Automation in Construction 170 (2025) 105904

Contents lists available at ScienceDirect

Automation in Construction

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

Hybrid deep learning model for accurate cost and schedule estimation in construction projects using sequential and non-sequential data

Min-Yuan Cheng *, Quoc-Tuan Vu, Frederik Elly Gosal

Department of Civil and Construction Engineering, National Taiwan University of Science and Technology, 10607, #43, Sec. 4, Keelung Rd., Taipei, Taiwan

ARTICLE INFO

Keywords:
Time-independent
Time-dependent
Hybrid deep learning
Neural network
Bidirectional gated recurrent unit
Optical microscope algorithm

ABSTRACT

Accurate estimation of construction costs and schedules is crucial for optimizing project planning and resource allocation. Most current approaches utilize traditional statistical analysis and machine learning techniques to process the vast amounts of data regularly generated in construction environments. However, these approaches do not adequately capture the intricate patterns in either time-dependent or time-independent data. Thus, a hybrid deep learning model (NN-BiGRU), combining Neural Network (NN) for time-independent and Bidirectional Gated Recurrent Unit (BiGRU) for time-dependent, was developed in this paper to estimate the final cost and schedule to completion of projects. The Optical Microscope Algorithm (OMA) was used to fine-tune the NN-BiGRU model (OMA-NN-BiGRU). The proposed model earned Reference Index (RI) values of 0.977 for construction costs and 0.932 for completion schedules. These findings underscore the potential of the OMA-NN-BiGRU model to provide highly accurate predictions, enabling stakeholders to make informed decisions that promote project efficiency and overall success.

1. Introduction

The construction industry has experienced significant growth in scale and complexity in recent years [1]. In response to this transformation, decision-making in the construction industry has become increasingly challenging as well as crucial to ensuring project success [2]. In project management, completing a project on schedule and within budget is a crucial indicator of project success. Aziz and Abdel-Hakam [3] noted construction delays disrupt project schedules and lead to significant cost overruns. The close interrelationships among on-schedule completion, budget compliance, and project success underscores the importance of making accurate and dependable predictions regarding project completion schedule and cost. Accurately forecasting completion schedule and associated cost is critical to avoiding overspending and minimizing the risk of disputes between contractors and clients across the various stages of a project.

Estimating construction project schedules to an acceptable level of accuracy is a challenging task primarily because of the multifaceted and ever-changing nature of the factors that influence construction timelines [4]. Yang and Wei [5] and Marzouk and El-Rasas [6] confirmed construction projects to be subject to the complex interplay of numerous variables, with each exerting a unique influence on the project timeline. In addition to issues with scheduling, the operational uncertainties inherent to the construction industry often lead to cost overruns, necessitating the continuous assessment of costs to ensure project profitability [7]. However, firms frequently overlook cost changes and information updates during construction, which hinders effective cost control [8]. Consequently, accurate cost estimation at various project stages is crucial, as poor estimates may reduce profits and cause project failure [9]. Cost estimation involves projecting expenses for materials, labor, equipment, methods, location, structural type, schedule, recycled materials [10] as well as economic factors such as liquidity and price indices, which, while factors of influence on cost fluctuations, are often overlooked [11]. Thus, achieving accurate and early cost estimation is essential for optimizing cost savings and promoting sustainability in construction projects. Consequently, accurately predicting construction completion schedule requires both a deep understanding of these variables and a robust modeling and prediction framework.

Data-driven decision-making has emerged as a critical approach for construction stakeholders to navigate dynamically changing environments [12]. By leveraging the wealth of data generated throughout the lifecycle of a project lifecycle or in historical cases, decision-makers can gain valuable insights, identify trends, and make informed choices that drive efficiency, mitigate risks, and optimize resource allocation. This

* Corresponding author.
E-mail addresses: myc@mail.ntust.edu.tw (M.-Y. Cheng), d11205810@mail.ntust.edu.tw (Q.-T. Vu), m11005810@mail.ntust.edu.tw (F.E. Gosal).

https://doi.org/10.1016/j.autcon.2024.105904
Received 16 July 2024; Received in revised form 19 November 2024; Accepted 30 November 2024
Available online 6 December 2024
0926-5805/

Table 1
Published Studies on Predicting Final Cost and Schedule in Construction Projects.

No.	Scholar	Method Type	Applied Methodology	Data Type		Problem-Solving	
				Non-Sequential	Sequential	Cost	Schedule
1	Kim & Reinschmidt [25]	Traditional and Statistical	Earned Value Management and Earned Duration Management	✓	–	✓	–
	Andrade et al. [26]			✓	–	–	✓
2	Thomas & Thomas [27]		Statistical Regression	✓	–	✓	✓
3	Al-Zwainy et al. [28]		Questionnaire Form and Analysis	✓	–	✓	–
4	Plebankiewcicz [29]		Fuzzy Mamdani	✓	–	✓	–
5	Nadafi et al. [30]		Interval Grey Numbers	✓	–	✓	✓
6	Abu El-Maaty et al. [31]		Linear Regression and Fuzzy	✓	–	✓	✓
7	Chakraborty et al. [32]	Basic Machine Learning	Light and Natural Gradient Boosting	✓	–	✓	–
8	Li & Liu [33]		Artificial Neural Network	✓	–	–	✓
	Naik & Radhika [34]			✓	–	✓	✓
	Peško et al. [35]			✓	–	✓	✓
	Balali et al. [36]			✓	–	✓	–
	Jaber et al. [37]			✓	–	✓	✓
	Ujong et al. [38]			✓	–	✓	✓
9	Cheng et al. [7]	Hybrid ML	Fast Messy Genetic Algorithm – Support Vector Machine	✓	–	–	✓
10	Cheng & Hoang [39]		Differential Evolution – Least Squared Support Vector Machine	✓	–	✓	–
11	Rafiei & Adeli [40]	Hybrid DL	Deep Boltzmann Machine Learning – Back-Propagation Neural Network/ SVM	✓	✓	✓	–
12	Cheng et al. [4]		Neural Network – Long Short-Term Memory	✓	✓	–	✓

activity is conventionally carried out using a statistical approach or simple machine learning techniques [13–15].Data in the construction industry are intricate because they encompass two distinct time-related attributes, namely time-independent (described using high-dimensional data) and time-dependent (described using sequential data with temporal dependencies). Traditional methods generally struggle to grasp the intricate relationships within high-dimensional data and the temporal dynamics of this data [16]. Thus, an advanced prediction model is needed to overcome this limitation to effectively address the related challenges encountered in engineering problems, particularly those dealing with construction projects.

Deep learning techniques are powerful tools that have been used to make predictions from high-dimensional data across various domains [17]. Neural Networks (NN) have shown remarkable performance in capturing patterns and extracting complex information and relationships between features. However, NN models cannot handle sequence data, which are commonly encountered in engineering applications involving temporal dependencies.

Recurrent Neural Network (RNN) is a class of NN developed to process sequential and temporal data effectively by capturing the information from the previous hidden state and using it in the current state [18,19]. Gated recurrent unit (GRU) is an advanced version of the standard RNN and a compact version of LSTM that addresses sequential problems by utilizing memory mechanisms to store information across the sequence [20]. GRU performs exceptionally well in terms of capturing temporal dependencies on sequential data by modeling dynamic patterns from the data [21,22]. The integration of bidirectional processing, a technique seamlessly assimilated into the RNN framework [23], enables the RNN model to process the sequence in time-dependent data in both forward and backward directions and then combine their output. Applying the bidirectional technique to GRU (BiGRU) allows GRU to process data from the past to the future and vice versa, which can increase the accuracy of the GRU. However, utilizing the BiGRU model alone may not fully leverage the ability of NN to capture complex feature relationships within the time-independent data.

In order to fill this gap and address the limitations of existing methods, a hybrid deep-learning model, NN-BiGRU, which fuses the advantages of neural network NN for time-independent variables and BiGRU for time-dependent variables, was proposed in this study to tackle the current challenges faced in prediction and forecasting problems in the construction and engineering sectors. The objective of this study was to leverage the complementary nature of these two techniques and improve prediction accuracy, especially in situations involving complex engineering datasets with high dimensions and temporal dynamics. Moreover, the Optical Microscope Algorithm (OMA), a novel optimization algorithm inspired by the optical zoom mechanism in physical microscopes [24], is used to optimize the initial parameters of the NN-BiGRU model and the weight of the combined output between NN and BiGRU to generate a final output. By fine-tuning the NN-BiGRU parameter, OMA can adapt the NN-BiGRU architecture to a specific case study and improve the generalizability of a developed model. Cost and schedule are the two primary objectives of construction management and control necessary to ensure project success. The ability to monitor and estimate the cost and schedule at project completion at any time during project implementation enables project managers to take timely actions to prevent cost overruns and schedule delays. The proposed novel hybrid model, OMA-NN-BiGRU, is intended for use as a data-driven decision-making tool to provide construction managers with accurate estimates of final project cost and schedule that may be used to formulate and execute decisions that effectively minimize the risks of cost overruns and scheduling delays.

2. Literature review

2.1. Related research

A list of significant research studies in which statistical analysis, machine learning, and deep learning methodologies have been applied to predictive modeling in the context of construction projects is given in Table 1. These studies provide valuable insights into how data-driven approaches may be utilized to optimize project planning, management, and control, with particular emphasis on the aspects of project schedule and cost.

Previous work by Kim & Reinschmidt [25], Andrade et al. [26], Thomas & Thomas [27], Al-Zwainy et al. [28], Plebankiewicz [29], and Nadafi et al. [30] to enhance the accuracy of traditional/statistical approaches employed for predicting costs, schedules, or both within the construction domain is summarized in Table 1. However, advanced approaches that surpass the capabilities of traditional statistical methods are needed to effectively address the challenges posed by the

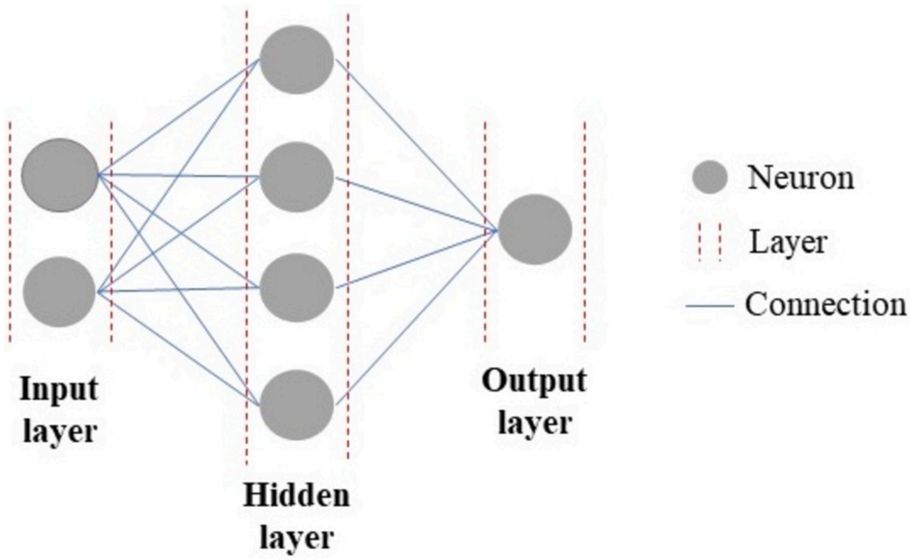

Fig. 1. Artificial Neural Network.

complexity of data (e.g., the concurrent use of time-dependent, time-independent variables) used in the construction domain. Artificial intelligence (AI) techniques that encompass machine learning (ML) and deep learning (DL) approaches are well-suited for capturing and modeling these complex relationships, making these techniques highly valuable for tackling estimation tasks within the construction industry.

Several studies have explored using ML applications in project cost and schedule forecasting. In particular, Abu El-Maaty et al. [31], Chakraborty et al. [32], Li & Liu [33], Naik & Radhika [34], Peško et al. [35], Balali et al. [36], Jaber et al. [37] and Ujong et al. [38] have conducted experiments in this area. However, ML model parameter optimization was not considered in these studies. To tackle this gap, Cheng et al. [7] and Cheng & Hoang [39] employed hybrid ML models that integrate metaheuristic algorithms to optimize model parameters. However, as data features are still assumed to be time-independent in these studies, additional efforts are necessary to preprocess and transform the time-dependent data into time-independent before model deployment.

Hybrid deep learning models have been deployed to utilize and exploit the full potential of time-dependent and time-independent data. Rafiei & Adeli [40] proposed an integrated prediction approach that combined unsupervised Deep Boltzmann Machine Learning (DBM) with a Back-Propagation Neural Network (BPNN) or Support Vector Machine (SVM), with the outputs of these DBM-BPNN and DBM-SVM models employed to estimate construction costs. In addition, Cheng et al. [4] introduced a Neural Network – Long Short-Term Memory (NN-LSTM) model to forecast project schedules and completion timelines. However, these abovementioned approaches do not account for model parameter optimization, thereby presenting an opportunity for further refinement to enhance prediction accuracy. Furthermore, because these studies implemented either cost or schedule only, it is impossible to evaluate the overall effectiveness.

In the light of the drawbacks, including the effort required for dataset handling in the construction domain and ML models, the challenges of optimizing deep learning model parameters, and the lack of comprehensive implementation in construction management, this study was developed to address these gaps using a holistic research approach. The goal is to effectively streamline data processing, optimize hyperparameters, and offer a comprehensive solution tailored to the needs of construction project management.

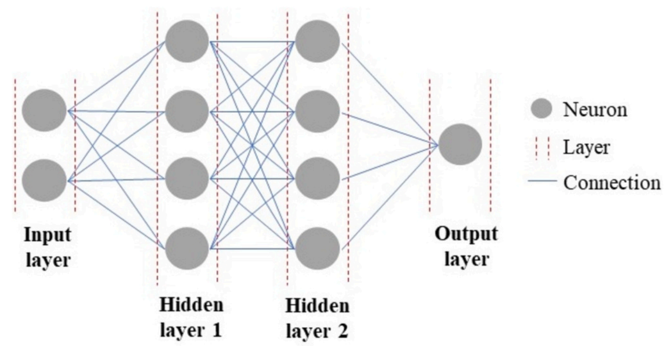

Fig. 2. Deep Neural Network.

2.2. Neural network

NN is a computational approach that simulates the neuronal systems of the human brain. Parallel-distributed processors comprise simple processing units (neurons) that perform computations and store information. NN has evolved significantly since it was first introduced. In 1958, Rosenblatt [41] introduced the perceptron, which is a node associated with a mathematical equation and the basis for the subsequent development of the Artificial Neural Network (ANN). The architecture of ANN with one input layer, one processing or hidden layer, and one output layer is shown in Fig. 1.

The ANN is a variant of NN that consists of one processing layer only. In 1968, Ivakhnenko introduced a new variant of NN, Multi-layer Perceptron (MLP), which consists of multiple perceptron layers [42]. The first trainable MLP model, introduced by Amari [43], consists of 5 hidden layers with 2 modifiable layers for classification tasks. Today, the MLP technique is also known as Deep Neural Network (DNN) to describe its utilization of an NN model with more than one processing layer. The architecture of DNN is shown in Fig. 2.

Computations are performed done in each perceptron or neuron, with the resultant value then passed to the subsequent layer. The mathematical equation used for each neuron node is:

$$Z_i = b + \sum_{i=1}^{i} (W_i, X_i) \quad (1)$$

Where Z_i is the value of the i^{th} neuron in the preceding layer. Weight

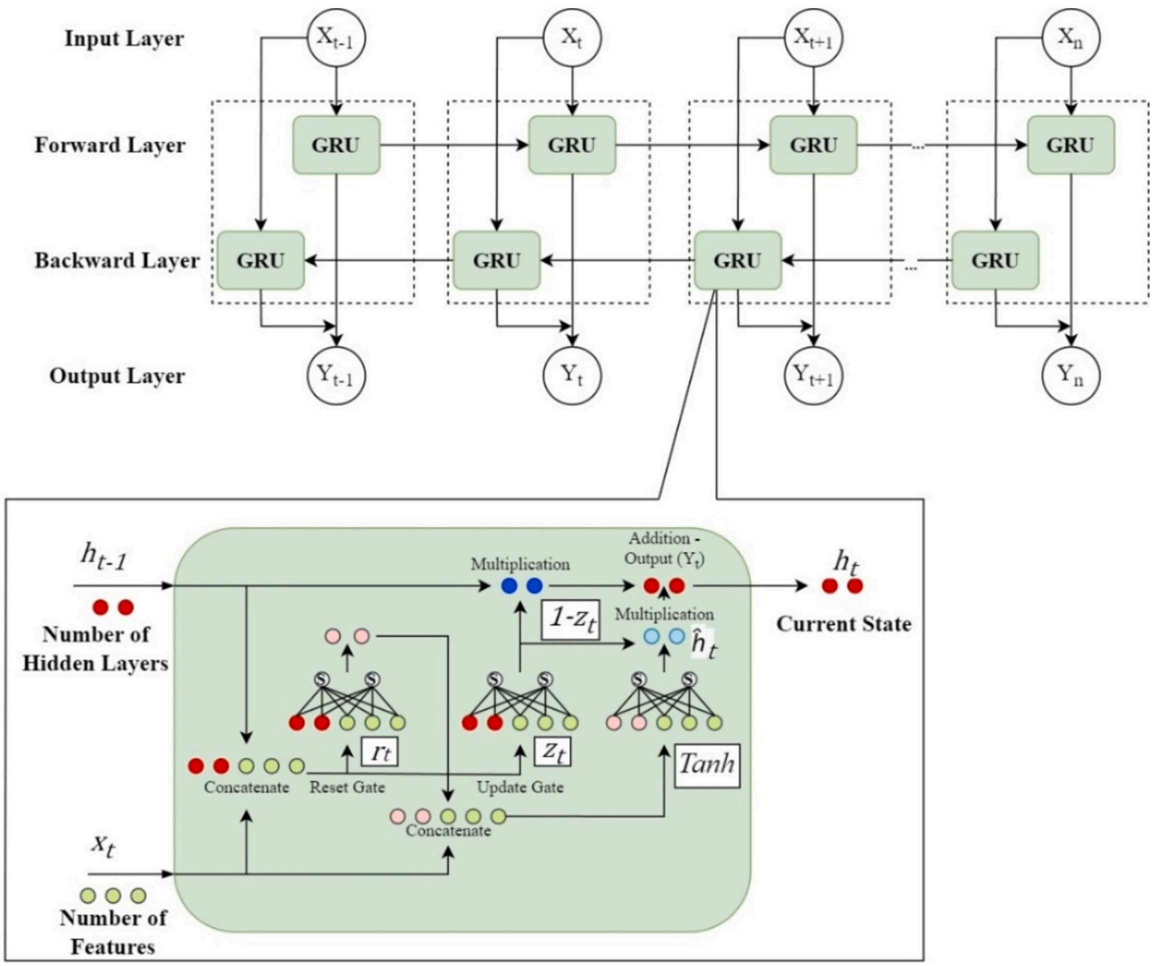

Fig. 3. GRU Memory Cell with Bidirectional Technique.

(W) and bias (b) are the trainable parameters used to train the DNN model.

2.3. Bidirectional gated recurrent unit

GRU utilizes a gate mechanism to process the flow of information through the memory cell. The GRU memory cell designed by Cho et al. [20] consists of two gates, including an update gate and a reset gate. The architecture and process inside a GRU memory cell or unit is shown in Fig. 3.

Update Gate (z_t): The update gate determines how much information from the previous hidden state is passed to the current timestep. This gate uses the previous hidden state and current input values to compute the value of the update gate using the sigmoid activation function.

$$z_t = \sigma(W_z.[h_{t-1}, x_t] + b_z) \tag{2}$$

Reset Gate (r_t): The reset gate determines how much information from the previous hidden state will be ignored when computing the candidate state. This gate uses the previous hidden state and current input values to compute the value of the reset gate using the sigmoid activation function.

$$r_t = \sigma(W_r.[h_{t-1}, x_t] + b_r) \tag{3}$$

Candidate hidden state ($\widehat{h}_t$): The candidate hidden state is a temporary hidden state in which the information is stored and serves as an input in the computation of the current hidden state value. The current input and reset gate values are used to compute the candidate hidden state value using the Tanh activation function.

$$\widehat{h}_t = \varnothing(W_h.[r_t \odot h_{t-1}, x_t] + b_h) \tag{4}$$

Hidden state update: This state generates the final value of the current hidden state by combining the previous hidden state and candidate state values based on the update gate value.

$$h_t = (1 - z_t) \odot h_{t-1} + z_t \odot \widehat{h}_{t-1} \tag{5}$$

In this equation, W_r, W_z, and W_h and b_r, b_z, and b_h are weights and biases corresponding, respectively, to the input, previous hidden state, and transformed previous hidden state.

The Bidirectional Technique, invented in 1997 by Schuster and Paliwal [23], increases the amount of input information available to a network. This technique may be applied to the vanilla version of RNN as well as to LSTM (BiLSTM) and GRU (BiGRU). BiGRU uses the bidirectional technique by utilizing finite sequences to predict or label each element of the sequence based on their past and future contexts. This is accomplished by concatenating the outputs of two GRUs, one of which processes the sequence from left to right, and the other of which processes the sequence from right to left.

The architecture of BiGRU is also shown in Fig. 3, with the reverse GRU learning the sequence from the future to the past, and the forward GRU learning the sequence from the past to the future. The good performance of BiGRU in the engineering field has been demonstrated in the accurate prediction results achieved by processing sequence inputs in time-series problems [44].

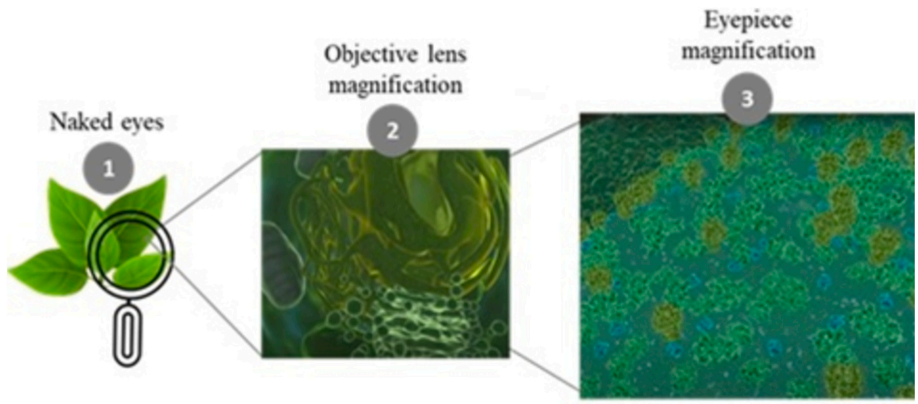

Fig. 4. OMA Search Concept [24].

2.4. Optical microscope algorithm

Optical Microscope Algorithm (OMA) is an optimization algorithm developed based on the magnification / zoom-in process used in physical microscopes [24]. An illustration of OMA search using the example of a leaf as an analogy is shown in Fig. 4.

OMA has 3 phases, including naked eyes, objective lens magnification, and eyepiece magnification.

1. Naked Eyes Phase: In this phase, the populations of the target object or solution are generated using the logistic chaotic map and the following equation:

$$M_i + 1 = M_1 \times 1 - M_i, 0 \leq M_0 \leq 1 \tag{6}$$

The number of solutions corresponds to the determined Number of Population (NP) in the previous phase and the best solution (M_{best}).

2. Objective Lens Phase: In this phase, the generated population is modified via magnification based on the magnifying power value (MP_O) = 1.4. The equation used to modify the population in this phase is:

$$M_{i\ new} = M_1 + rand(0,1) \times MP_o \times M_{best} \tag{7}$$

3. Eyepiece Magnification Phase: In this phase, a local search is conducted by searching the area around the target object or the best solution with the expectation of finding a better solution. First, the search space limit around the best solution is calculated using the following equation:

$$Space = M_j - M_i \text{ if } f(M_i) \geq f(M_j) \tag{8}$$

$$Space = M_i - M_j \text{ if } f(M_i) \leq f(M_j) \tag{9}$$

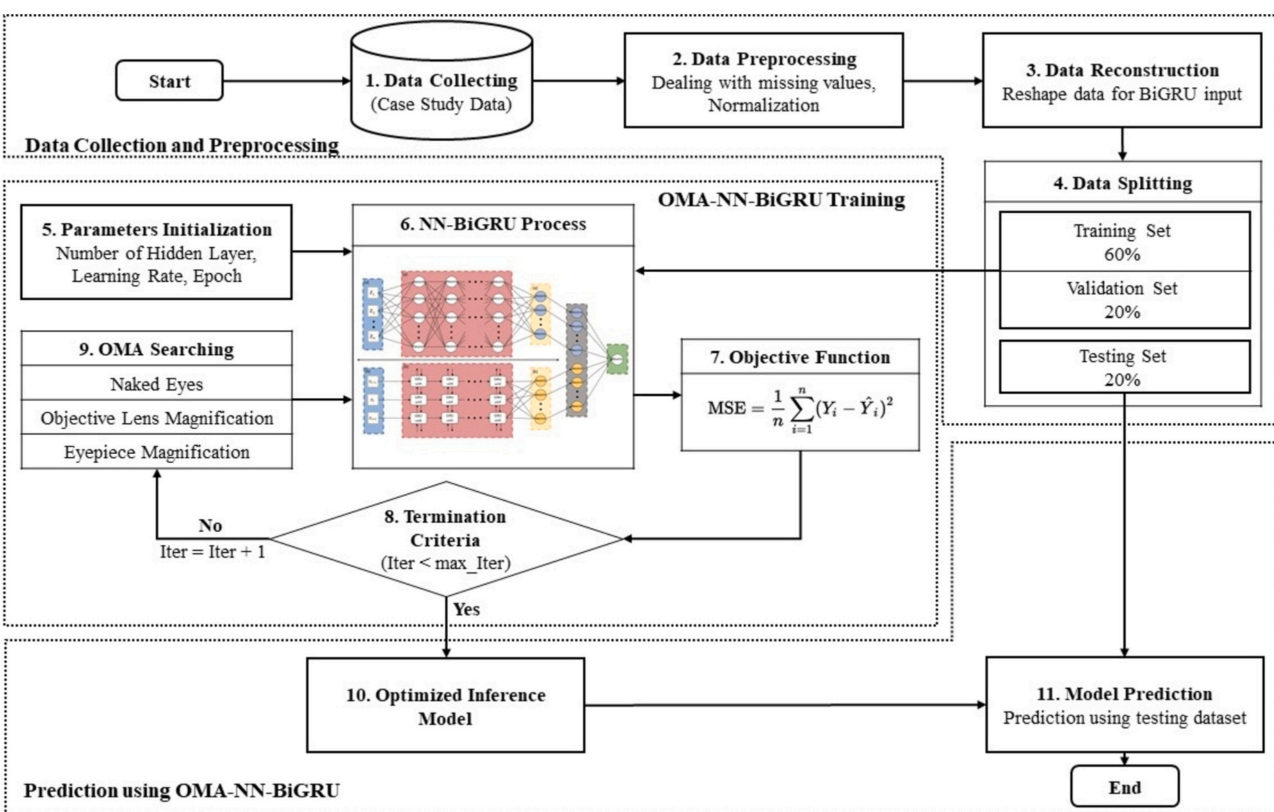

Fig. 5. Model Framework.

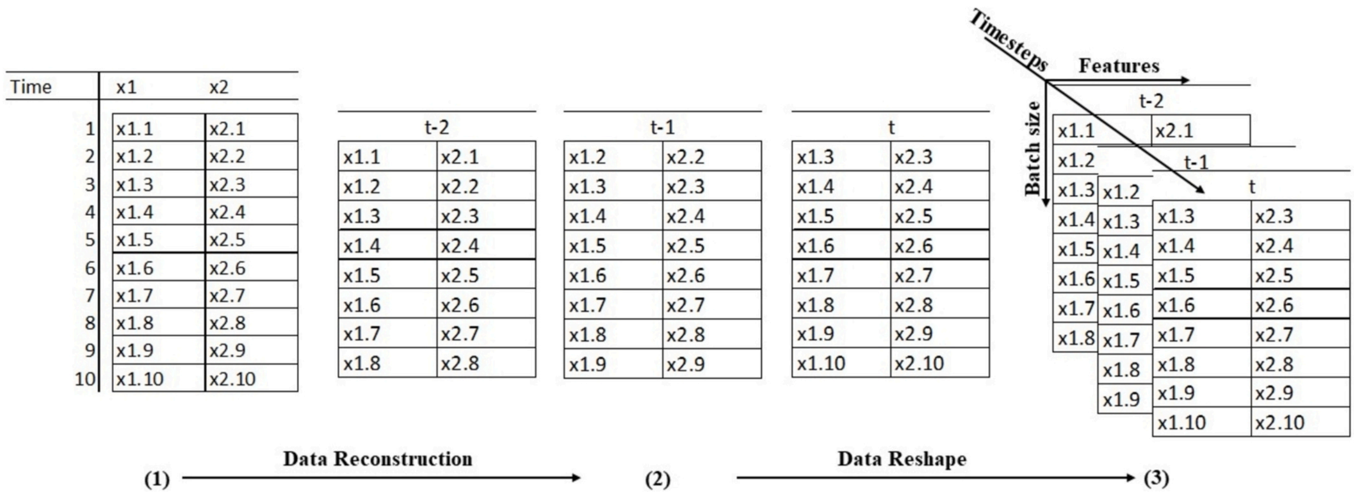

Fig. 6. Reshaping BiGRU Input Data.

Next, the previous best solution is modified by adapting the calculated space and magnifying power (MP_E) = 0.55 using the following equation.

$$M_{i\ new} = M_i + rand(0,1)*MP_E*Space \tag{10}$$

Finally, the new modified population is compared with the current best solution. If the new-modified solution is better than the current best solution, it will replace the current best solution. Otherwise, the current best solution will be retained.

3. Model construction

A detailed explanation of the methodology used to construct the OMA-NN-BiGRU model, including architectural design, internal parameter settings for the OMA optimizer, and the performance evaluation metrics used, is given in this section.

3.1. OMA-NN-BiGRU model framework

The comprehensive framework for the hybrid NN-BiGRU model and its optimization process is illustrated in Fig. 5. The framework covers steps ranging from data collection and preprocessing to OMA-NN-BiGRU training and optimized model prediction.

The framework diagram acts as a guide for navigating through the model implementation process to specific case studies. The following is a detailed explanation of each step in the model framework.

Step 1: Data Collection.

Data may be collected from various sources, including databases, open-source repositories and historical cases. The data collected includes both time-independent, and time-dependent variables. When time-independent and time-dependent variable status is not distinguished in the original source data, the user must interpret this data and assign these variables to the proper category.

The data used in this study fulfills the necessary prerequisites for the proposed model, including both time-dependent and independent variables.

Step 2: Data Preprocessing.

Next, the data is preprocessed to ensure data quality and suitability to the model using the following treatments:

1. Data cleaning: Variables with consistent values are removed to reduce model complexity and enhance computational efficiency.
2. Handling missing values: Linear interpolation is used to fill in missing values, preventing noise and irrelevant patterns.
3. Normalization: All variables, including the output variable, are normalized using min-max scaling to a range of 0 to 1 for accurate outcomes and optimal model performance. The normalization formula used in this study was:

$$x_{norm} = \frac{x - min(x)}{max(x) - min(x)} \tag{11}$$

Where x_{norm} represents the normalized value, x stands for the actual value, *max(x)* signifies the maximum value of the variables, and *min(x)* represents the minimum value of the variables.

4. Categorial variable encoding: Categorical variables are transformed into numerical format using (based on the data characteristics and number of categories) techniques such as one-hot encoding, label encoding, ordinal encoding, binary encoding, and frequency encoding.

Step 3: Data Reconstruction.

In this step, only time-dependent category data are reconstructed for the BiGRU component. BiGRU networks require 3-dimensional inputs to process sequential data with temporal and feature dimensions. Collected data, commonly 2-dimensional, must be reshaped into a 3-dimensional structure. For illustrative purposes, the process of data reconstruction using 3 timesteps [(t-2) ➔ (t-1) ➔ (t)] is shown in Fig. 6 (left to right). In addition, the choice of timesteps is case-dependent, as different datasets and models may require different numbers of timesteps for optimal performance.

After finishing data reconstruction, the time-independent data are assigned to each time step, effectively merging it with the time-dependent data to create a sample.

Step 4: Data Splitting.

In this model, data are not randomly shuffled, as in machine learning applications, to ensure older data is used for training and recent data is used for prediction. By using the Out-of-Sample Validation (OOS) method, the dataset is split into two groups [45]. The first subset, comprising 80 % of the data, is dedicated to training and validating the OMA-NN-BiGRU model, with 60 % of this subset allocated for model training and 20 % allocated for validation, as outlined in **Step 6**. The second subset, comprising the remaining 20 % of the data, is set aside for model prediction and evaluation purposes, as described in **Step 11**.

Step 5: Parameter Initialization.

Prior to OMA-NN-BiGRU model training, NN-BiGRU architecture parameters such as the number of hidden layers, numbers of NN neurons and BiGRU units in the hidden layer, batch size, learning rate, dropout rate, and number of epochs must be manually assigned.

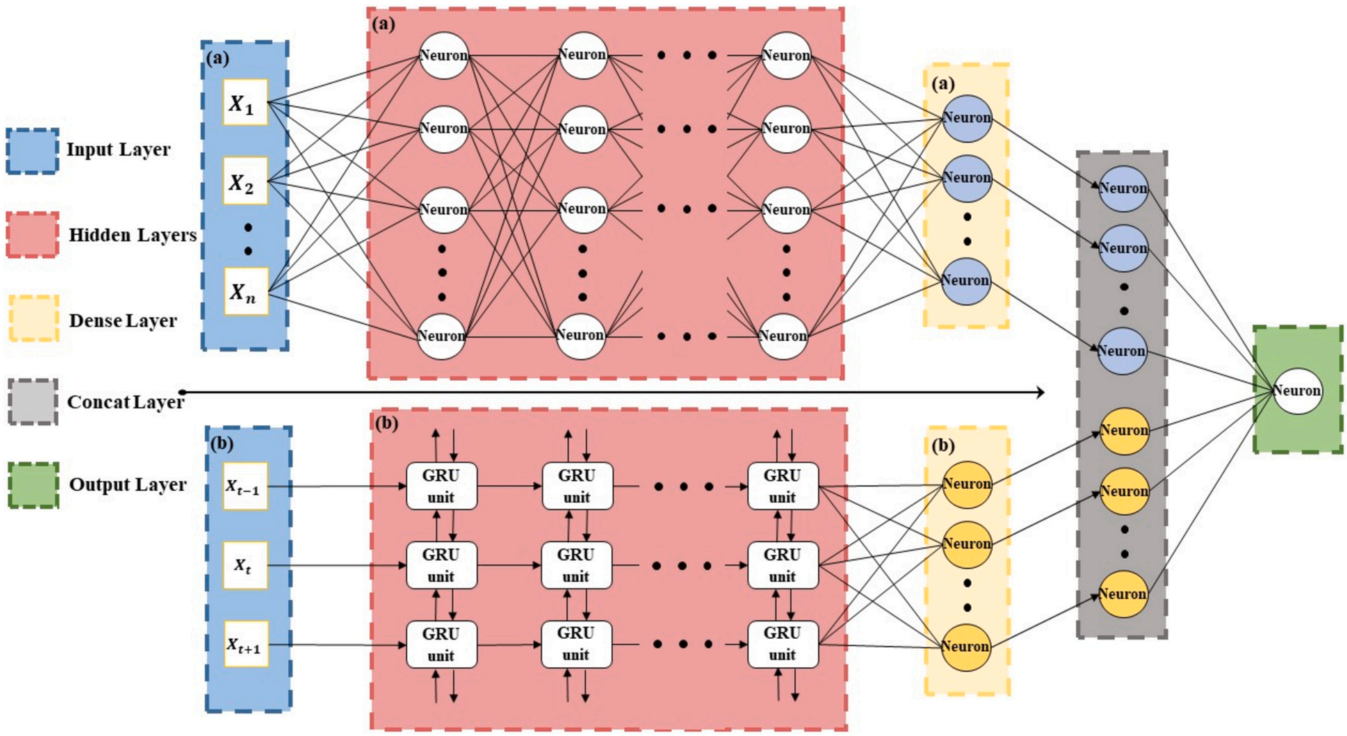

Fig. 7. NN-BiGRU Architecture.

Step 6: NN-BiGRU Process.

NN-BiGRU serves as the learning tool of the model. After the data is preprocessed and parameters are initialized, the data may be fed as inputs into the model to begin the training and validation process. The architecture of NN-BiGRU is shown in Fig. 7.

In this step, the training data (60 % of the total dataset) are split into two streams, which were previously determined in **Step 1**. One stream sends the time-independent data to the NN component, while the other sends the time-dependent data to the BiGRU component. This parallel processing architecture allows the model to effectively handle both time-invariant and temporal aspects of the data simultaneously.

The architecture of NN-BiGRU consists of 5 main layers, namely Input, Hidden, Dense, Concat, and Output. The NN-BiGRU architecture and activation function in each layer are explained below.

1. Input Layer: Raw data enters the model in the input layer, which sets the data shape and format for the subsequent layers. For NN, input has batch size and features. For time-dependent data (e.g., BiGRU), input has batch size, timesteps, and features.
2. Hidden Layer(s): Hidden layers are intermediary layers where inputs are computed and transformed. Each layer has neurons that apply weighted sums and activations to enable complex pattern learning. In both NN and BiGRU, these layers have learned parameters, including neuron/unit count and dropout rate. Dropout prevents overfitting by randomly deactivating neurons during training at a rate determined by the dropout rate [46].
3. Dense Layer: The dense layer links each neuron to all of the neurons in the previous layer. After the hybrid NN and BiGRU layers, it merges with learned representations to make predictions, serving to capture features, transform and align dimensions, and reduce redundancy.
4. Concat Layer: The concatenation layer merges NN and BiGRU dense layers into a single tensor using the concatenation or stacking technique that does not alter the neuron sequence.

Table 2
Activation Functions.

Layer	Activation Function
Input Layer	None
Hidden Layers (NN)	ReLU
Hidden Layers (BiGRU)	Tanh
Concatenate Layer	None
Dense Layer (NN & BiGRU)	Linear
Output Layer	Sigmoid

5. Output Layer: The crucial output layer is the final step to generating predictions. In regression, this layer holds a single neuron that generates continuous predictions.
6. Activation Function: The activation functions used in each model layer are summarized in Table 2:

After completing the training process, NN-BiGRU model performance is evaluated using the Mean Squared Error (MSE) metric. This involves computing the MSE on the training set by comparing the model's predicted outputs against the actual target values. The MSE quantifies the prediction errors during training, allowing for further model optimization.

Step 7: Objective Function.

Upon completing the training phase, the NN-BiGRU model is tasked with predicting the outputs for the 20 % validation set. This set assesses model performance by computing the validation MSE objective function value between predicted and actual values. The validation MSE, as defined in Eq. (12), serves as the performance metric to identify the ideal network configuration through iterative optimization.

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(Y_i - \hat{Y}_i)^2 \tag{12}$$

Where n represents for the number of data points, Y_i is the actual values and $\widehat{Y}_i$ is the predicted values.

Step 8: Termination Criterion.

Table 3
Parameters to be Optimized.

Model	Layer	Description	Parameter
NN	Hidden Layer	Number of Neurons	nn_neurons
		Dropout Rate	nn_dropout
	Dense Layer	Number of Neurons	nn_dense_neurons
BiGRU	Hidden Layer	Number of Units	bigru_units
		Dropout Rate	bigru_dropout
	Dense Layer	Number of Neurons	bigru_dense_neurons
NN-BiGRU		Learning Rate	lr

NN-BiGRU training iterates until reaching the maximum allowed number of iterations (max-iter). When max-iter is not met, a new loop begins and the OMA in Step 8 is applied to find an updated set of parameters. These new parameters then drive the next cycle of the training and validation datasets. This iterative process, controlled by the optimization algorithm, tunes model parameters progressively to improve performance.

Step 9: OMA Searching.

In this step, OMA utilizes its three core processes (Naked Eyes, Objective Lens, and Eyepiece Magnification) to search for the optimal NN-BiGRU parameter values by generating a new parameter set. The resultant MSE value is compared against the MSE value of the previous iteration, and the configuration with the lower error is retained and applied in the next loop (**Steps 6–8**).

OMA optimizes NN-BiGRU parameters, including NN-BiGRU architecture parameters and the output weight.

1. NN-BiGRU Model Architecture Optimization: OMA optimizes NN and BiGRU hyperparameters (shown in Table 3) to minimize the objective function and addressing overfitting and underfitting.

2. NN-BiGRU Output Weight Optimization: Output weights (W_i), which link the concatenate layer to the output layer, are optimized using OMA to enhance predictive accuracy. Each weight, within the range of -1 to 1, affects the final prediction, as illustrated in Fig. 8.

The equations below compute output using sigmoid activation:

$$z = \sum W_i X_i \tag{13}$$

$$Y = f(z) \tag{14}$$

$$Y = \frac{1}{1 + e^{-z}} \tag{15}$$

Where Y represents the prediction result, X_i is the value of the neuron in the concat layer, W_i is the weight associated with links, z is the sum of the multiplication of W_i and X_i, and f is the sigmoid function.

3. OMA Parameter Setting: The OMA searches iteratively for the optimal NN-BiGRU value 50 times, with 10 sets of solutions for each iteration and the boundaries LB (lower-bound) and UB (upper-bound) defining the search range. The parameters and boundaries used in the OMA search are shown in Table 4.

Step 10: Optimized Inference Model.

When the termination criteria are met, the optimal parameter set tuned on the training data is obtained. Next, the final inference model is constructed using these optimized parameters and then evaluated using the testing data to assess its performance and generalization ability.

Step 11: Model Prediction.

The set of optimal parameter values for a specific case study obtained in the previous process is applied to the NN-BiGRU architecture and output weight (OMA-NN-BiGRU). The model is now ready to predict using testing data (20 %) or new specific case study data.

3.2. Performance evaluation metrics

Model performance must be objectively evaluated. In this study, metrics assessing accuracy, quality, and reliability were used, with the results for OMA-NN-BiGRU compared to state-of-the-art methods. OMA-NN-BiGRU focuses on regression tasks. In addition to MSE, the following metrics were used for regression:

1. Root Mean Squared Error (RMSE)

$$RMSE = \sqrt{\left(\frac{1}{n}\sum_{i}^{n}(y_i - f_i)^2\right)} \tag{16}$$

Table 4
OMA Parameters and Search Range.

Parameter	Value (LB – UB)
nn_neurons	5–200
nn_dropout	0–0.5
nn_dense_neurons	1–20
bigru_units	5–100
bigru_dropout	0–0.5
bigru_dense_neurons	1–20
Output Weight (W_i)	(-1) – 1
lr	0.0001–0.1
Iteration	50
Population	10

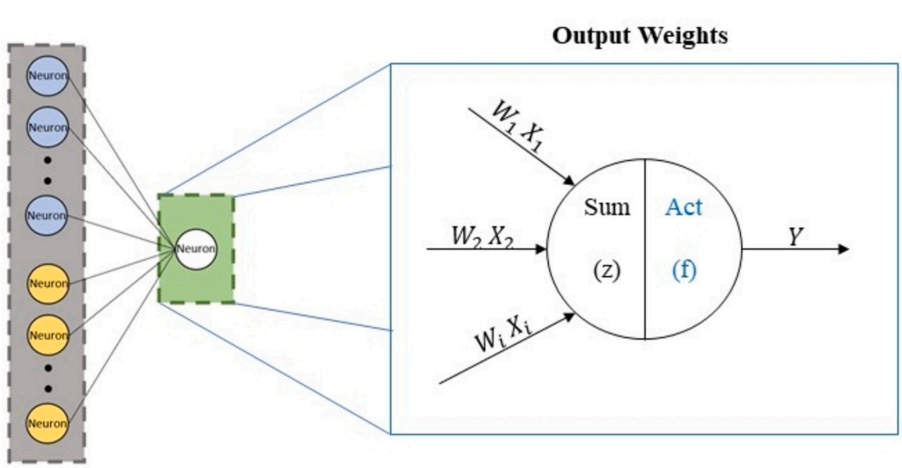

Fig. 8. Output Weight.

2. Mean Absolute Error (MAE)

$$MAE = \frac{1}{n}\sum_{i}^{n}|y_i - f_i| \tag{17}$$

3. Mean Absolute Percentage Error (MAPE)

$$MAPE = \frac{100}{n}\sum_{i}^{n}\left|\frac{y_i - f_i}{y_i}\right| \tag{18}$$

4. Coefficient Correlation (R)

$$R = \frac{n\sum_{i}^{n} y_i f_i - \left(\sum_{i}^{n} y_i\right)\left(\sum_{i}^{n} f_i\right)}{\sqrt{\left[n\left(\sum_{i}^{n} y_i^2\right) - \left(\sum_{i}^{n} y_i\right)^2\right]\left[n\left(\sum_{i}^{n} f_i\right)^2 - \left(\sum_{i}^{n} f_i\right)^2\right]}} \tag{19}$$

5. Coefficient of Determination (R^2)

$$R^2 = 1 - \frac{\sum_{i}^{n}(y_i - f_i)^2}{\sum_{i}^{n}(y_i - \overline{y})^2} \tag{20}$$

Where y_i represents the actual observed values in the dataset; f_i represents the predicted values generated by the model; and n denotes the total number of observations or data points.

The results of the five metrics employed in this study were combined into a composite score for overall model performance called the Reference Index (RI). RI represents model performance across all of these metrics, with individual scores equally weighted and then summed. The equation used to calculate RI is noted below:

$$RI = \frac{R + R^2 + (1 - RMSE_{norm}) + (1 - MAE_{norm}) + (1 - MAPE_{norm})}{5} \tag{21}$$

The models compared in this study were ranked based on RI value (ranging between 0 and 1), with higher values indicating better model performance. The variables, namely $RMSE_{norm}$, MAE_{norm} and $MAPE_{norm}$, should be normalized before calculating RI. These are obtained by scaling each error metric to fall within a comparable range to avoid dominance by anyone metric. This normalization allows for consistent comparison and integration into RI.

4. Model implementation and evaluation

Case 1. Residential Building Construction Cost

In the first case study, the construction cost of a residential building project was estimated based on historical project data taken from Rafiei & Adeli [40]. The dataset was obtained from 372 low- and mid- rise residential buildings with 3–9 aboveground floors built between 1993 and 2008 in Tehran, Iran. The dataset consists of 8 project physical and financial variables (P&F), 19 economic variables and indices (EV&I), and construction cost, which serves as the target value. The objective of this case study was to predict the construction cost of the Iranian residential building project considering the economic variables and indices.

The input variables for this case study were P&F variables and EV&I variables. The physical and financial variables for the project were input for NN due to the time-independent nature of those variables. EV&I variables were used as BiGRU inputs due to the time-dependent nature of the data, as these variables were collected within a predetermined time prior to the start of project construction and recorded each quarter of the year. The time-independent inputs for NN and the time-dependent inputs for BiGRU are shown in Table 5.

Table 5
Case 1 Input Variables.

No.	Description	Unit	Category	
			Time-independent	Time-dependent
1	Project locality defined in terms of zip codes	–	●	
2	The total floor area of the building	m^2	●	
3	Lot area	m^2	●	
4	Total preliminary estimated construction cost based on the prices at the beginning of the project	Dollars	●	
5	Preliminary estimated construction cost based on the prices at the beginning of the project	Dollars/m^2	●	
6	Equivalent preliminary estimated construction cost based on the prices at the beginning of the project in a selected base year	Dollars/m^2	●	
7	Duration of construction	month	●	
8	The price of the unit at the beginning of the project per square meter	Dollars/m^2	●	
9	The number of building permits issued	–		●
10	Building services index (BSI) for a preselected base year	–		●
11	Wholesale price index (WPI) of building materials for the base year	–		●
12	Total floor areas of building permits issued by the city/ municipality	m^2		●
13	Cumulative liquidity	millions of dollars		●
14	Private sector investment in new buildings	millions of dollars		●
15	The land price index for the base year	millions of dollars		●
16	The number of loans extended by banks in a time resolution	–		●
17	The number of loans extended by banks in a time resolution	millions of dollars		●
18	The interest rate for the loan in a time resolution	%		●
19	The average construction cost of buildings by the private sector at the time of completion of construction	millions of dollars/m^2		●
20	The average construction cost of buildings by the private sector at the beginning of the construction	millions of dollars/m^2		●
21	The official exchange rate for dollars	%		●
22	Nonofficial (street market) exchange rate for dollars	%		●
23	Consumer price index (CPI) in the base year	–		●
24	CPI of housing, water, fuel & power in the base year	–		●
25	Stock market index	–		●
26	Population of the city	People		●
27	Gold price per ounce	Dollars		●

The dataset was already reconstructed into 5 timesteps for the EV&I variables. Therefore, the total number of variables was 8 P&F variables added up by 5 timesteps and multiplied by 19 EV&I variables, summed to a total of 103 variables. The 372 historical cases used in Case 1 are

Table 6
Case 1 Historical Cases.

No	Physical and Financial Variables				Economic Variables and Indices							Construction Cost (C)
					timestep 1			…	timestep 5			
V	1	2	…	8	9	…	27	…	9	…	27	28
1	1.00	3150.00	…	1200.00	6713	…	628,133	…	7196	…	601,988	410.00
2	1.00	7600.00	…	2900.00	3152	…	1,188,996	…	3678	…	929,027	1000.00
3	1.00	4800.00	…	630.00	1627	…	524,765	…	2693	…	377,829	170.00
4	1.00	685.00	…	140.00	2580.93	…	141,543	…	1381	…	122,032	30.00
5	1.00	3000.00	…	5000.00	6790	…	2,318,397	…	5606	…	1,734,974	700.00
	.	.	.	.	.	.	.	…	.		.	.
	.	.	.	.	.	.	.	…	.		.	.
368	20.00	1350.00	…	830.00	2700	…	1,067,772	…	2734	…	865,878.9	150.00
369	20.00	600.00	…	570.00	6713	…	648,845.6	…	5728	…	606,524.2	80.00
370	20.00	1900.00	…	640.00	2918	…	1,181,856.2	…	2700	…	1,067,772	220.00
371	20.00	510.00	…	790.00	2247	…	833,494.6	…	6796	…	669,640.3	110.00
372	20.00	890.00	…	350.00	7196	…	601,988.1	…	3670	…	629,797.2	50.00

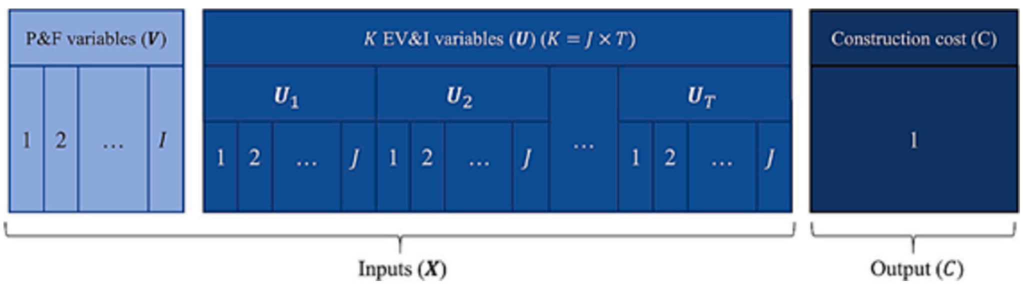

Fig. 9. Case 1 Data Structure [40].

Table 7
Case 1 Initial Model Parameters.

Model	Parameter	Value
NN	Number of Neurons	30
	Dropout Rate	0.2
	Learning Rate	0.01
	Epoch	500
GRU/BiGRU/LSTM/ BiLSTM	Number of Units	20
	Dropout Rate	0.2
	Learning Rate	0.01
	Epoch	500

summarized in Table 6, with the general raw data structure used in this case shown in Fig. 9.

4.1. Parameter setting and optimization

The value of the parameters in the base model, including neuron count for NN, unit number for GRU/BiGRU/LSTM/BiLSTM, dropout rate, number of epochs, and learning rate, were set to the same values, reflecting the best set of parameters from the Grid Search method. The parameters used for NN and GRU/BiGRU/LSTM/BiLSTM in this case study are shown in Table 7.

The parameter settings for the GRU and LSTM models were also applied to the BiGRU and BiLSTM models. In addition, the parameter values for both NN and GRU/LSTM were used for the hybrid model. Applying consistent parameter settings across all models helps ensure a fair and reliable comparison of model performance.

The optimal parameters for NN-BiGRU architecture parameters are shown in Table 8. This table shows the optimized value searched by OMA and Symbiotic Organism Search (SOS) [47] for each NN and BiGRU layer such as the number of neurons or units and dropout rate in the hidden layer, the number of neurons in the dense layer, and the NN-BiGRU learning rate.

4.2. Results and model comparison

After comparing OMA-NN-BiGRU against other hybrid and base models using the dataset for this case study, the evaluation metrics for regression are shown in Table 9. The hybrid models include OMA-NN-LSTM, SOS-NN-BiGRU, SOS-NN-LSTM, NN-BiGRU, and NN-LSTM, and the base models include NN, GRU, BiGRU, LSTM, and BiLSTM.

As shown in Table 9, OMA-NN-BiGRU not only predicted actual residential construction costs to a very high degree of precision but also outperformed all of the other models. The results also demonstrated the superiority of hybrid models in predicting complex datasets. In addition, based on the RI value, using OMA to optimize NN-BiGRU enhanced the level of prediction accuracy significantly from 0.949 to 0.977. A

Table 8
Case 1 Optimal NN-BiGRU Parameters Optimized by OMA and SOS.

Model	Layer	Parameter	Initial Value	OMA Optimal Value	SOS Optimal Value
NN	Hidden Layer	nn_neurons	30	63	46
		nn_dropout	0.2	0.0997	0.2192
	Dense Layer	nn_dense_neurons	5	2	8
BiGRU	Hidden Layer	bigru_units	20	10	23
		bigru_dropout	0.2	0.1029	0.320
	Dense Layer	bigru_dense_neurons	5	3	12
NN-BiGRU	–	lr	0.01	0.0204	0.0018

Table 9
Case 1 Comparison of Model Results.

Model		R	R^2	RMSE	MAE	MAPE	RI	Rank
Optimized Hybrid Deep Machine Learning	OMA-NN-BiGRU	0.995	0.989	0.012	0.009	0.076	0.977	1
	OMA-NN-LSTM	0.985	0.970	0.024	0.018	0.096	0.963	3
	SOS-NN-BiGRU	0.988	0.976	0.018	0.014	0.086	0.969	2
	SOS-NN-LSTM	0.981	0.963	0.023	0.017	0.104	0.960	4
Hybrid Deep Machine Learning	NN-BiGRU	0.980	0.961	0.028	0.021	0.149	0.949	5
	NN-LSTM	0.976	0.952	0.033	0.026	0.205	0.933	6
Bidirectional Primitive	BiGRU	0.955	0.911	0.050	0.041	0.277	0.900	7
	BiLSTM	0.942	0.888	0.056	0.043	0.305	0.886	8
Primitive Deep Machine Learning	GRU	0.928	0.862	0.053	0.044	0.372	0.864	10
	LSTM	0.918	0.843	0.068	0.057	0.440	0.839	11
	NN	0.928	0.862	0.053	0.044	0.372	0.864	9

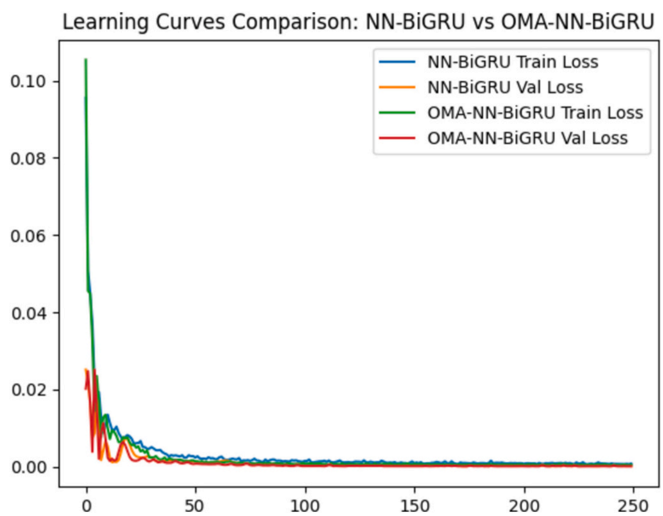

Fig. 10. Learning Curve Comparison (Case 1): NN-BiGRU vs OMA-NN-BiGRU.

similarly significant improvement was also reflected in the decline in MAPE value from 14.9 % for NN-BiGRU to 7.6 % for OMA-NN-BiGRU. In addition, when MSE is used as the objective function, the learning curve (shown in Fig. 10) shows the model optimized with OMA attained the minimum value faster than the model without optimization.

In comparison, the OMA algorithm demonstrated superiority over SOS on the same NN-BiGRU architecture, earning an RI index of 0.977 compared to 0.969 for SOS. The Convergence Graph (shown in Fig. 11) further illustrates OMA's advantage over the SOS algorithm in both the NN-BiGRU and NN-LSTM model structures. Furthermore, the proposed model demonstrated its superiority over the DBM-BPNN model [40] by earning a significantly lower MAPE (7.6 % vs. 9.7 %, respectively).

Case 2. Estimate Schedule to Completion

In the second case study, the estimated schedule to completion (ESTC) was predicted based on historical project data. The dataset used, collected from a construction company based in northern Taiwan, consisted of 11 projects completed between 2000 and 2007 [4]. Information on each project in this dataset is summarized in Table 10.

The data gathered included building and contracted specifications. The durations of these construction projects ranged from 457 to 749 days; the total floor areas ranged from 3094 to 31,797 square meters; and contracted project costs ranged from 85.7 million to 530 million Taiwan dollars. All of the projects in this case were reinforced concrete buildings. The objective of Case 2 was to predict the Estimate Schedule to Completion (ESTC) based on schedule progress as represented using each project's performance indices calculated using the Earned Value Management (EVM) technique. The historical cases for this case study were divided into building data and contract data.

The characteristics of the input data in this case study adopted the characteristics previously described in this study. The project physical and financial variables were used as the time-independent inputs for the NN, and the EVM variables were used as the time-dependent inputs for the BiGRU. The input variables and their respective categories are shown in Table 11.

The historical samples used in this case study are outlined in

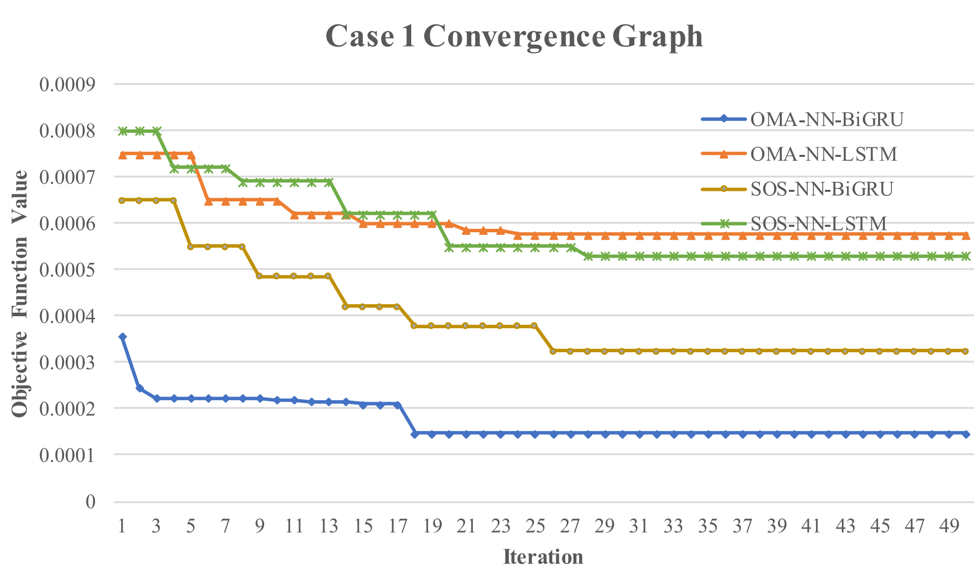

Fig. 11. Case 1 Convergence Graph.

Table 10
Summarized Each Project Information in Case 2 Dataset.

ID	Building Data				Contract Data				Number of Periods
	Total Floor Area (m^2)	Underground Floors	Above- Ground Floors	Buildings	Project Start Date	Project End Date	Duration (Days)	Contract Amount (including tax)	
A	12,622	2	9	1	2003/12/01	2005/08/22	630	289,992,000	24
B	4919	3	11	1	2003/12/13	2005/11/10	698	149,300,000	21
C	31,797	2	9	4	2001/07/04	2003/03/31	635	530,000,000	19
D	7707	2	14	1	2001/11/24	2003/10/20	695	153,500,000	22
E	10,087	3	14	1	2002/06/18	2004/07/06	749	216,000,000	22
F	3479	1	10	1	2003/06/02	2004/09/30	486	85,714,286	17
G	6352	4	11	1	2004/03/05	2006/02/18	715	202,241,810	26
H	4774	2	11	1	2004/02/21	2006/02/20	730	145,377,589	25
I	3986	2	12	1	2004/04/24	2005/10/19	543	102,500,000	16
J	7289	2	8	1	2005/06/15	2006/09/15	457	190,844,707	20
K	3094	2	7	1	2005/10/01	2007/02/28	515	102,500,000	14
							Total		226

Table 11
Case 2 Input Variables.

Factor	Description	Category	
		Time-independent	Time-dependent
Total floor area	Total floor area	●	
Number of floors	Total number of floors	●	
Construction cost	Total amount of the contracted cost	●	
Labor market	Rate of joining the labor force – the rate of leaving the labor force	●	
Productivity	Construction productivity index	●	
Material prices	Construction material price index	●	
Weather	Cumulative rainy days/ contract duration	●	
Percent change in contract amount	New contract amount/ original contract amount		●
Actual duration (AD)	Cumulative work duration/contract duration		●
Percent of scheduled work completed	ΣPV/BAC		●
Percent of actual work completed	ΣEV/BAC		●
Percent of budget spent	ΣAC/BAC		●
Schedule Performance (SPI)	EV/PV		●
Cost Performance (CPI)	EV/AC		●

Table 12.

The data in Table 12 was derived from Cheng et al. (2019) [4] and follows the EVM method, a technique commonly used in the construction industry to identify variances by comparing planned versus actual project performance. Because the duration of different projects can vary widely, using day as the unit for ESTC does not facilitate comparison. Thus, in this study, ESTC unit is converted from day to a percentage using the following equation: remaining duration to completion divided by total contract duration.

Finally, a fixed number of timesteps is required for data reconstruction to prepare the time-dependent input variables for the BiGRU model. In Case 2, three timesteps (Quarterly Intervals) were used, meaning that 3 previous time-dependent variable datapoints were considered in predicting the output. In addition, in comparison with previous research by Cheng et al. [4], the authors identified a potential issue with the k-fold cross-validation approach employed when dealing with time-dependent data, finding it may cause overfitting and bias in the evaluation process [48]. To avoid this problem, the OOP technique [45] was used to split the data into training, validation, and testing datasets with respective percentages of 60, 20, and 20.

4.3. Parameter setting and optimization

Each of the base model parameters were set to the same values for number of neurons (for NN), number of units (for GRU/BiGRU/LSTM/BiLSTM), dropout rate, number of epochs, and learning rate, using the best set of parameters from the Grid Search method. The parameters used for this case study are shown in Table 13 for NN and GRU/BiGRU/LSTM/BiLSTM.

The parameter settings used for the GRU and LSTM models were also applied to the BiGRU and BiLSTM models. In addition, the parameter values for both NN and GRU/LSTM were applied to the hybrid models. Applying consistent parameter settings across models helps ensure a fair and reliable comparison of their performance.

The optimal parameters for NN-BiGRU architecture parameters are shown in Table 14. The optimized value searched by OMA and SOS for each NN and BiGRU layer such as number of neurons or units and dropout rate in the hidden layer, and number of neurons in the dense layer, and the NN-BiGRU learning rate.

4.4. Results and model comparison

The evaluation metrics for regression after comparing OMA-NN-BiGRU with the other hybrid (OMA-NN-LSTM, SOS-NN-BiGRU, SOS-NN-LSTM, NN-BiGRU, and NN-LSTM) and base (NN, GRU, BiGRU, LSTM, and BiLSTM) models using the case study dataset are shown in Table 15.

As shown in Table 15, OMA-NN-BiGRU not only predicted the value of ESTC to a very high degree of precision but also outperformed all of the other models. Also, the results again demonstrated the superiority of hybrid models in predicting complex datasets. A similarly significant improvement was also reflected in the decline in MAPE value from 20.2 % for NN-BiGRU to 12.4 % for OMA-NN-BiGRU. Moreover, the learning curve (shown in Fig. 12), using MSE as the objective function indicates that the OMA-optimized model attained the minimum value faster than the non-optimized model, effectively avoiding overfitting. In contrast, the NN-BiGRU model shows signs of overfitting.

In case 2, the OMA algorithm once again outperformed SOS on the same NN-BiGRU architecture, earning an RI index of 0.932 compared to 0.917 for SOS. The Convergence Graph (shown in Fig. 13) further highlights OMA's superiority over the SOS algorithm in both the NN-BiGRU and NN-LSTM model structures.

Table 12
Case 2 Historical Cases.

No	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Item	Total floor area (m^2)	Number of floors	Contract amount (1000 * NTD)	Rate of change in the Labor market (%)	Productivity index	Project price index	% of Cumulative rainy days	% change in contract amount Percentage	Cumulative work duration in %	% of Scheduled work completed	% of actual work completed	% of budgets spent	Schedule performance (SPI)	Cost performance (CPI)	Remaining completion period/ contract duration (%)
1	12,622	11	289,992	−1.30	108.95	67.71	0.01	1.00	0.08	0.05	0.04	0.02	0.69	2.01	0.96
2	12,622	11	289,992	−0.61	108.95	72.91	0.02	1.00	0.12	0.09	0.08	0.03	0.89	2.24	0.92
3	12,622	11	289,992	−0.37	108.95	75.73	0.03	1.00	0.16	0.10	0.10	0.05	0.99	2.05	0.88
224	3094	9	102,500	00.75	85.05	83.57	0.16	1.00	0.77	0.71	0.74	0.63	1.05	1.18	0.35
225	3094	9	102,500	00.67	85.05	84.15	0.17	1.00	0.88	0.91	0.87	0.73	0.95	1.18	0.23
226	3094	9	102,500	−1.58	101.71	85.79	0.18	1.00	1.00	0.98	0.98	0.78	1.00	1.26	0.12

Table 13
Case 2 Initial Model Parameters.

Model	Parameter	Value
NN	Number of Neurons	20
	Dropout Rate	0.2
	Learning Rate	0.01
	Epoch	500
GRU/BiGRU/LSTM/BiLSTM	Number of Units	70
	Dropout Rate	0.2
	Learning Rate	0.01
	Epoch	500

5. Conclusions and recommendations

This paper combined NN and BiGRU techniques to improve prediction accuracy in time-related engineering problems. Moreover, OMA was used as the optimization algorithm to fine-tune the parameters of the NN-BiGRU model. OMA optimization of NN-BiGRU architecture parameters together with the replacement of output weights with a trial-and-error approach achieved optimal results. The performance of the proposed model was then compared with other base and hybrid models such as SOS-NN-LSTM, NN-LSTM, and NN-BiGRU. The results of this comparison were ranked by RI value, which was calculated as the average of several evaluation metrics, including RMSE, MAE, MAPE, R, and R^2.

The performance and generalizability of the proposed model was tested on two construction engineering case studies involving, respectively, residential building construction and reinforced concrete building projects. In terms of estimating construction cost, the proposed model earned an RMSE = 0.012, MAE = 0.009, MAPE = 7.63 %, R = 0.995 and R^2 = 0.989, with an RI index of 0.977. In terms of predicting schedule to completion, the proposed model earned an RMSE = 0.061, MAE = 0.051, MAPE = 12.4 %, R = 0.965, and R^2 = 0.931, with an RI index of 0.932.

The high level of accuracy achieved by the OMA-NN-BiGRU model in these case studies highlights its utility in delivering reliable and actionable insights for construction management. In Case 1, the model provided reliable cost estimates, empowering project managers to make data-informed decisions in bidding and early-stage planning and helping align projected budgets with financial requirements early in the project lifecycle. In Case 2, the prediction results generated by the model may be used with EVM metrics to calculate Estimate Schedule at Completion (ESAC), allowing project managers to assess estimated versus planned schedules in real-time. This approach enhances the ability of project managers to monitor project performance in real-time to predict the potential for scheduling issues that may delay project completion. In summary, the case studies demonstrated the OMA-NN-BiGRU model to be a practical, dynamic tool for ensuring projects remain on track, enabling proactive responses and actions such as schedule replanning, schedule crashes, and resource allocations to fulfill project needs and achieve project success.

Based on the results of these case studies, a key contribution of this research was the development of the OMA-NN-BiGRU, which harnesses the complementary strengths of NN in processing complex feature relationships in time-independent data and BiGRU in capturing temporal dependencies within the construction domain. By integrating these powerful techniques and leveraging OMA's optimization approach, the developed hybrid model achieves enhanced accuracy and generalizability, positioning it as a robust tool for data-driven decision-making. The practical nature and reliability of the model were demonstrated on a randomly selected 20 % of the data (the last 75 residential building projects in Case 1 and projects I, J, and K in Case 2). The performance of the developed model on this test dataset highlighted its suitability for use in active project management and real-time control applications.

The findings regarding the effectiveness and superiority of the hybrid model in addressing complex problems and capturing intricate patterns

Table 14
Case 2 Optimal NN-BiGRU Parameters Optimized by OMA and SOS.

Model	Layer	Parameter	Initial Value	OMA Optimal Value	SOS Optimal Value
NN	Hidden Layer	nn_neurons	20	28	45
		nn_dropout	0.1	0	0.1691
	Dense Layer	nn_dense_neurons	5	9	10
BiGRU	Hidden Layer	bigru_units	70	53	32
		bigru_dropout	0.2	0.1174	0.2395
	Dense Layer	bigru_dense_neurons	5	3	9
NN-BiGRU	–	lr	0.01	0.0378	0.0068

Table 15
Case 2 Results and Model Comparison.

Model		R	R^2	RMSE	MAE	MAPE	RI	Rank
Optimized Hybrid Deep Machine Learning	OMA-NN-BiGRU	0.965	0.931	0.061	0.051	0.124	0.932	1
	OMA-NN-LSTM	0.946	0.895	0.081	0.069	0.176	0.903	3
	SOS-NN-BiGRU	0.954	0.910	0.073	0.059	0.149	0.917	2
	SOS-NN-LSTM	0.945	0.893	0.083	0.065	0.194	0.899	4
Hybrid Deep Machine Learning	NN-BiGRU	0.932	0.869	0.091	0.075	0.202	0.887	5
	NN-LSTM	0.913	0.834	0.094	0.076	0.218	0.861	6
Bidirectional Primitive	BiGRU	0.884	0.773	0.142	0.122	0.326	0.813	8
	BiLSTM	0.863	0.745	0.158	0.135	0.349	0.793	10
Primitive Deep Machine Learning	GRU	0.874	0.763	0.143	0.123	0.328	0.808	9
	LSTM	0.815	0.664	0.148	0.121	0.339	0.774	11
	NN	0.869	0.756	0.101	0.086	0.228	0.842	7

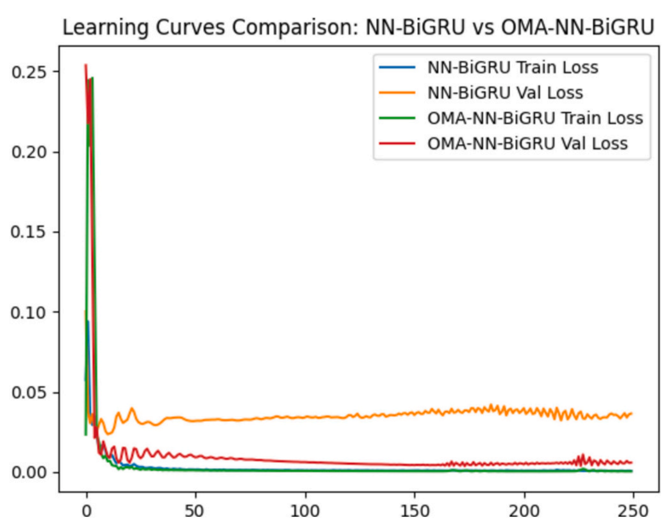

Fig. 12. Learning Curve Comparison (Case 2): NN-BiGRU vs OMA-NN-BiGRU.

in the data highlight the potential for this model to provide construction project managers with accurate, data-driven insights for decision-making. Accurate prediction models empower construction engineering stakeholders to make informed decisions, leading to improved project planning, enhanced risk management, cost optimization, efficient resource allocation, increased productivity, the promotion of sustainable construction practices, and, ultimately, improved efficiency and profitability in the construction industry.

The proposed method demonstrated great performance across two distinct case studies, proving its ability to generate accurate predictions and its generalizability. However, due to time and effort limitations, the performance of the proposed model was only tested on two case studies in the construction industry. Future work should further validate its superior performance on other case studies in the construction and other industry sectors. In addition, another weakness of this study is that the performance of the model on classification tasks was not investigated, which may be addressed in future related research. Also, further adjustments to the model architecture such as adding a CNN layer to the NN's hidden layer and/or to the GRU's hidden layer to improve feature extraction capabilities for highly complex datasets are recommended. However, one drawback of using this black-box model is that it does not provide insights into the impact of each input variable on the output. Future research should explore methods to identify the discrete contributions of input variables to model outcomes.

Furthermore, determining whether variables are time-dependent or time-independent still requires the guidance/advice of related experts in the field of construction management due to the inherent complexity of data in this industry. This study was developed using the Critical Path Method (CPM) for project scheduling, which assumes activity duration to be a constant value. Probability and confidence interval prediction may be explored more thoroughly in future research using the Program Evaluation and Review Technique (PERT) to evaluate the various project stages (e.g., early, middle, late). Incorporating PERT factors may be expected to facilitate a more-nuanced understanding of project progression and offer additional valuable insights beyond the capabilities of CPM-based approaches.

The need to collect comprehensive building data and information is a significant practical challenge to implementing the proposed method in construction management. Diversifying the dataset by incorporating a wider range of building sizes and types would enable future studies to be more comprehensive in scope and facilitate the development of more robust models applicable to various types of construction projects. Another crucial factor that will influence the future development and use of this method is the ability to appropriately select variables for the data. This may be achieved through the application of feature selection methods. Lastly, it will be essential to identify whether variables are time-dependent or time-independent and to account for this distinction in the modeling process.

CRediT authorship contribution statement

Min-Yuan Cheng: Writing – review & editing, Supervision, Resources, Methodology, Investigation, Conceptualization. **Quoc-Tuan Vu:** Writing – review & editing, Visualization, Validation, Software, Formal analysis, Data curation. **Frederik Elly Gosal:** Writing – original draft, Visualization, Software, Methodology, Investigation, Conceptualization.

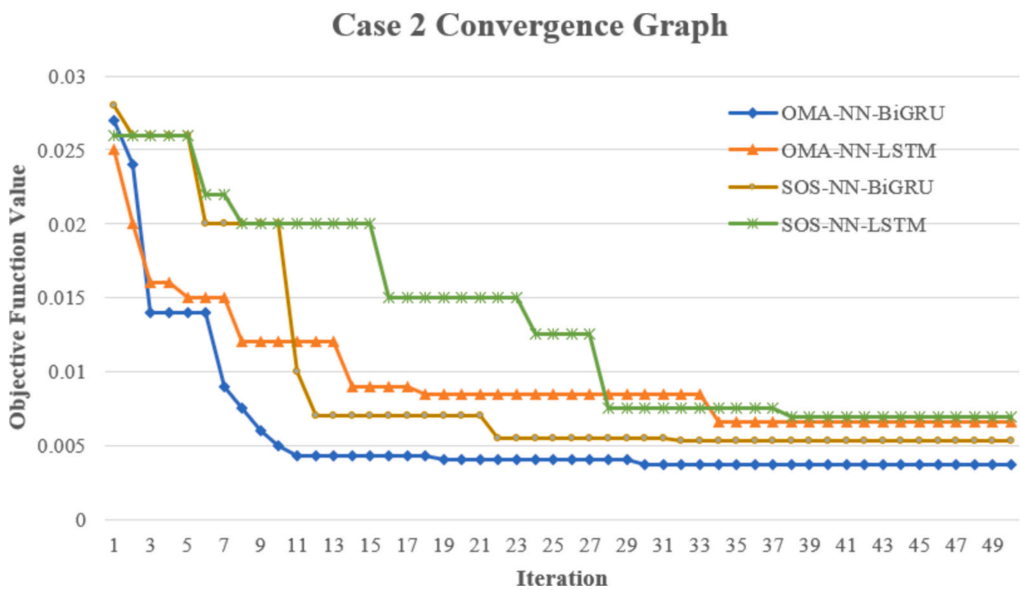

Fig. 13. Case 2 Convergence Graph.

Declaration of competing interest

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

Data availability

The materials generated and analyzed in this study are available at https://www.researchgate.net/profile/Min-Yuan-Cheng-2 and from the corresponding author upon reasonable request.

References

[1] A.P. Chan, D. Scott, A.P. Chan, Factors affecting the success of a construction project, J. Constr. Eng. Manag. 130 (1) (2004) 153–155, https://doi.org/10.1061/(ASCE)0733-9364(2004)130:1(153).

[2] A.K. Munns, B.F. Bjeirmi, The role of project management in achieving project success, Int. J. Proj. Manag. 14 (2) (1996) 81–87, https://doi.org/10.1016/0263-7863(95)00057-7.

[3] R.F. Aziz, A.A. Abdel-Hakam, Exploring delay causes of road construction projects in Egypt, Alex. Eng. J. 55 (2) (2016) 1515–1539, https://doi.org/10.1016/j.aej.2016.03.006.

[4] M.-Y. Cheng, Y.-H. Chang, D. Korir, Novel approach to estimating schedule to completion in construction projects using sequence and nonsequence learning, J. Constr. Eng. Manag. 145 (11) (2019) 04019072, https://doi.org/10.1061/(ASCE)CO.1943-7862.0001697.

[5] J.-B. Yang, P.-R. Wei, Causes of delay in the planning and design phases for construction projects, J. Archit. Eng. 16 (2) (2010) 80–83, https://doi.org/10.1061/(ASCE)1076-0431(2010)16:2(80).

[6] M.M. Marzouk, T.I. El-Rasas, Analyzing delay causes in Egyptian construction projects, J. Adv. Res. 5 (1) (2014) 49–55, https://doi.org/10.1016/j.jare.2012.11.005.

[7] M.-Y. Cheng, H.-S. Peng, Y.-W. Wu, T.-L. Chen, Estimate at completion for construction projects using evolutionary support vector machine inference model, Autom. Constr. 19 (5) (2010) 619–629, https://doi.org/10.1016/j.autcon.2010.02.008.

[8] L. Liu, K. Zhu, Improving cost estimates of construction projects using phased cost factors, J. Constr. Eng. Manag. 133 (1) (2007) 91–95, https://doi.org/10.1061/(ASCE)0733-9364(2007)133:1(91).

[9] G.-H. Kim, S.-H. An, K.-I. Kang, Comparison of construction cost estimating models based on regression analysis, neural networks, and case-based reasoning, Build. Environ. 39 (10) (2004) 1235–1242, https://doi.org/10.1016/j.buildenv.2004.02.013.

[10] D. Myers, Construction Economics: A New approach, Routledge, 2022, https://doi.org/10.1201/9781003287513.

[11] M.H. Rafiei, H. Adeli, A novel machine learning model for estimation of sale prices of real estate units, J. Constr. Eng. Manag. 142 (2) (2016) 04015066, https://doi.org/10.1061/(ASCE)CO.1943-7862.0001047.

[12] M.-Y. Cheng, Y.-W. Wu, C.-C. Huang, Hybrid Gaussian process inference model for construction management decision making, Int. J. Inf. Technol. Decis. Mak. 19 (04) (2020) 1015–1036, https://doi.org/10.1142/s0219622020500212.

[13] F.E. Gosal, E. Tambunan, Effect of side frictions on level of service of DI Panjaitan street in Cawang - East Jakarta, IOP Conf. Ser. Earth Environ. Sci. 878 (1) (2021) 012048, https://doi.org/10.1088/1755-1315/878/1/012048.

[14] S. Shoar, N. Chileshe, J.D. Edwards, Machine learning-aided engineering services' cost overruns prediction in high-rise residential building projects: application of random forest regression, J. Build. Eng. 50 (2022) 104102, https://doi.org/10.1016/j.jobe.2022.104102.

[15] A. Tsanas, A. Xifara, Accurate quantitative estimation of energy performance of residential buildings using statistical machine learning tools, Energ. Buildings 49 (2012) 560–567, https://doi.org/10.1016/j.enbuild.2012.03.003.

[16] M.A. Musarat, W.S. Alaloul, M.S. Liew, Impact of inflation rate on construction projects budget: a review, Ain Shams Eng. J. 12 (1) (2021) 407–414, https://doi.org/10.1016/j.asej.2020.04.009.

[17] S.O. Abioye, L.O. Oyedele, L. Akanbi, A. Ajayi, J.M. Davila Delgado, M. Bilal, O. O. Akinade, A. Ahmed, Artificial intelligence in the construction industry: a review of present status, opportunities and future challenges, J. Build. Eng. 44 (2021) 103299, https://doi.org/10.1016/j.jobe.2021.103299.

[18] L. Medsker, L.C. Jain, Recurrent Neural Networks: Design and Applications, 1st ed., CRC press, 1999 https://doi.org/10.1201/9781003040620.

[19] D.E. Rumelhart, G.E. Hinton, R.J. Williams, Learning representations by back-propagating errors (nature, no. 6088), 1986, pp. 533–536, https://doi.org/10.1038/323533a0.

[20] K. Cho, B. Van Merriënboer, D. Bahdanau, Y. Bengio, On the properties of neural machine translation: Encoder-decoder approaches, arXiv preprint (2014), https://doi.org/10.48550/arXiv.1409.1259 arXiv:1409.1259.

[21] X. Meng, T. Zhu, C. Li, Construction of perfect dispatch learning model based on adaptive GRU, Energy Rep. 8 (2022) 668–677, https://doi.org/10.1016/j.egyr.2022.02.250.

[22] J. Zhou, Y. Qin, D. Chen, F. Liu, Q. Qian, Remaining useful life prediction of bearings by a new reinforced memory GRU network, Adv. Eng. Inform. 53 (2022) 101682, https://doi.org/10.1016/j.aei.2022.101682.

[23] M. Schuster, K.K. Paliwal, Bidirectional recurrent neural networks, IEEE Trans. Signal Process. 45 (11) (1997) 2673–2681, https://doi.org/10.1109/78.650093.

[24] M.-Y. Cheng, M.N. Sholeh, Optical microscope algorithm: a new metaheuristic inspired by microscope magnification for solving engineering optimization problems, Knowl.-Based Syst. (2023) 110939, https://doi.org/10.1016/j.knosys.2023.110939.

[25] B.-C. Kim, K.F. Reinschmidt, Combination of project cost forecasts in earned value management, J. Constr. Eng. Manag. 137 (11) (2011) 958–966, https://doi.org/10.1061/(ASCE)CO.1943-7862.0000352.

[26] P.A. de Andrade, A. Martens, M. Vanhoucke, Using real project schedule data to compare earned schedule and earned duration management project time forecasting capabilities, Autom. Constr. 99 (2019) 68–78, https://doi.org/10.1016/j.autcon.2018.11.030.

[27] N. Thomas, A.V. Thomas, Regression modelling for prediction of construction cost and duration, Appl. Mech. Mater. 857 (2017) 195–199, https://doi.org/10.4028/www.scientific.net/AMM.857.195.

[28] F.M. Al-Zwainy, "a state-of-the-art survey to estimate construction costs in highway and bridge projects: analytical diagnostic study," *international journal of civil*, Eng. Technol. 9 (5) (2018) 795–821 [Online]. Available: http://iaeme.com/Home/issue/IJCIET?Volume=9&Issue=5.

[29] E. Plebankiewicz, Model of predicting cost overrun in construction projects, Sustainability 10 (12) (2018) 4387, https://doi.org/10.3390/su10124387.
[30] S. Nadafi, S.H. Moosavirad, S. Ariafar, Predicting the project time and costs using EVM based on gray numbers, Eng. Constr. Archit. Manag. 26 (9) (2019) 2107–2119, https://doi.org/10.1108/ECAM-07-2018-0291.
[31] A.E. Abu El-Maaty, A.M. El-Kholy, A.Y. Akal, Modeling schedule overrun and cost escalation percentages of highway projects using fuzzy approach, Eng. Constr. Archit. Manag. 24 (5) (2017) 809–827, https://doi.org/10.1108/ECAM-03-2016-0084.
[32] D. Chakraborty, H. Elhegazy, H. Elzarka, L. Gutierrez, A novel construction cost prediction model using hybrid natural and light gradient boosting, Adv. Eng. Inform. 46 (2020) 101201, https://doi.org/10.1016/j.aei.2020.101201.
[33] Y. Li, L. Liu, Hybrid artificial neural network and statistical model for forecasting project total duration in earned value management, Int. J. Networking Virtual Organisations 10 (3–4) (2012) 402–413, https://doi.org/10.1504/IJNVO.2012.046460.
[34] M.G. Naik, V. Radhika, Time and cost analysis for highway road construction project using artificial neural networks, J. Construct. Eng. Project Manag. 5 (1) (2015) 26–31, https://doi.org/10.6106/JCEPM.2015.5.1.026.
[35] I. Peško, V. Mučenski, M. Šešlija, N. Radović, A. Vujkov, D. Bibić, M. Krklješ, Estimation of costs and durations of construction of urban roads using ANN and SVM, Complexity 2017 (2017), https://doi.org/10.1155/2017/2450370.
[36] A. Balali, A. Valipour, J. Antucheviciene, J. Šaparauskas, Improving the results of the earned value management technique using artificial neural networks in construction projects, Symmetry 12 (10) (2020) 1745, https://doi.org/10.3390/sym12101745.
[37] F.K. Jaber, N.A. Jasim, F.M. Al-Zwainy, Forecasting techniques in construction industry: earned value indicators and performance models, Przegląd Naukowy. Inżynieria i Kształtowanie Środowiska 29 (2) (2020), https://doi.org/10.22630/PNIKS.2020.29.2.20, 88.
[38] J.A. Ujong, E.M. Mbadike, G.U. Alaneme, Prediction of cost and duration of building construction using artificial neural network, Asian J. Civ. Eng. 23 (7) (2022) 1117–1139, https://doi.org/10.1007/s42107-022-00474-4.
[39] M.-Y. Cheng, N.-D. Hoang, Interval estimation of construction cost at completion using least squares support vector machine, J. Civ. Eng. Manag. 20 (2) (2014) 223–236, https://doi.org/10.3846/13923730.2013.801891.
[40] M.H. Rafiei, H. Adeli, Novel machine-learning model for estimating construction costs considering economic variables and indexes, J. Constr. Eng. Manag. 144 (12) (2018) 04018106, https://doi.org/10.1061/(ASCE)CO.1943-7862.0001570.
[41] F. Rosenblatt, The perceptron: a probabilistic model for information storage and organization in the brain, Psychol. Rev. 65 (6) (1958) 386–408, https://doi.org/10.1037/h0042519.
[42] C.M. Berners-Lee, Cybernetics and forecasting, Nature 219 (5150) (1968) 202–203, https://doi.org/10.1038/219202b0.
[43] S. Amari, A theory of adaptive pattern classifiers, IEEE Transact. Electronic Computers 3 (1967) 299–307, https://doi.org/10.1109/PGEC.1967.264666.
[44] M. Rezaei, A. Mohammadifar, H. Gholami, M. Mina, M.J.P.M. Riksen, C. Ritsema, Mapping of the wind erodible fraction of soil by bidirectional gated recurrent unit (BiGRU) and bidirectional recurrent neural network (BiRNN) deep learning models, CATENA 223 (2023) 106953, https://doi.org/10.1016/j.catena.2023.106953.
[45] V. Cerqueira, L. Torgo, J. Smailović, I. Mozetič, A comparative study of performance estimation methods for time series forecasting, in: 2017 IEEE International Conference on Data Science and Advanced Analytics (DSAA), IEEE, 2017, pp. 529–538, https://doi.org/10.1109/DSAA.2017.7.
[46] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, R. Salakhutdinov, Dropout: a simple way to prevent neural networks from overfitting, J. Mach. Learn. Res. 15 (1) (2014) 1929–1958, https://doi.org/10.5555/2627435.2670313.
[47] M.-Y. Cheng, D. Prayogo, Symbiotic organisms search: a new metaheuristic optimization algorithm, Comput. Struct. 139 (2014) 98–112, https://doi.org/10.1016/j.compstruc.2014.03.007.
[48] C. Bergmeir, R.J. Hyndman, B. Koo, A note on the validity of cross-validation for evaluating autoregressive time series prediction, Comput. Statistics Data Analysis 120 (2018) 70–83, https://doi.org/10.1016/j.csda.2017.11.003.