www.itcon.org - Journal of Information Technology in Construction - ISSN 1874-4753

PROCESS TIME ESTIMATION FOR WORKSTATIONS IN MODULAR CONSTRUCTION PRODUCTION LINE

SUBMITTED: March 2025
REVISED: May 2025
PUBLISHED: June 2025
EDITOR: Frédéric Bosché
DOI: 10.36680/j.itcon.2025.044

Angat Pal Singh Bhatia, Ph.D., Research Associate
Department of Building, Civil and Environmental Engineering, Concordia University, Montréal, QC, Canada
angatpalsingh.bhatia@concordia.ca

Osama Moselhi, Ph.D., Professor
Centre for Innovation in Construction and Infrastructure Engineering and Management (CICIEM), Department of Building, Civil and Environmental Engineering, Concordia University, Montréal, QC, Canada
moselhi@encs.concordia.ca

SangHyeok Han, Ph.D., Associate Professor
Centre for Innovation in Construction and Infrastructure Engineering and Management (CICIEM), Department of Building, Civil and Environmental Engineering, Concordia University, Montréal, QC, Canada
sanghyeok.han@concordia.ca

SUMMARY: *Modular construction companies produce module components following a make-to-order process to realize client-defined customization requirements. This customization leads to varying process times of prefabricating module components at workstations, making it difficult for production line managers to accurately predict their process times for planning purposes. To address these challenges, this paper proposes a novel method that employs Deep Neural Networks, artificial neural networks, and multiple linear regression models for predicting workstation production process times at a module prefabrication plant. A Genetic Algorithm is employed to refine the structure of the Deep Neural Networks and find a near-optimum number of hyperparameters. In a case study, a wood-based wall panel production line is analyzed to demonstrate the use of the developed method and test its performance. The developed method for process time prediction is found to achieve a mean absolute error of less than 2.50 min for most workstations, with the symmetric mean absolute percentage error ranging between 22% and 28%. The research contributions of this study include the development of prediction models for all the workstations of the production line and the implementation of a Genetic Algorithm to find the near-optimal hyperparameters of Deep Neural Networks. This assists production managers in making data drive decisions and overcomes the reliance on experience-based methods for estimating process times and creating production plans.*

KEYWORDS: *Modular Construction, Production Line, Prediction Model, Deep Neural Network (DNN), Genetic Algorithm (GA).*

REFERENCE: *Angat Pal Singh Bhatia, Osama Moselhi & SangHyeok Han (2025). Process time estimation for workstations in modular construction production line. Journal of Information Technology in Construction (ITcon), Vol. 30, pg. 1080-1098, DOI: 10.36680/j.itcon.2025.044*

1. INTRODUCTION

According to the CSA Public Policy Centre (Dragicevic and Riaz, 2024), Canada requires 4.3 million affordable homes for low-income individuals and more long-term care homes (or similar facilities) to accommodate its growing population. In this regard, modular construction is growing in popularity as a promising alternative for rapid construction of affordable housing units. Moreover, a report on the modular construction market in North America stated that the market is expected to reach $ 80 billion by 2031, driven by its cost- saving benefits in construction projects (Assaad et al., 2022). Modular construction is a process of prefabricating modules or module components (e.g., two-dimensional wall, floor, and roof panels)—often following a mass production approach—within a controlled factory environment. These are then transported to the site to be installed as components of a building. Modular construction manufacturing (MCM) follows standardized operating procedures (SOPs) at each workstation in order to streamline the production of these module components. This standardization allows for innovative technologies such as automation and robotics to be implemented, thereby reducing both process time and cost. Modular construction offers opportunities to: (i) improve productivity by usually moving tasks to workers at fixed workstations rather than having workers move to tasks in production line environment (Feld 2000), which is the common practice in traditional construction (e.g. wall panels are transferred from the framing station to the sheathing station, while workers remain at their designated station to perform repetitive tasks in line with the sheathing workstation standard operating procedures (SOPs); (ii) minimize material waste and actively support the realization of a circular economy; and (iii) achieve faster return on investment, since the modular construction process allows to reduce the construction cycle time such that developers can begin making profits sooner (Mazdak et al., 2021). However, client-defined customization introduces variation in design specifications (e.g., wall panels with/without doors and windows, different dimensions and number of studs), leading to varying production rates, imbalanced production lines, and increased makespan (i.e., total completion time) (Bhatia et al., 2023).

These variations in design specifications make it difficult to accurately predict and estimate process times (Bhatia et al., 2022). Process time (i.e., the time it takes to process one module component at a given workstation), a critical performance metric in MCM, must be accurately forecasted in order to: (i) manage hourly and daily production line operations; (ii) gain insights into underlying patterns in production; and (iii) make data-driven decisions regarding resource planning (e.g., labor deployment) and scheduling of sequences of operations along the production line. Additionally, better process time estimation reduces idle time of workers and minimizes production cost. It should be noted that, for the purpose of the present study, “module” refers to a volumetric module, while “module component” refers to wall, floor, and roof panels, whereas “modular” refers to a construction method that involves using individual modules as the basis for construction. In practice, production managers often rely on their experience and historical average process times (e.g., linear meters of wall panels per minute) in planning and scheduling the sequences of module components. However, average process time does not accurately reflect actual production conditions. Alsakka et al. (2023) showed that the use of average rates leads to overly optimistic cycle time predictions. These unrealistic estimates, in turn, put undue pressure on workers and management to meet unattainable schedules and ultimately contribute to delays in the production timeline. The dynamic nature of MCM, which requires frequent adjustments, makes resource planning and scheduling in the absence of accurate process times particularly challenging. To avoid subjectivity, researchers have used probabilistic techniques to model the uncertainty associated with process times at workstations in modular fabrication facilities. For instance, Altaf et al. (2018) used probability distributions to forecast the process times of wall panels at workstations. Khandar et al. (2021), meanwhile, assumed that the process time of workstations on a production line follow a triangular distribution. However, these probability distributions can yield misleading results, as they do not account for variations in design specifications arising from client-defined customization.

The primary goal of the present study is to address these limitations through the development of a predictive method for accurately forecasting process times at each workstation of the in MCM production line using Deep Neural Network (DNN) combined with Genetic Algorithm (GA). The proposed method encompasses: (i) the development of predictive models that combine DNN, artificial neural network (ANN), and multiple linear regression (MLR) to predict the process times of module components at workstations; (ii) the use of GA for identifying near-optimal hyperparameters (i.e., number of hidden layers, neurons, momentum and learning rate) for the DNN; and (iii) comparison of the prediction performance of DNN with that of ANN and MLR.

Table 1: Research Gaps.

Number	Author and Year	Method	Gaps
1	Alsakka et al. (2024)	Used computer vision based data to develop cycle time prediction framework for modular construction production line. Various machine-learning models, including ANN were trained to estimate the cycle time.	• Prediction models have not been created for all workstations along the production line. This does not ensure better project planning. • Manual tuning was employed to determine the optimal parameters for the neural networks. This method introduces subjectivity into the parameter tuning process and does not guarantee optimal neural network configuration. • The dataset used to train the prediction models covers only a shorter period of operation. This may not yield better prediction performance for projects with complex designs and larger datasets.
2	Mohsen et al. (2022)	Predicted the cycle time of wall panels in a production line using ML models and using historical RFID data.	• Prediction models are not developed for every workstation along the production line. • Manual tuning was used to identify the parameters for the neural networks. This approach introduces subjectivity into the tuning process and does not ensure optimal parameters.
3	Bhatia et al. (2022)	Applied MLR models to predict the process times of wall panels at each workstation of the production line.	• The dataset used to train the model was small (i.e., 200), which may not yield better prediction performance for projects with complex designs and larger datasets.
4	Altaf et al. (2018)	Predicted wall panel process time using probability distribution functions (e.g., beta, triangular and gamma distributions).	• Probability distributions can yield misleading results as they do not consider design variations arising from clients' customization requirements, which are unique to module components when predicting production durations.
5	Taiwo et al. (2022)	Developed multiple linear regression model to predict the productivity of installing modules in modular integrated construction projects.	• The regression results are not compared with other machine-learning algorithms. • The dataset used to train the prediction models covers only a shorter period of operation. This may not yield better prediction performance for projects with complex designs and larger datasets.
6	Khandar et al. (2020)	Predicted duration at each workstation using triangular distribution with 15% upper and lower bound.	• Probability distributions can yield misleading results as they do not consider design variations arising from clients' customization requirements, which are unique to module components when predicting production durations.
7	Moon et al. (2023)	Developed multilayer perceptron artifical neural network to predict the production days for the cable manufacturing production line.	• Manual tuning was employed to determine the optimal parameters for the neural networks. This method introduces subjectivity into the parameter tuning process of neural networks and does not guarantee optimal parameters.
8	Aghajamali et al. (2024)	Applied ANN to predict cycle time of workations at steel fabrication production line.	• Manual tuning was employed to determine the optimal parameters for the neural networks. This method introduces subjectivity into the parameter tuning process of neural networks and does not guarantee optimal parameters. • The dataset used to train the model was small, which may not yield better prediction performance for projects with complex designs and larger datasets.
9	Mohsen et al. (2023)	Developed a data-driven machine-learning model using Random forest and gradient boosted decision trees to predict the total duration of prefabricating pipe spool.	• Manual tuning was employed to determine the optimal parameters for the neural networks. This method introduces subjectivity into the parameter tuning process of neural networks and does not guarantee optimal parameters.

2. LITERATURE REVIEW

A modular construction production line must operate at its full capacity in order to maximize productivity and meet on-site demands. However, due to the highly customized nature of project orders, with multiple orders in progress in the plant at any given time, historical data on process times of module components across workstations is unevenly distributed, creating uncertainty in process time estimation (Bhatia et al., 2025). Given that the complexity of predicting process times increases with customization, an accurate prediction model that would allow production line managers to make real-time adjustments to production plans is needed (Taiwo et al., 2022). Accurate prediction of process times would also aid production managers in efficiently allocating resources and reducing waiting times of module components between workstations. However, in modular construction production lines, the amount and nature of the data are crucial factors in training the prediction model. For instance, if the collected data only includes working periods when experienced workers are operating, the prediction models are likely to perform differently when new workers are involved. Similarly, if the collected data is from a project with simple design factors, the prediction model may not perform well for projects with complex designs (Mohsen et al., 2023). In this regard, to sustain and enhance the performance of the prediction model, a large set of data covering an extended period of time is required. This can be achieved through automated data acquisition (e.g., RFID system), which has shown promise in the context of modular construction (Altaf et al., 2018). Given the promising performance of RFID system in automated data acquisition, further investigation is needed into the performance of prediction models trained using RFID datasets in estimating process times.

MLR models have been widely used in the development of data-driven prediction metrics in various construction, infrastructure, and fabrication applications (Chu et al., 2020; Omar and Moselhi 2022). The MLR model assumes a linear relationship between dependent and independent variables, which is crucial for achieving accurate results (Mohsenijam et al., 2017). In construction management, MLR has been employed in applications such as predicting durations of earthmoving operations (Smith, 1999) and forecasting cycle times for one-span installations in precast bridges (Mohsenijam et al., 2017). Additionally, there have been various applications of MLR in production line settings. For instance, Bhatia et al. (2022) applied MLR models to predict process times of wall panels at workstations, achieving an R^2 of over 70% with a small dataset (i.e., 200 data points). However, MLR works with the underlying assumption that there is a linear relationship between the independent variables and the dependent variable (e.g., process time), and this limits its performance when the relationship is non-linear. Other limitations of MLR become evident when it is applied to more complex manufacturing systems (including MCM). For instance, in the case of aircraft assembly, which is a non-linear system, regression has not provided satisfactory predictions of productivity (Mattsson, 2017).

In this context, the application of ANN to enhance the accuracy of prediction models has been explored in numerous studies. For instance, ANN has been used in engineering operations and management in the development of predictive models for decision support (Moon et al., 2023). ANNs are particularly advantageous because they: (i) perform well for highly uncertain, nonlinear, and complex problems; and (ii) are effective for predictions in nonlinear manufacturing and supply chain settings (Moon et al., 2023; Ambrogio and Gagliardi 2013). Moselhi et al. (1991) identified ANNs as the most commonly used supervised learning method for analyzing` the relationship between inputs and outputs. This model encompasses a collection of processing elements, usually arranged into layers (i.e., input, hidden, and output). The target is to predict one or more dependent variables based on independent variables. The input layer accepts the data (i.e., independent variables), which the hidden layer uses to represent relationships, and the output layer then generates the network response (i.e., dependent variable). ANN is defined as a neural network that has one hidden layer, whereas DNN is characterized by having two or more hidden layers (Aggarwal 2018). The "deep" refers to the presence of multiple hidden layers, enabling the network to learn complex representations from the input data. ANNs have been extensively applied in the construction and manufacturing industries. For example, Ambrogio and Gagliardi (2013) applied ANN to predict the performance of a manufacturing production line, achieving improved prediction accuracy. Basma and Moselhi (2022) used ANN to predict the duration and cost of highway projects, achieving mean absolute percentage error (MAPE) of 7.4% and 4.5%, respectively. El-Sawah and Moselhi (2014) applied and compared different neural network models (back propagation neural network, probabilistic neural network, generalized regression network), and regression analysis in order to estimate the costs at the pre-design stage for structural steel buildings and short-span timber bridges. Their results showed that the probabilistic neural network outperformed the regression method, with a MAPE of 1.91%. Moon et al. (2023) successfully implemented a multilayer perceptron ANN to predict production and latency days for manufacturing production facilities. Pannakkong et al. (2022) implemented ANN for predicting the quality of the raw materials in the production process. RSM (Response Surface Methodology) was utilized for finding optimal hyperparameters (learning rate and number of hidden layers and neurons). Huang and Chang (2024) stochastic gradient descent with momentum (SGDM) was used for hyperparameter tuning (number of hidden layers and neurons) of ANN. The objective was to predict the quality of injection-moulded parts. Schneckenreither et al. (2021) developed ANN based prediction model to estimate flow time for a make-to-order flow-shop.

Studies such as those by Mohsen et al. (2022) and Alsakka et al. (2024) have advanced the development of workstation process time prediction in MCM using various machine-learning models, including ANN. Similarly, Aghajamali et al. (2024) used data augmentation techniques and machine-learning techniques to estimate productivity for a fitting workstation in a steel production factory. However, these prediction models: (i) have targeted individual workstations rather than encompassing all workstations in the production line to ensure improved project planning and (ii) have used manual tuning and GridSearchCV to find the hyperparameters for the neural networks. Such an approach introduces subjectivity in the process of tuning the hyperparameters (i.e., does not guarantee optimal hyperparameters), increases the model prediction error, and entails a high computational burden for finding the hyperparameters (Callens et al., 2020).

Various studies in construction and infrastructure engineering contexts have used GA for hyperparameter tuning of machine-learning models. For example, Assad and Bouferguene (2022) employed GA to enhance the accuracy of water main condition prediction by optimizing the parameters of various data-mining techniques (e.g., DNN and decision tree). Koc et al. (2021) applied tree-based (e.g., Random Forest) machine-learning models to predict the post-accident disability status of workers in the construction industry. In their study, the machine-learning parameters were tuned using GA in order to improve the prediction accuracy. However, this optimization approach omitted key hyperparameters such as learning rate and momentum, which inclusion could enhance convergence efficiency and reduce overall training time. Given the successful use of GA for tuning the parameters of machine-learning models in literature, the present study implements GA to determine the near-optimal parameters for a neural network model for predicting process times of module components in a production line. It is worth noting that this hyperparameter tuning problem belongs to the class of non-deterministic polynomial-time hard (NP-hard) problem. In such problems, there are multiple combinations (number of hidden layers, neurons, learning rate and momentum) to train the model, and, with an increase in the number of combinations, the complexity of the problem (i.e., the search space) increases, making it increasingly difficult for existing techniques to effectively find the near optimal combination within a reasonable model runtime. In this regard, GA is utilized to find near-optimal parameters for a DNN in reasonable run time.

In summary, as shown in Table 1, existing methods for predicting process times of module components in MCM exhibit the following limitations: (i) in these studies, manual tuning has been employed to determine the optimal parameters for the neural networks, introducing subjectivity into the parameter tuning process and yielding sub-optimal parameters; (ii) the prediction models have focused on one or a few workstations rather than encompassing all workstations along the production line; (iii) the dataset used to train the model has been small, meaning that the model may not achieve acceptable prediction performance for projects with complex designs and larger datasets; and (iv) the probability distributions may produce misleading results, as they do not account for design variations stemming from clients' customization requirements (which is an important consideration when predicting production durations in MCM). To address these limitations, this study developed a predictive method for accurately forecasting process times at each workstation of the MCM production line rather than focusing on few workstations within the production line using Deep Neural Network (DNN) combined with Genetic Algorithm (GA).

3. DEVELOPED METHOD

Figure 1 presents the main components of the developed method for predicting process times in MCM production lines. The method consists of three modules. (i) The first is the data input module, designed to facilitate and organize the input data pertinent to the unique characteristics of each module component for fabrication. It includes data descriptions that specify the module component design parameters (e.g., number of studs and doors/windows) and timestamps (i.e., start and finish times) from the collected RFID data. (ii) The second is the data pre-processing module. In this module, the captured data is cleaned and prepared. This entails combining attributes that have similar properties (e.g., windows + large windows are combined as ‘windows’) and identifying and removing outliers. (iii) The third is the time prediction module, which houses three tools for time prediction (ANN, DNN and MLR) and which enables comparative study to select the most suitable tool for predicting fabrication process times. Subsequently, GA is also used, in conjunction with DNN, to fine tune the architecture of the network (i.e., number of hidden layers and neurons, known also as "processing elements") and training parameters (momentum and learning rate). It is worth mentioning that in this paper hyperparameter optimization focuses on tuning both training-related parameters and structural design of the neural network. Additionally, a base ANN architecture consists of one hidden layer and five neurons. A trial-and-error method is used to determine the number of neurons in the single hidden layer of the ANN. The number of neurons is varied (1, 2, 3, 4, 5, and 6) with 100 epochs, and the resulting output—i.e., mean absolute error (MAE)—is monitored. This procedure is repeated for each neuron count, and the network configuration (number of hidden layers and number of neurons) with the lowest error rate is chosen.

Additionally, MLR is employed in the present study due to its successful application in previous studies for predicting process times and due the ease with which its results can be interpreted. In MLR, a linear relationship is established between the independent variables and the dependent variable. This relationship is a combination of the independent variables with predetermined weights and a bias coefficient. This relationship, for the context of the present study, is expressed in Equation 1 (Hasan and Lu, 2022).

$$Y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \ldots\ldots\ldots\ldots + \beta_k x_k \tag{1}$$

where Y is the dependent variable (process time); X_k represents the independent variables (number of studs and doors/windows); β_0 is the bias value; and β_k represents the weighting coefficients for the independent variables.

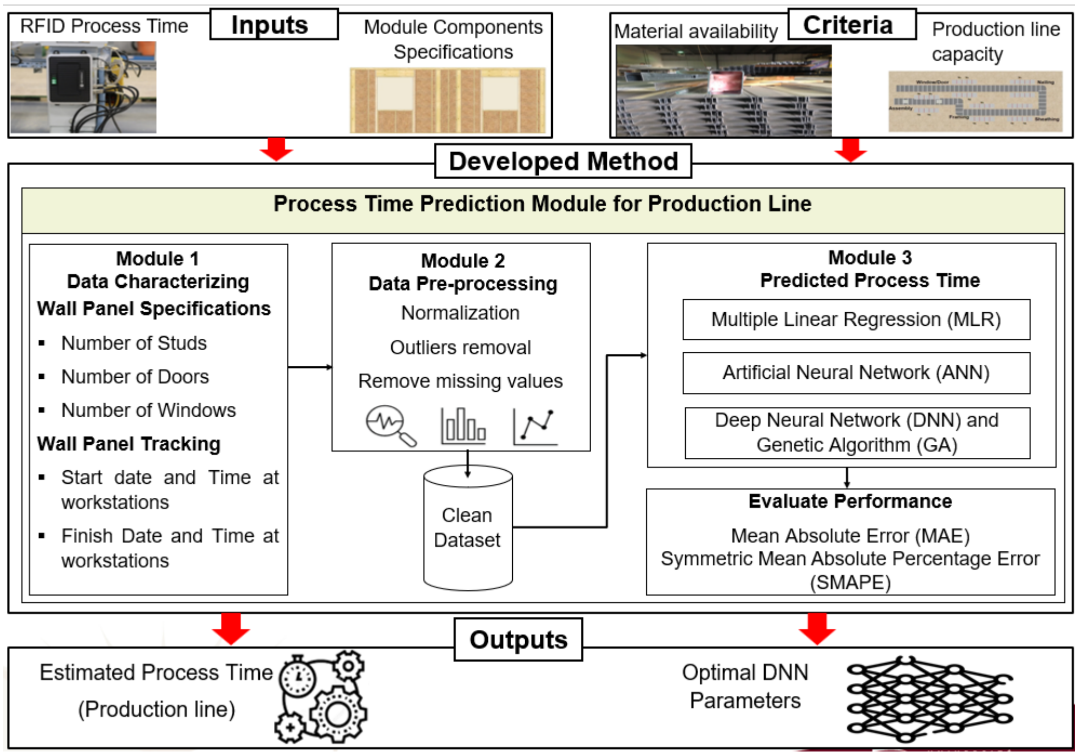


Figure 1: Overview of the developed method (Images courtesy of Mohammed Sadiq Altaf).

The data used in this study spans the period from July, 2015, to August, 2018. The data was given by case study company and this is the reason the authors are using the RFID data for this specific time period. The data is captured using an RFID system, with passive RFID tags attached to each module component. In total, 416,950 timestamps (i.e., start and finish times) for module components moving through the production line are gathered. The captured dataset, after pre-processing, is randomly divided into training (80%) and testing (20%) subsets. The criteria in the developed method include the capacity of workstations, defined by the maximum module component length they can accommodate (e.g., the capacity of a framing workstation may be a length of 40 ft).

3.1 Data characterizing and data-preprocessing

Data characterization is carried out in two main steps: (i) gaining understanding of the production line (e.g., identifying which workstations are fully automated and which are manual), and (ii) extracting process times. In the first step, the SOPs of each workstation are reviewed to gain a deeper understanding of the production line. Additionally, shop drawings are reviewed in order to classify module components (e.g., the number of studs and windows in interior and exterior wall panels). In the second step, before extracting process times, breakpoints (i.e., the start and finish points for the various work processes at the workstations) are identified in order to determine which RFID antennas and readers correspond with each workstation. The RFID system, described in an earlier study (Altaf et al., 2018), is used to collect the production line data to be employed in developing the prediction models. Although, a complete discussion of the RFID data collection process is beyond the scope of this paper, a brief description is given to explain the components regarding RFID system used for data collection. The system consists of: (i) an RFID printer, which generates the ID tag numbers for module components; (ii) RFID tags, which are attached to each module component; (iii) RFID antennas, which are installed at each workstation (i.e., read zone) and which detect the tag signals, capturing the movement of module components along the production line; and (iv) RFID readers, to which these antennas are connected and which transfer the timestamps (i.e., initial and last read time) into the database. The next task in this step is to extract the process times and relevant attributes of the module components (e.g., number of studs and doors) at each workstation from the RFID raw data file provided by the case company. It is worth mentioning that the authors received RFID data from the case study company, and for most workstations (e.g., framing, sheathing, and nailing), the waiting time of the wall panels was removed by the industrial partner using the RANSAC algorithm. The process times (i.e., the time required to complete one module component at a given workstation) is extracted using Equation 2 (Mohsen et al. 2022):

$$PT_{m,w} = IRT_{m,w+1} - IRT_{m,w} \tag{2}$$

where PT is the process time of module component ID “m” at current workstation “w”, “w+1” is the next workstation, and IRT is the initial read time of the module component at a given workstation of the production line.

The next task in this step is to perform data pre-processing and ensure that the dataset is clean for prediction purposes. Data pre-processing includes: (i) discarding records with missing values; (ii) removing outliers using data visualization techniques (i.e., pie charts) and “Mean ± 1.5 SD” (i.e., data points that are above the upper threshold or below the lower threshold are considered statistical outliers); (iii) combining attributes with similar properties (e.g., doors + large doors are combined as “Doors”, and different types of studs, such as studs + Dstud + Lstud + Mstud, are combined as “Studs”); and (iv) data normalization is implemented using Equation 3.

$$X' = \frac{(X - X_{min})}{(X_{max} - X_{min})} \tag{3}$$

where X_{min} is the minimum value and X_{max} is the maximum value of the independent variable, and X is the original value.

3.2 Prediction of process time

Figure 2 presents a flowchart of the proposed model for predicting production time, which combines DNN and GA to predict the process times of module components (e.g., wall panels) at workstations using historical data. As noted above, the term “deep” signifies that there are multiple hidden layers, such that the network can learn from the input data. Part of the rationale for employing a DNN is that its multiple hidden layers allow it to efficiently perform complex non-linear transformations. Each layer consists of several neurons representing the input, transfer, and output variables.

The dataset, having been pre-processed, is divided into training (80%) and testing (20%) subsets and, based on the training subset, a prediction model is developed. The ReLu (Rectified linear unit) activation function is selected for this purpose, with the search range defined as [3,10] for hidden layers and [6,100] for number of neurons. Moreover, cross-validation is carried out to prevent overfitting and to ensure robust evaluation of the predictive model. In specific, k-fold cross-validation is adopted for testing the performance of a model due to its ability to reduce variance in model evaluation and better handling of class imbalance. The dataset is divided into k groups, where the predictive model is trained using (k−1) groups, and the remaining fold is used to test the accuracy of the model. The process is then iteratively repeated, holding a different group for validation and using the remaining ones for training. The overall performance is then measured as the average performance of each iteration. The parameters of the prediction algorithms are automatically tuned based on an optimization procedure in order to minimize the MAE resulting from the 10-fold cross-validation. It is worth noting that bootstrapping, is not used as it involves creating many resamples (e.g., 1,000 or more) and calculating the statistic for each one can be computationally expensive, especially with large datasets.

It should be noted that the parameters of the model (i.e., the number of hidden layers and the number of neurons) have a significant impact on the model's performance. For instance, if there is a small number of neurons, the model cannot be trained well. With a large number of neurons, on the other hand, performance can be enhanced, but the large number of connections will increase the computation time. It is thus critical to establish parameters that allow for the network to be trained within a reasonable computation time while yielding an error that is within the tolerance limit. For this purpose, hyperparameter tuning—the process of identifying optimum parameters (i.e., number of hidden layers, number of neurons, learning rate and momentum)—is performed. GA optimization is employed for this task due to its efficiency in finding near-optimal solutions to prediction problem. The optimization objective is to minimize the model's prediction error (i.e., MAE), as expressed in Equation 4.

$$MAE = \frac{\sum_{i=1}^{n} |A_i - P_i|}{n} \quad (4)$$

where Ai is the actual process time of a module component; and Pi is the predicted process time of a module component.

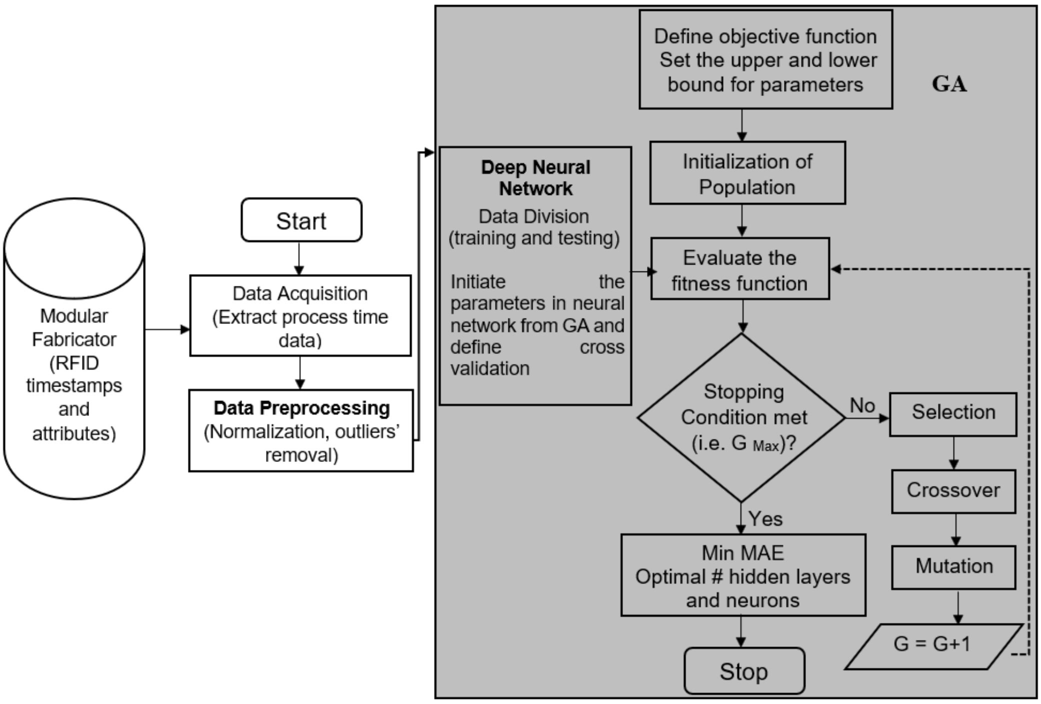


Figure 2: Flowchart of Deep Neural Network optimization.

Moreover, error percentage is calculated using the symmetric mean absolute percentage error (SMAPE), which is a modified MAPE and which is obtained by dividing the absolute error by the average of the actual observation and the predicted process time, as shown in Equation 5. According to Makridakis (1993), MAPE is asymmetric, as it imposes a greater penalty on predictions that exceed the actual values than on those that fall short (i.e., a small actual value in the denominator leads to a very high percentage error). SMAPE addresses this limitation by treating overestimation and underestimation errors more equally. This is achieved by incorporating both actual and predicted values in the denominator, ensuring a more balanced evaluation of over-predictions and under-predictions.

$$SMAPE = \frac{1}{N} \sum_{i=1}^{N} \left| \frac{A_i - p_i}{(A_i + p_i)/2} \right| \quad (5)$$

The optimization process is carried out in four steps. Step 1 is to create an initial population, which consists of chromosomes representing various configurations of the number of neurons, number of hidden layers, learning rate, and momentum. This initial population is generated randomly, with the number of chromosomes determined based on the population size. Each chromosome contains multiple genes, each representing a specific aspect of the DNN configuration (i.e., number of neurons, number of hidden layers, learning rate, and momentum). Notably, the number of neurons and the number of hidden layers is represented as discrete values, chosen randomly from the combinatorial range of their upper and lower bounds. Learning rate and momentum, on the other hand, are represented as continuous values. In the present study, neurons have a step size of 2, and hidden layers have a step size of 1. Please note that the step size refers to the interval by which a parameter is increased during tuning; here, neurons are adjusted with a step size of 2, meaning the neuron count is increased by 2 each time, while hidden layers are adjusted with a step size of 1. Figure 3(a) shows the structure of a generated chromosome, which represents a potential solution. For example, in the chromosome "8,4,0,1,1,1,1,0,0,1", "8" is the number of neurons in each hidden layer (a discrete value), "4" is the number of hidden layers (also a discrete value), and the binary values that follow represent the learning rate and momentum (continuous values). In this chromosome, the first four binary digits represent the learning rate, while the latter four binary digits represent momentum. These binary strings are decoded using Equation 6 and 7 in order to convert the encoded parameters back into real-world values.

$$\mathrm{D} = (\Sigma bit \times 2^{i}) \times P + \alpha \tag{6}$$

$$\mathrm{P} = \frac{b-a}{2^{l}-1} \tag{7}$$

where D is the decoding process, P is the parameter (see Equation 6 and 7), variables a and b are the lower and upper bounds of the parameter, and l is the length of the chromosome. For instance, to decode the variable learning rate, which has lower (a) and upper (b) bounds of 0.01 and 0.30, respectively, and a chromosome length of 4 (Figure 3b), the decoded value (D) is 0.15, which falls within the acceptable range (i.e., 0.01 to 0.30).

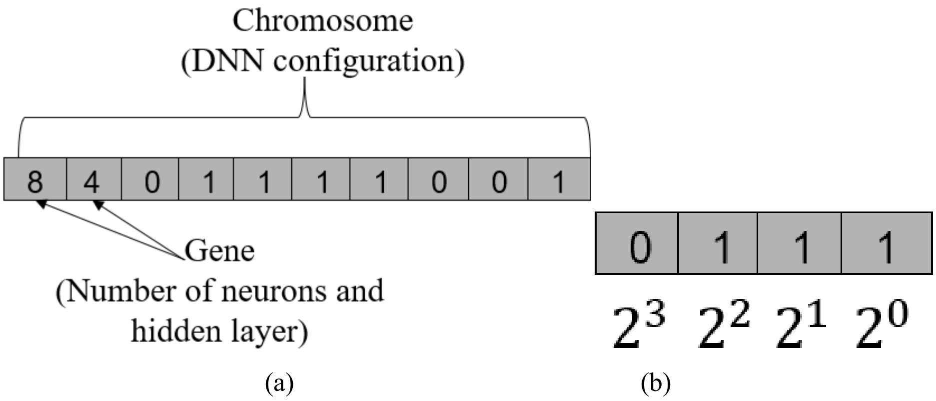


Figure 3: (a) Structure of the chromosome; and (b) decoding of chromosome (example of learning rate parameter).

Step 2 of the optimization process is evaluation, in which the fitness values of each chromosome are evaluated based on the optimization criteria (i.e., minimum error rate). Step 3, meanwhile, is the selection process, which involves choosing the best chromosomes (i.e., potential parents) from the population as the best solution (i.e., the best combination of number of hidden layers, number of neurons, learning rate, and momentum) based on fitness value. The tournament selection technique is used for this purpose, this technique being chosen due to its efficiency, ease of implementation, and relatively low computation time compared to other methods (e.g., roulette wheel selection) (Razali and Geraghty, 2011). In the tournament selection technique, a number of chromosomes (represented as k, i.e., the tournament size) are randomly selected from the population, and the chromosome with the best fitness score among the k chromosomes is selected as parent 1. This process is repeated until the desired number of parents for crossover is selected. For example, in a population of 10 with a tournament size (k) of 3, chromosomes 1, 2, and 3, with fitness values of 0.75 min, 0.82 min, and 0.68 min, respectively, are randomly picked. The chromosome with the best fitness (i.e., minimum error rate)—in this case, chromosome 3 (fitness 0.68)—is selected as the first parent, and the process continues with the identification of a second parent.

In Step 4 of the optimization process, finally, after two parents have been selected, crossover is performed to create new chromosomes for the next generation. The number of hidden layers and neurons are exchanged between parent

1 and parent 2, while, for the continuous parameters (i.e., learning rate and momentum), a two-point crossover is used. This method is preferable due to its successful application in various optimization problems (Murata et al., 1996). For each set of parents, a crossover point is chosen randomly. Figure 4 demonstrates the crossover process, where offspring 1 inherits the number of neurons (8), number of hidden layers (5), and continuous values for learning rate and momentum from parent 2. Mutation is subsequently performed, and this process (i.e., the sequence of selection, crossover, and mutation) continues, with each DNN configuration being trained, tested, and evaluated using the fitness function, until the termination criterion (i.e., maximum number of generations) is met. The developed model uses: (i) a population size of 20; (ii) a maximum of 50 generations; (iii) a mutation probability of 0.1; (iv) a crossover probability of 1; (v) a tournament size of 3; (vi) a momentum range of [0.01,0.99]; and (vii) a learning rate range of [0.01, 0.30]. The GA parameter values are determined by experimenting with various values and selecting the one yielding the best results.

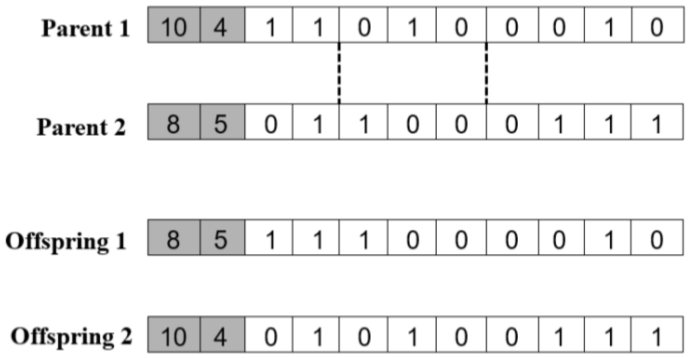


Figure 4: Two-point crossover.

The DNN results are compared with those of ANN and MLR in order to determine the best tool for predicting the process times at each workstation of the production line. The input layer of the neural networks (both ANN and DNN) comprises a number of independent variables, which vary according to the SOPs at the workstations. For example, as indicated in Table 2, the framing workstation has five variables (number of studs, number of doors, number of windows, length, and height), whereas the window/door installation workstation has four variables (number of doors, number of windows, length, and height). The number of hidden layers and the number of neurons in the DNN are determined using the GA optimization technique as described above. Each hidden layer is formed (and the number of neurons in the hidden layer determined) as a weighted sum of all input features based on the connection weights. In this feed-forward step, the output layer receives the values from neurons and calculates their weighted sum in order to produce a prediction result. The output layer, in turn, provides the predicted dependent variable, i.e., the predicted process time (duration). As noted above, ReLu is used as an activation function to introduce non-linearity into the network. It should be noted that the input factors (i.e., independent variables) used to train the MLR, ANN and DNN models vary depending on the workstation (see Table 2).

Table 2: Input variables by workstation.

Input Layer Variables	Framing	Sheathing	Nailing	Butterfly	Window/ Door
Number of Studs	✓	X	X	X	X
Length (ft)	✓	✓	✓	✓	✓
Height (ft)	✓	✓	✓	✓	✓
No. of Doors	✓	✓	✓	✓	✓
No. of Windows	✓	✓	✓	✓	✓
SheetPartial	X	✓	✓	X	X
SheetFull	X	✓	✓	X	X
Area	X	✓	✓	X	X
Nailcount	X	X	✓	X	X
Nailline	X	X	✓	X	X

It should be noted that error rate (e.g., MAE) is used in the present study as the measure of goodness of fit. According to several studies available in the literature, it is a superior alternative to R^2 for evaluating model performance in the case of non-linear data. For instance, Kvålseth (1983) has noted that the use of R^2 in evaluating model performance with non-linear data leads to misinterpretations and produces misleading conclusions. Moreover, in the mathematical literature, it has been concluded that R^2 generally does not increase even when evaluating better-performing non-linear predictive models (Spiess and Neumeyer 2010).

4. CASE STUDY

The proposed method is implemented on a wood frame wall panel production line operated by a modular fabricator in Edmonton, Canada. The workstations of the production line are equipped with computer numerical control (CNC) machines for automated processes, while some workstations require manual work. Figure 5 shows an overview of the wall panel production line, which consists of the following workstations: (i) framing workstation, where wall elements (studs, cripples, and sill plates) are fastened together using CNC machinery to form an interior or exterior wall panel frame; (ii) sheathing workstation, where OSB (oriented strand board) is cut manually and placed on wall panel frames; (iii) nailing workstation where OSB is nailed to the frame using CNC machinery; (iv) panels are then moved to the butterfly table to undergo other operations such as cutting of multiwall panels to single wall panels; and (v) window/door installation station , where windows/doors are installed on the wall panels, which are then transferred to the wall magazine to be stored as they await delivery to the site. It should be noted that interior wall panels are moved from the butterfly table through the window/door bypass to the wall magazine for storage as they await loading, whereas exterior wall panels are transferred from the butterfly table to the window/door installation line. It is worth mentioning that the framing and nailing workstations are semi-automated, while other workstations are operated manually.

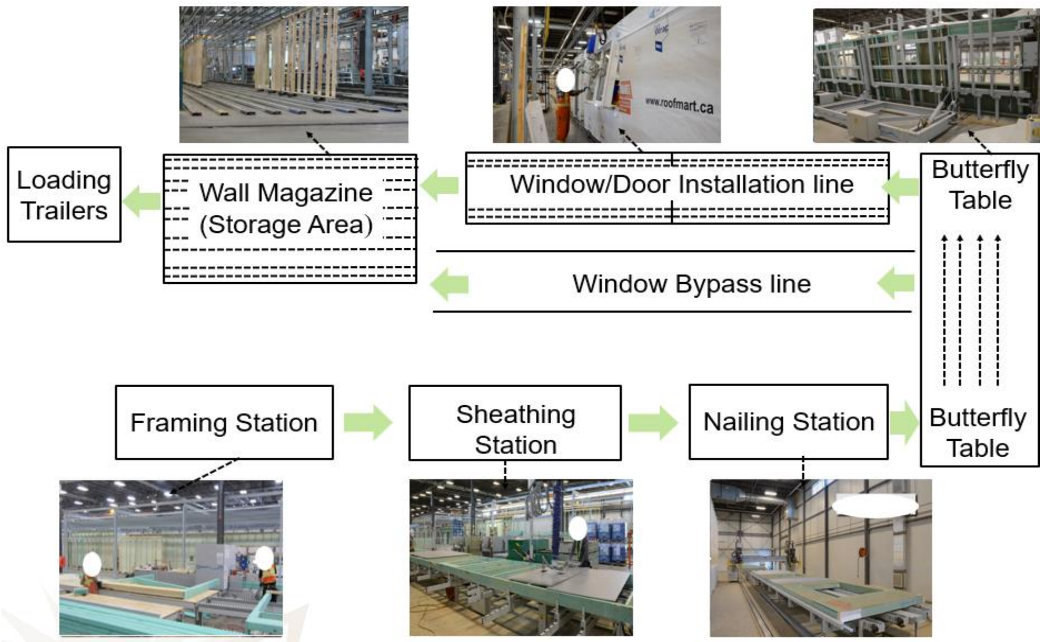


Figure 5: Wall panel production line (Images courtesy of Mohammed Sadiq Altaf).

The process times of wall panels at each workstation are collected using an RFID system (i.e., RFID printer, tag, antenna and reader), as noted above. The data spans the period between July, 2015, and August, 2018, and consists of (i) 416,950 timestamps (i.e., start and finish times) for wall panels, where each timestamp includes 10 attributes (e.g., tagID, panel number, antenna description, etc.); and (ii) the design attributes of each wall panel (e.g., number of studs, number of doors/windows, and width of wall panels), for a total of 93,660 records (see Table 3). A total of 37 design attributes, such as floor, Dstud, Mstud, and Drillhole, are included. The historical RFID data having been obtained from the case company, the next step is to extract the process times of wall panels at each workstation from the “RFID raw data” file based on the difference between start times at consecutive antenna locations (i.e., workstations), as expressed in Equation 1.

Table 3: Summary of wall panel raw data from production line.

Panel	Type	Job	Length (mm)	Width (mm)	Stud	LStud	Windows	Door	Duration
E-16_30DES-17-10-11_10	EXT	30DES-17-10-11	11,919	2,467	22	5	0	0	13
E-17_30DES-17-10-11_10	EXT	30DES-17-10-11	12,175	2,467	23	5	0	0	10
E-18_30DES-17-10-11_10	EXT	30DES-17-10-11	12,168	2,467	14	5	3	0	9
I-2_10GLR-17-0016_00	INT	10GLR-17-0016	12,035	2,467	24	1	0	2	6
I-3_10GLR-17-0016_00	INT	10GLR-17-0016	12,195	2,467	25	1	0	3	6
I-4_10GLR-17-0016_00	INT	10GLR-17-0016	7,329	2,467	14	1	0	2	5
E-6_10GLR-17-0016_00	EXT	10GLR-17-0016	11,576	2,467	6	7	3	0	8
E28_10GLR-17-0016_00	EXT	10GLR-17-0016	4,407	2,467	1	2	1	0	4

Table 4 presents the statistical details (i.e., mean and standard deviation) for each workstation on the production line. For example, the mean process time at the framing workstation is 8.15 min, with a standard deviation of 2.98 min. In addition, initial data visualization is performed in order to gain understanding of the operating procedures at each workstation and the effect of these operations on process times.

Table 4: Mean and standard deviation for each workstation.

Workstation	Mean process time (min)	Standard deviation (min)
Framing	8.15	2.98
Sheathing	9.39	2.79
Nailing	8.35	3.02
Butterfly	6.18	2.48
Window/Door	18.50	5.54

Figure 6(a) shows that there is a high level of variance in process times. For example, at the sheathing workstation, the process time of wall panel 3 is 9 min, that of wall panel 4 is 12 min, and that of wall panel 6 is 2 min. Moreover, the process times variations at the butterfly workstation are very high compared to those at other workstations during the month of June 2016. This is due to the influence of client-defined design customizations, which result in variations in the number of windows, panel length, and number of studs. This variation in process times affects the daily productivity of the workstations, as shown in Figure 6(b). These productivity results show, for example, that the daily production output on March 30 is 12 panels at the sheathing workstation, 22 panels at the framing workstation, and 6 panels at the nailing workstation, indicating an imbalanced production line. It should be noted that the first three workstations (i.e., framing, sheathing, and nailing) operate on multi-wall panels, while from the butterfly station onwards, these multi-wall panels are cut into individual wall panels, which leads to an increase in number of wall panels detected at this workstation by RFID system.

As noted above, prior to development of the predictive model, data pre-processing is implemented. Pre-processing involves: (i) retaining only the initial reading for a given wall panel at a given workstation and omitting subsequent ones if there are multiple readings (for example, as shown in Figure 7a, the initial reading for wall panel E29 at the wall magazine (i.e., record A12, time 2:34 pm) is retained while subsequent readings of this panel at this workstation (i.e., 2:35, 2:36, 2:37, and 2:39 pm) are removed); and (ii) removing TagID, antenna description, location source antenna, first and last read date, backing, floor, siding, weight, model and parent unit (due to the irrelevance of this information to wall panel process times prediction). Additionally, data visualization (i.e., pie chart) is used to remove outlier process times. For example, Figure 7(b) presents the distribution of process times at the butterfly workstation), window/door installation workstation, and transfer table. Here, the process times above 60 min (butterfly workstation), 80 min (window/door installation workstation), and 10 min (transfer table), are removed. The reason for removing these points is that they have excessive process times (i.e., 61 to 410,000 min) that likely represent: (i) work disruptions due to errors in shop drawings and/or (ii) a wall panel being operated on at the given workstation over two successive work days.

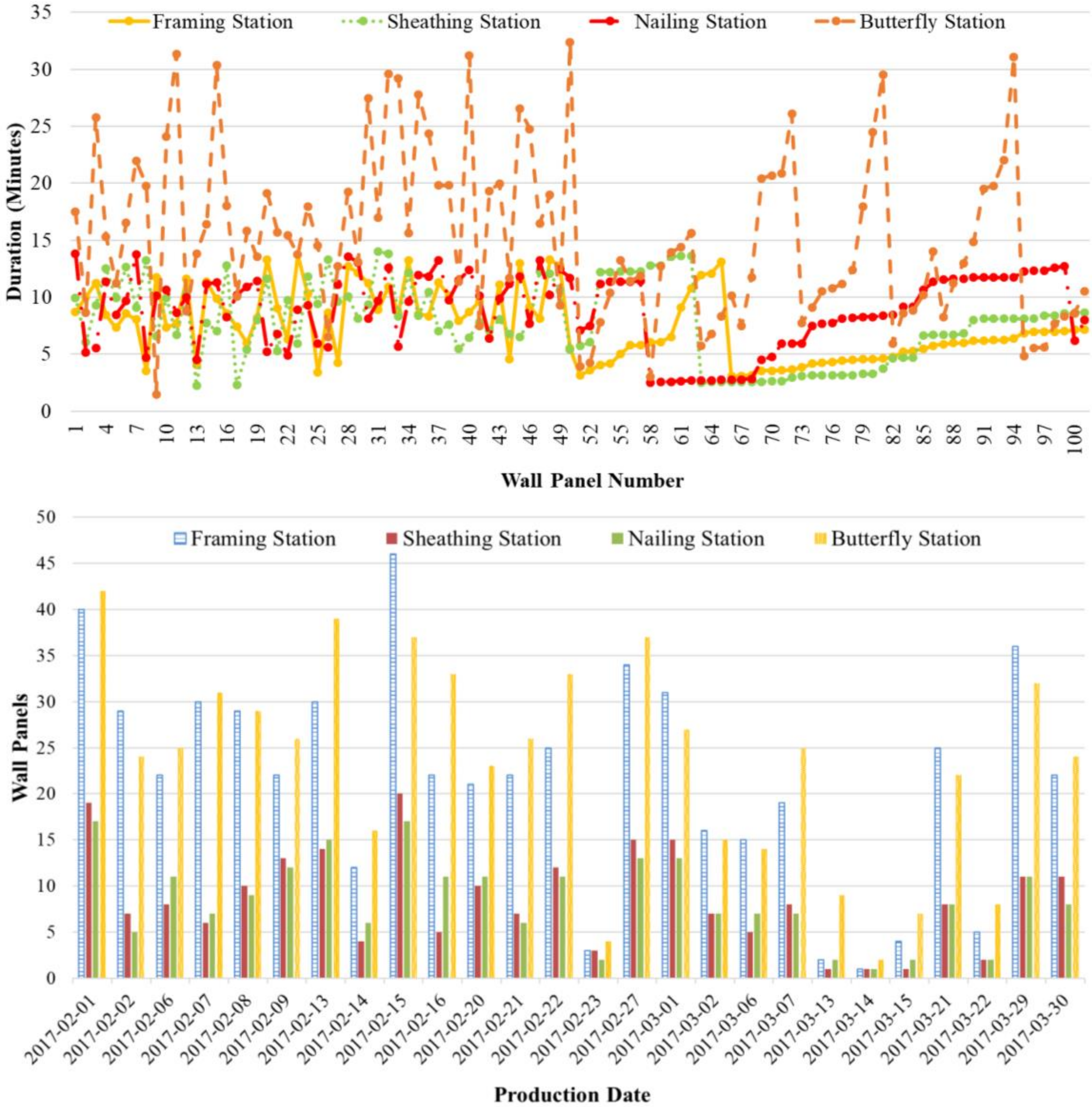
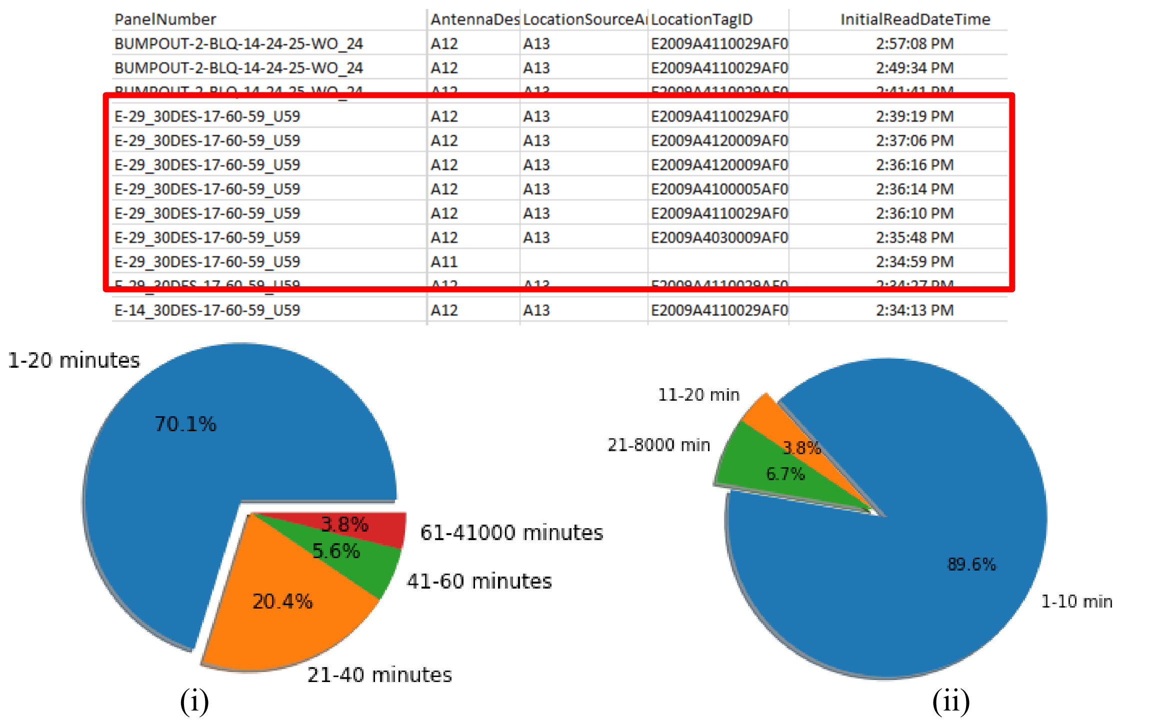


Figure 6: (a) Process times of wall panels; and (b) daily production of wall panels.

Moreover, similar physical attributes of a wall panel (e.g., studs, window, and sheetfull) are combined into a single attribute in order to reduce data dimensions and computation time. For example: (i) DStud, LStud, and MStud are combined into a single attribute ("stud"); and (iii) door and large door are combined as "door". Finally, the min–max normalization technique is applied in order to transform the data into values ranging between 0 and 1 prior to prediction model development. Following data pre-processing, the datasets number 7,256, 2,885, 3,035, 19,998, and 1,868 for the framing workstation, sheathing workstation, nailing workstation, butterfly table, and window/door installation workstation, respectively. It should be noted that the steps in data pre-processing are specific to the case study (i.e., they are a function of the given wall panel design attributes). For other cases, the data pre-processing phase would need to be modified according to the given module component design attributes.



PanelNumber	AntennaDes	LocationSourceA	LocationTagID	InitialReadDateTime
BUMPOUT-2-BLQ-14-24-25-WO_24	A12	A13	E2009A4110029AF0	2:57:08 PM
BUMPOUT-2-BLQ-14-24-25-WO_24	A12	A13	E2009A4110029AF0	2:49:34 PM
E-29_30DES-17-60-59_U59	A12	A13	E2009A4110029AF0	2:39:19 PM
E-29_30DES-17-60-59_U59	A12	A13	E2009A4120009AF0	2:37:06 PM
E-29_30DES-17-60-59_U59	A12	A13	E2009A4120009AF0	2:36:16 PM
E-29_30DES-17-60-59_U59	A12	A13	E2009A4100005AF0	2:36:14 PM
E-29_30DES-17-60-59_U59	A12	A13	E2009A4110029AF0	2:36:10 PM
E-29_30DES-17-60-59_U59	A12	A13	E2009A4030009AF0	2:35:48 PM
E-29_30DES-17-60-59_U59	A11			2:34:59 PM
E-14_30DES-17-60-59_U59	A12	A13	E2009A4110029AF0	2:34:13 PM

Figure 7: (a) RFID data of wall panels at workstations; and (b) distribution of wall panel process times at: (i) butterfly workstation; and (ii) transfer table.

As noted above, variation in process times at workstations due to differences in design attributes of customized wall panels leads to an imbalanced production line, hence the development of a predictive model for estimating process times (for use in scheduling and planning) that considers wall panel design factors. Based on the training dataset (i.e., 80% of the data) and the independent variables (e.g., panel length, number of studs, doors, windows, etc.) used as model inputs, the DNN is developed and validated using a 10-fold cross-validation technique. The developed DNN consists of an input layer, multiple hidden layers, and an output layer (i.e., process times of wall panels), and each node element is connected and layered with the neurons of the next layer. In addition, ANN and MLR models are developed for the purpose of comparing the results with those generated by the DNN model.

Table 5 presents a comparison of the MAE values obtained for the different numbers of neurons in a single hidden layer of the ANN for each workstation. The number of neurons tested ranges from 1 to 6, and the best result for each workstation (i.e., the minimum MAE achieved) is selected for comparison with the results from DNN. For instance, at the framing workstation, the test with 5 neurons achieves the lowest MAE (2.21 min).

GA optimization is used to minimize the MAE by identifying the near -optimal number of hidden layers, number of neurons, momentum, and learning rate for the DNN model. The computational time required to train the prediction models for the workstations ranges between 2-3 hours. The ReLu activation function is employed, with the search ranges [3,10] for number of hidden layers, [6,100] for number of neurons, [0.01,0.99] for momentum, and [0.01,0.30] for learning rate. It is worth mentioning that the selection of these ranges of the neural network was based on a combination of prior literature, and practical considerations (i.e., balance computational feasibility with sufficient search space exploration). Additionally, these ranges cover a broad spectrum of hyperparameter values for DNN in order to achieve near-optimal solution in a reasonable run time (i.e., explore broad search space, while achieving computational efficiency). The optimization parameters, meanwhile, are assigned as follows: (i) a population size of 20; (ii) a maximum number of generations of 50; (iii) a mutation probability of 0.1; (iv) a crossover probability of 1; and (v) a number of tournaments of 3. As shown in Table 6, most of the workstations (i.e., framing, sheathing, butterfly and nailing) are found to have a MAE of less than 2.50 min. If we compare this to the work of Mohsen et al. (2022), in their study the MAE of the prediction algorithms ranges from 4.4 min to 9.2 min—meaning that the model developed in the present study achieves better performance. Additionally, the percentage errors achieved by the prediction model for the various workstations are consistent with those of previous prediction models reported in the literature (Mohsen et al. 2022; Alsakka et al., 2024). The DNN results are also compared with ANN and MLR. As these results show, the DNN model is found to achieve superior MAE values with the exception of the nailing workstation (with ANN achieving a better result for this workstation). These various prediction models were developed utilzing Python in a Jupyter notebook environment, which allows to efficiently code, visualize results, and iterate through different modeling approaches. Additionally, a paired t-test was conducted to compare the MAE values of the DNN and regression models for five workstations. The mean difference in MAE for these workstations was -0.12 with a standard deviation of 0.07. The computed t-statistic of -4 exceeded the critical t-value (± 2.776) at a 95% confidence level, n value of 5 and degree of freedom 4. This indicates that the DNN model's predictive performance was statistically better than the regression model and is unlikely to be due to random chance.

Table 5: Mean absolute error measurement for ANN.

Neurons	Framing (MAE)	Sheathing (MAE)	Nailing (MAE)	Butterfly (MAE)	Window/Door (MAE)
1	2.48	5.72	4.93	4.12	8.03
2	2.40	2.30	2.51	3.07	7.76
3	2.42	2.32	2.48	3.10	6.01
4	2.37	2.29	4.46	2.58	5.90
5	2.21	2.36	2.40	2.18	5.12
6	2.39	2.38	2.56	2.93	5.88

Although, the prediction model yields relatively high SMAPE values (ranging between 22% and 28%); however, it is due to the absence of key factors that can affect the process time of wall panels at the workstation. For example, according to the SOP file provided by the case company: (i) workers need to set up the CNC machine after completing each wall panel, but the current data lacks any variables related to this setup time; and (ii) at the sheathing workstation, workers spend time installing guard wrap on the top plate of the wall panel, cutting OSB sheets, labelling the wall number on the top plate, and conducting a quality check (e.g., ensuring that end studs are aligned with the plate); these activities are not included as variables in the current dataset. Additionally, the dataset does not include variables for worker idle time at workstations due to material shortages or machine breakdowns. This is due to the fact that the RFID system records only the timestamps (start and end times) of wall panels at each workstation, and does not capture activities occurring between these timestamps or account for factors that might result in delays (e.g., movement of workers to pick wall panel elements and tools). The developed prediction model can be improved by collecting data related to activities such as OSB cutting, inspection, and labelling of wall panels at the sheathing workstation.

Table 6: Comparison of DNN, ANN and MLR.

Workstations	DNN Selected Value		DNN MAE (min)	DNN SMAPE (%)	ANN Value		ANN MAE (min)	ANN SMAPE (%)	Reg. MAE (min)	Reg SMAPE (%)
	Hidden Layers	Neurons			Hidden layers	Neurons				
Framing	3	74	2.17	25.78%	1	5	2.21	27.94%	2.24	28.86%
Sheathing	3	72	2.11	22.49%	1	4	2.29	24.87%	2.28	24.81%
Nailing	3	14	2.41	28.42%	1	5	2.40	28.31%	2.43	28.68%
Butterfly	7	70	2.05	28.87%	1	5	2.18	31.59%	2.24	32.12%
Window/Door	8	42	4.48	27.62%	1	5	5.12	31.56%	4.61	28.42%

The case study demonstrates that hyperparameter tuning utilizing GA to optimize number of hidden layers/nodes, learning rate and momentum is beneficial to improving the accuracy of the prediction model. The method described in this study is capable of efficiently estimating the fabrication process times of wall panels at workstations of the production line, thereby removing the guesswork from production planning and scheduling. However, it should be noted that the factors (independent variables) used as inputs in the prediction model of this study are a function of the given product design specifications of the case study production line. For other cases, practitioners would need to modify the input phase based on the given design specifications. Additionally, in modular construction, due to unpredictable on-site demand, production managers must frequently alter their plans to accommodate change orders. In this respect, average process times and linear fixed rate (sq. ft. per minute) are used to estimate the process time of module components at workstations for multiple projects. As an alternative to this challenging and error-prone approach, the framework implemented in the case study can be deployed to develop predictive models for estimating process times and streamline the MCM planning and scheduling process as illustrated in the works of Bhatia (2025).

5. CONCLUSION

In modular construction, the module components (e.g., wall, roof, and floor panels) are of varying sizes and design specifications, and these client-defined design variations necessitate dynamic changes to the production line. This poses a challenge for production line managers in their efforts to accurately predict process times, leading to inefficient production line schedules and reduced productivity. In this respect, this research proposes a novel method that uses historical process time data from the wall panel production line as an input in DNN, ANN, and MLR models to predict the process times at each workstation. GA optimization is then implemented for hyperparameter tuning in DNN to find the near- optimal number of hidden layers, number of neurons in each layer, momentum, and learning rate. The implementation of the proposed method in a wall panel production line shows that the developed method is capable of predicting the process times with a MAE of less than 2.50 min for most of the workstations. DNN provides the best results for most of the workstations, although ANN is found to perform better in the case of the nailing workstation.

The contributions of this study include: (i) implementing GA to determine the near -optimal DNN hyperparameters, considering learning rate and momentum along with number of hidden layers and neurons to reduce the prediction error; (ii) developing prediction models for all production line workstations rather than focusing on only one or a few workstations within the production line; (iii) providing production insights (e.g., wall elements that effect the process times) to management, enabling them to improve the productivity of module component prefabrication. Overall, this approach overcomes the reliance on experience-based methods for estimating process times and creating production schedules; and (iv) accurate estimation of process times will enable modular construction companies to quantitively analyse their operations and, hence, to identify sources of waste in the production line (e.g., waiting time of workers at workstations and idle times of wall panels between workstations).

It should be noted that this study is subject to some limitations. First, other metaheuristic algorithms, such as particle swarm optimization (PSO) are not investigated. Therefore, to introduce more diversity, future studies could assess these algorithms for hyperparameter tuning and compare their outcomes with those of GA. Second, the number of workers at each workstation (for manual workstations) and machine setup time (for automated workstations) are not included in the RFID data and are not used as independent variables in the process time prediction. Therefore, future research could incorporate these factors in order to enhance the accuracy of the process times prediction. Third, this research primarily emphasizes the development of predictive method with module design specifications, without focusing on factors such as number of workers, work shifts, material availability and on-site change orders that can affect process time. In this respect, the future work would consider these factors as an input in order to increase the accuracy of the prediction model.

ACKNOWLEDGMENTS

The authors wish to acknowledge the support of Dr. Mohammed Sadiq Altaf in providing the production line data. Special thanks are also extended to Dr. Hassan Bardareh for aiding in automating the extraction of RFID-based production line data.

DATA AVAILABILITY STATEMENT

The data that support the findings of this study are available from the authors upon reasonable request.

REFERENCES

Alsakka F., Yu H., Hamzeh F., and Al-Hussein M. (2023). Data- and knowledge-driven cycle time estimation in offsite construction. Preprints, https://doi.org/10. 20944/preprints202310.1151.v1 (Under Review in Journal of Production Planning and Control)

Alsakka F., Yu H., El-Chami I., Hamzeh F., and Al-Hussein M. (2024). Digital twin for production estimation, scheduling and real-time monitoring in offsite construction. Computers & Industrial Engineering, Vol. 191, 110173. doi: 10.1016/j.cie.2024.110173.

Aghajamali K., Metvaei S., Suliman A., Lei Z., and Chen Q. (2024). Development of a prefabricated construction productivity estimation model through BIM and data augmentation processes. Construction Management and Economics, 1-20.

Altaf M.S., Bouferguene A., Liu H., Al-Hussein M., and Yu H. (2018). Integrated production planning and control system for a panelized home prefabrication facility using simulation and RFID. Automation in Construction, Vol. 85, 369–383. doi: 10.1016/j.autcon.2017.09.009

Ambrogio G., and Gagliardi F. (2013). Design of an optimized procedure to predict opposite performances in porthole die extrusion. Neural Computing and Applications, Vol. 23, 195–206. doi: 10.1007/s00521-012-0916-3

Aggarwal C.C. (2018). Neural networks and deep learning. Cham:Springer, 10, 978, 3.

Assad A. and Bouferguene A. (2022). Data Mining Algorithms for Water Main Condition Prediction Comparative Analysis. Journal of Water Resources, Vol. 148, 2, 04021101. doi: 10.1061/(ASCE)WR.1943-5452.0001512

Bhatia A., Han S.H. and Moselhi O. (2022). A simulation-based statistical method for planning modular construction manufacturing. Journal of Information Technology in Construction, Vol. 27,130–144. doi: 10.36680/j.itcon.2022.007

Bhatia A., Moselhi O. and Han S.H., (2023). Optimized production scheduling for modular construction manufacturing. In ISARC. Proceedings of the International Symposium on Automation and Robotics in Construction, Vol. 40, 270-277. doi: https://doi.org/10.22260/ISARC2023/0038

Bhatia A.P.S. (2025). Automated Palnning and Scheduling Method for Modular Construction Manufacturing. PhD thesis, Concordia University, Montreal QC, Canada.

Bhatia A., Moselhi O. and Han S. (2025). Hybrid Optimization Approach for Scheduling in Panelized Construction Manufacturing. Journal of Construction Engineering and Management, Vol. 151(8), 04025107. doi: 10.1061/JCEMD4.COENG-16565.

Chu W., Han S.H., Zhen L., Hermann U. and Hu D.(2020). A predictive model for scaffolding man-hours in heavy industrial construction projects. Proceedings of the International Symposium on Automation and Robotics in Construction. Vol. 37, 976–983. doi:10.22260/ISARC2020/0135

Callens A., Morichon D., Abadie S., Delpey M. And Liquet B. (2020). Using Random forest and Gradient boosting trees to improve wave forecast at a specific location. Applied Ocean Research. Vol. 104, 102339. doi: 10.1016/j.apor.2020.102339

Dragicevic N. and Riaz K. (2024). Seizing the Modular Construction Opportunity, CSA public policy centre (accessed Feb, 2024).

El-Sawah H. and Moselhi O. (2014). Comparative study in the use of neural networks for order of magnitude cost estimating in construction. Journal of Information Technology in Construction. Vol. 19(27), 462–473.

Feld W.M. (2000). Lean manufacturing: tools, techniques, and how to use them. CRC press.

Hasan M. and Lu M. (2022). Variance Analysis on Regression Models for Estimating Labor Costs of Prefabricated Components. Journal of Computing in Civil Engineering, Vol. 36(5), 04022019. doi: 10.1061/(ASCE)CP.1943-5487.0001037

Huang M.S. and Chang C.H. (2024). Hyperparameter optimization strategy of multilayer perceptron model for injection molding quality prediction. The International Journal of Advanced Manufacturing Technology, Vol. 131(3), 1879-1895.

Kvålseth T.O. (1983). Note on the R2 measure of goodness of fit for non-linear models. Bulletin of the Psychonomic Society, Vol. 21, 79–80. doi: 10.3758/BF03329960

Koc K., Ekmekcioğlu O. and Gurgun A.P. (2021). Integrating feature engineering, genetic algorithm and tree-based machine learning methods to predict the post-accident disability status of construction workers. Automation in Construction.Vol. 131, 103896. doi: 10.1016/j.autcon.2021.103896

Mohsen O., Mohamed Y. and Al-Hussein M. (2022). A machine learning approach to predict production time using real-time RFID data in industrialized building construction. Advanced Engineering Informatics, Vol. 52, 101631. doi: 10.1016/j.aei.2022.101631

Mohsen O., Petre C. and Mohamed, Y. (2023). Machine-learning approach to predict total fabrication duration of industrial pipe spools. Journal of Construction Engineering and Management, Vol. 149(2), 04022172. doi: 10.1061/JCEMD4.COENG-11973

Mohsenijam A., Siu M.F.F. and Lu M. (2017). Modified stepwise regression approach to streamlining predictive analytics for construction engineering applications. Journal of Computing in Civil Engineering, Vol. 31(3), 04016066. doi: 10.1061/(ASCE)CP.1943-5487.0000636

Mattsson S. (2017). Measuring operator emotion objectively at a complex final assembly station. Advances in neuroergonomics and cognitive engineering, Florida, USA, 223–232.

Moon S., Hou L. and Han S.H. (2023). Empirical study of an artificial neural network for a manufacturing production operation. Operations Management Research, Vol. 16, 311–323. doi: 10.1007/s12063-022-00309-0

Moselhi O., Hegazy T. and Fazio P. (1991). Neural networks as tools in construction. Journal of construction engineering and management, Vol. 117, 4, 606–625. doi: 10.1061/(ASCE)0733-9364(1991)117:4(606)

Mohamed B. and Moselhi O. (2022). Conceptual estimation of construction duration and cost of public highway projects. Journal of Information Technology in Construction, Vol. 27(29), 595–618. doi: 10.36680/j.itcon.2022.029

Makridakis S. (1993). Accuracy measures: theoretical and practical concerns. International Journal of Forecasting, Vol. 9, 527–529. doi: 10.1016/0169-2070(93)90079-3

Murata T., Ishibuchi H. and Tanaka H. (1996) Multi-objective genetic algorithm and its applications to flowshop scheduling. Computers & Industrial Engineering, Vol. 30(4), 957–968. doi: 10.1016/0360-8352(96)00045-9

Nik-Bakht M., An C., Ouf M., Hafeez G., Dziedzic R., Han S.H., Nasiri, F., Eicker U., Hammad A., Moselhi O. (2021). Value Stream Mapping of Project Lifecycle Data for Circular Construction. International Symposium on Automation and Robotics in Construction, Vol. 38, 1033–1042. doi: 10.22260/ISARC2021/0139

Omar A. and Moselhi O. (2022). Hybrid feature selection framework for predicting bridge deck conditions. Journal of Information Technology in Construction, 27. doi: 10.36680/j.itcon.2022.050

Pannakkong W., Thiwa-Anont K., Singthong K., Parthanadee P., and Buddhakulsomsiri J. (2022). Hyperparameter tuning of machine learning algorithms using response surface methodology: A case study of ANN, SVM, and DBN. Mathematical problems in engineering, (1), 8513719.

Razali N.M. and Geraghty J. Genetic algorithm performance with different selection strategies in solving TSP. Proceedings of the World Congress on Engineering, Vol. II, London, UK, Jul 6–8, 2011. 6 pages.

Assaad R.H., El-adaway I.H., Hastak M, LaScola Needy k, (2022). Quantification of the state of practice of offsite construction and related technologies: current trends and future prospects, J. Constr. Eng. Manag. Vol. 148, 1–20.

Rashid K.M., and Louis J. (2021). Automated active and idle time measurement in modular construction factory using inertial measurement unit and deep learning for dynamic simulation input. Proceedings of the 2021 Winter Simulation Conference, Phoenix, AZ, Dec. 13–15, 8 pages. doi: 10.1109/WSC52266.2021.9715446

Smith S. D. (1999). Earthmoving productivity estimation using linear regression techniques. Journal of Construction Engineering and Management, Vol. 125, 133–141. doi: 10.1061/(ASCE)0733-9364(1999)125:3(133)

Schneckenreither, M., Haeussler, S. and Gerhold, C., (2021). Order release planning with predictive lead times: a machine learning approach. International Journal of Production Research, 59(11),.3285-3303.

Spiess A.N, Neumeyer N. (2010). An evaluation of R2 as an inadequate measure for non-linear models in pharmacological and biochemical research. BMC Pharmacology, 10, 6. doi: 10.1186/1471-2210-10-6

Taiwo R, Hussein M, Zayed T. (2022). An Integrated Approach of Simulation and Regression Analysis for Assessing Productivity in Modular Integrated Construction Projects. Buildings, 12(11), 2018. doi: 10.3390/buildings12112018