https://doi.org/10.1038/s41592-019-0686-2.
[88] R.S. Olson, W.L. Cava, Z. Mustahsan, A. Varik, J.H. Moore, Data-driven advice for applying machine learning to bioinformatics problems, in: Pacific Symposium on Biocomputing 2018: Proceedings of the Pacific Symposium, World Scientific, 2018, pp. 192–203, https://doi.org/10.1142/9789813235533_0018.