DEVELOPMENT OF A MACHINE LEARNING MODEL FOR IN-MOTION TRAIN WEIGHT ESTIMATION ON RAILWAY BRIDGES

Tran Le Anh Duc, Phan Hoang Nam*

The University of Danang – University of Science and Technology, Vietnam

*Corresponding author: phnam@dut.udn.vn

(Received: April 24, 2025; Revised: June 11, 2025; Accepted: June 19, 2025)

DOI: 10.31130/ud-jst.2025.23(10C).667E

Abstract - This paper presents a fast, accurate approach for estimating train weight during bridge crossings by coupling a validated train–track–bridge dynamic interaction model with an XGBoost regressor optimized via Bayesian hyperparameter tuning. A high-speed railway bridge in China is used as a case study. We generate a dataset of 1000 labeled samples from finite-element dynamic simulations driven by Latin Hypercube Sampling, and validate the interaction model against available experimental data. The learned model maps measured bridge responses to train loads and is suitable for real-time deployment. Results show high predictive accuracy, with coefficients of determination $R^2 \approx 0.996$ for motor cars and 0.993 for trailer cars. Gain-based feature importance indicates that the bridge's maximum vertical displacement during passage is the most influential input, substantially outweighing other variables. The proposed framework demonstrates a practical path toward in-service train load estimation and monitoring without intrusive onboard sensors or extensive instrumentation.

Key words - High-speed railway; weight-in-motion; XGBoosting; Latin Hypercube Sampling; Bayesian Optimization

1. Introduction

Railways play a vital role in modern passenger and freight transportation systems due to their efficiency, capacity, and reliability, making them a preferred mode of transport in many countries. As global economies continue to grow, particularly in both developed and developing nations, the demand for rail transport is rapidly increasing. Consequently, the scale of passenger and freight operations has expanded significantly.

To maximize operational efficiency and profit, freight trains are often loaded to their maximum capacity. However, this practice frequently results in overloading, which can lead to severe consequences. Prolonged operation under excessive loads can damage railway infrastructure, elevate derailment risk, and compromise operational safety. In addition to safety hazards, overloading may also result in substantial economic losses. Therefore, the need for effective and accurate solutions to monitor and control train load is of great interest to researchers, engineers, and rail operators.

Several commercial systems have been developed to detect and estimate train loads, including solutions that use fiber-optic technology [1] or indirect measurement devices that record force and displacement at contact points during train passage [2]. These systems often rely on Weigh-In-Motion (WIM) algorithms [1] to estimate axle loads, car weights, and total train weights. Research by Fernando Marques et al. [3] combined existing B-WIM algorithms, particularly the Moses method based on collected strain data, to estimate axle loads, axle spacing, and train speed. This approach was optimized using a genetic algorithm to minimize errors between measured and simulated structural responses, thereby enhancing the reliability of the assessment. Aleš Žnidarič et al. [4] also proposed research on the development and testing of a dynamic train weighing system while in motion on a bridge by adjusting and modifying the standard BWIM algorithm to allow each carriage to have a different velocity. Despite these advancements, accurately and continuously estimating static wheel, axle, and wagon loads during train movement remains challenging, limiting the precision of current monitoring systems under operational conditions.

To address these limitations, there is a need to develop alternative solutions capable of continuously monitoring and analyzing train loads with high precision and in real-time, even under variable speed conditions and environmental disturbances. In response to this need, the present study proposes and develops a train load estimation model using machine learning techniques to enhance load monitoring and predict permissible overloads as trains cross bridges. By leveraging machine learning algorithms, real-world sensor data can be analyzed with higher accuracy, facilitating precise train load estimation under complex operational scenarios.

2. Train - rail - bridge interaction model

2.1. Case study

In this study, the Gouhe Bridge with simply supported girders (continuous track) and the Pioneer EMU (Electric Multiple Unit) train operating on the Qin-Shen High-Speed Railway (Qinhuangdao–Shenyang) are analyzed using the proposed model. The results are validated by comparing the bridge's dynamic response under train-track-bridge coupled system analysis with experimental data from Xia et al. [5]. The general layout of the bridge and train track is shown in Figure 1.

The Gouhe Bridge in Northeast China consists of 28 simple spans of 24 m with a typical cross-section of a box section with a double-track ballastless deck. Each box girder is 24.6 m long with a deck width of 12.4 m, a girder bottom width of 6.08 m, and a height of 2.2 m. The bridge is designed for a maximum speed of 250 km/h and an operating speed of 200 km/h. The cross-section and main dimensions of the girders are shown in Figure 2.

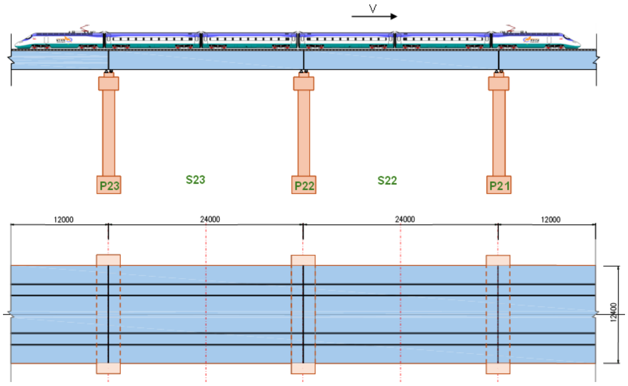


Figure 1. *General layout of Gouhe Bridge*

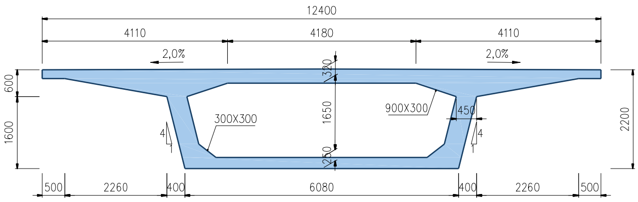


Figure 2. *Main beam cross-section*

The train under consideration in this case study is a Pioneer EMU consisting of 6 cars, of which cars 1, 2, 4, and 6 are motor cars and cars 2 and 5 are trailer cars. The composition of the first 3 cars is shown in Figure 3.

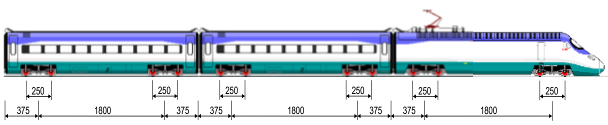


Figure 3. *Pioneer EMU train*

2.2. Two-way train-rail-bridge interaction model

This study employs a 2D coupled finite element model to analyze the dynamic interaction between train, track, and bridge systems. The model evaluates bridge response under train loads for ballastless track systems [6, 7]. In which the train is represented as a multi-vehicle chain using a mass-spring-damper system. Meanwhile, the track system is modeled as a continuous beam supported on a spring-damper foundation, and the bridge is represented as a 2D simply-supported Euler-Bernoulli beam. All substructures of the model are coupled into a single system. The equations of motion of the train and the infrastructure are described by the mass, stiffness, damping, and force matrices. The Hertzian contact theory used in the 2D TTBI model allows the contact points of the contact loss simulation to describe the interaction forces between the wheel and the track irregularities.

The train-track-bridge interaction is a combination of subsystems defined by the equation of motion:

$$\boldsymbol{M}_x\ddot{\boldsymbol{u}}_x + \boldsymbol{C}_x\dot{\boldsymbol{u}}_x + \boldsymbol{K}_x\boldsymbol{u}_x = \boldsymbol{F}_x \quad (1)$$

where $\boldsymbol{M}$, $\boldsymbol{C}$, $\boldsymbol{K}$ are the matrices of mass distribution along the length of the beam, viscous damping, and stiffness, respectively; $\boldsymbol{F}$ are vectors of modal forces and external forces; $\ddot{\boldsymbol{u}}$, $\dot{\boldsymbol{u}}$, $\boldsymbol{u}$ are the acceleration, velocity, and displacement vectors, respectively; while x denotes the subsystem to which the matrix refers and can be replaced by V, T, and B to denote the subsystems as vehicle, track, and bridge, respectively. The coupled equation of motion of this interactive simulation [8] is written in block matrix form as follows:

$$\begin{pmatrix} M_V & 0 & 0 \\ 0 & M_T & 0 \\ 0 & 0 & M_B \end{pmatrix}\begin{Bmatrix} \ddot{u}_V \\ \ddot{u}_T \\ \ddot{u}_B \end{Bmatrix} + \begin{pmatrix} C_V & C_{V,T} & 0 \\ C_{T,V} & C_T & C_{T,B} \\ 0 & C_{B,T} & C_B \end{pmatrix}\begin{Bmatrix} \dot{u}_V \\ \dot{u}_T \\ \dot{u}_B \end{Bmatrix} + \begin{pmatrix} K_V & K_{V,T} & 0 \\ K_{T,V} & K_T & K_{T,B} \\ 0 & K_{B,T} & K_B \end{pmatrix}\begin{Bmatrix} u_V \\ u_T \\ u_B \end{Bmatrix} = \begin{Bmatrix} F_V \\ F_T \\ F_B \end{Bmatrix} \quad (2)$$

The train–rail–bridge interaction model and the coupling of the subsystems through their equations of motion are implemented in MATLAB. The developed MATLAB program enables the efficient computation of the system matrices using advanced algorithms, ensuring high accuracy and computational efficiency. Furthermore, the wheel–rail interaction forces are described using the nonlinear Hertzian contact theory [9] in the following equation:

$$F = \begin{cases} k_C(u_w - u_r - r_w), & (u_w - u_r - r_w) > 0 \\ 0, & (u_w - u_r - r_w) \le 0 \end{cases} \quad (3)$$

where F is the interaction force between the wheel and the rail, k_C, u_w, u_r, r_w are the Hertzian interaction stiffness constant, the vertical displacement of the wheel, the vertical displacement of the rail, and the surface irregularity of the rail surface, respectively.

The random track irregularities are quantitatively characterized through Power Spectral Density (PSD) functions. While multiple standardized PSD formulations exist across different railway administrations, this study adopts the Federal Railroad Administration spectrum, which is mathematically expressed as:

$$S(\omega) = \frac{A_v\omega_2^2(\omega^2 + \omega_1^2)}{\omega^4(\omega^2 + \omega_2^2)} \quad (4)$$

Detailed PSD calculation procedures and implementation examples can be found in the works of Du Kim and Warnitchai [10] and Ferrara [11].

2.2.1. Train model

The train is represented by a finite element model, defined by the combination of individual sub-assemblies, including the car body with 2 degrees of freedom, the bogie, and the wheels, both with 4 degrees of freedom. The model of the train running on a random track irregularity is shown in Figure 4.

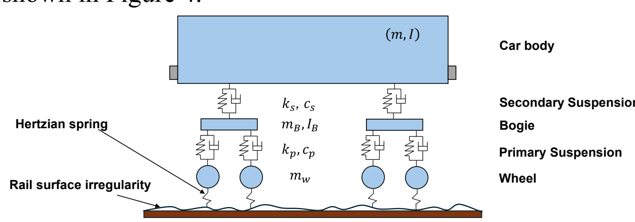


Figure 4. *Train model*

The geometrical and mechanical input parameters of the train model depend on the specific train being modeled. The parameters of the train model are detailed in Table 2. The main body of the train is modeled as a rigid bar with masses $(m_v)$and moments of inertia (I_v). The main and secondary suspensions are represented by a system of linear springs (k_p, k_s) and viscous dampers (c_p, c_s) in parallel. The wheels are specifically modeled as masses on rails connected to the bogies by the main suspension.

Table 1. *Train model parameter table*

Parameter	Symbol	Unit
Main body mass	m_v	kG
Moment of inertia of the main body	I_v	$kG.m^2$
Main body length	L_v	m
Additional segment length	L_E	m
Number of redirections	N_b	
Mass of bogie	m_B	kG
Moment of inertia of the bogie	I_B	$kG.m^2$
Distance between bogies	L_B	m
Number of wheels	N_w	
Cake weight	m_w	kG
Primary suspension stiffness	k_{pi}	N/m
Primary suspension damping	c_{pi}	Ns/m
Secondary suspension stiffness	k_{si}	N/m
Secondary suspension damping	c_{si}	Ns/m

2.2.2. Model of the rail structure system on the bridge

The track and bridge system is modeled using a 2D finite element approach. In this model, the bridge is represented as a simply supported beam, while the track is modeled as a continuous elastic beam resting on a multi-layer support system. Discrete masses are used to represent the sleepers, which are connected to the rail beam through spring-damper elements that simulate the rail pads. Both the bridge and the rail are modeled as Euler-Bernoulli beams to capture their flexural behavior accurately. For ballastless track configurations, the system typically consists of the rail, rail pad, sleeper, and sleeper pad arranged directly above the bridge structure. The detailed ballastless track model is illustrated in Figure 5. The technical parameters of the track components are listed in Table 2. In addition, the input parameters of the bridge are also given in Table 3.

Table 2. *Rail system parameter*

Parameter		Symbol	Unit
Rail	Modulus of elasticity of material	E_R	N/m^2
	Moment of inertia	I_R	m^4
	Mass per unit length	μ_R	kg/m
	Damping coefficient	c_R	%
Rail pads	Vertical stiffness of the rail pad	k_P	N/m
	Vertical damping coefficient of rail pad	c_P	Ns/m
Sleeper	Distance between sleepers	L_S	m
	Mass	m_S	Kg

Table 3. *Bridge model parameter*

Parameter	Symbol	Unit
Span length	L	m
Modulus of elasticity	E	N/m^2
Moment of inertia	I	m^4
Damping coefficient	η	%

In this model, the sleeper structure is placed directly on the concrete foundation instead of a traditional ballast bed. The connection between the sleeper and foundation is mediated by an elastic sleeper pad, as detailed in Figure 5. The technical parameters of this layer are presented in Table 4.

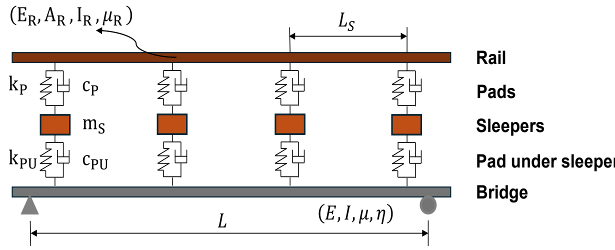


Figure 5. *Ballastless track system structure model*

Table 4. *Table of parameters of sleeper pads*

Parameter		Symbol	Unit
Sleeper pad	Vertical stiffness of the sleeper pad	k_{PU}	N/m
	Vertical damping coefficient of the sleeper pad	c_{PU}	Ns/m

2.2.3. Solution method

The method is used in this study to solve the coupled equations of motion of the train-rail-bridge interaction simulation for each time step. Since the nonlinear contact between the vehicle and the rail depends on the position and time, the Newmark Beta method is used to increase the efficiency of the solution process.

3. Model sampling and building a dataset

In this study, Latin Hypercube Sampling (LHS) [12] is employed to generate statistically representative samples of the uncertain model parameters, specifically including train speed, motor car weight, and trailer car weight. The selected parameters, along with their corresponding probability distributions, are summarized in Table 5.

Table 5. *Selected parameters and their probability distributions*

Parameter	Distribution	Min, Max	Mean	Std
Speed (km/h)	Uniform	160, 279	219.5	34.37
Weight of motor car (kg)	Uniform	42400, 66000	54200	6816.04
Weight of trailer car (kg)	Uniform	44400, 66000	55200	6353.94

Using the LHS method, a total of 1000 samples of the uncertain parameters are generated. Subsequently, the numerical model described in Section 2 is executed for each of these samples, resulting in 1000 simulation runs. Each simulation captures variations in train speed, total motor car weight, and total trailer car weight, along with the corresponding dynamic responses of the bridge.

The output dataset thus constructed comprises 1000 data samples, each containing the uncertain input parameters and the dynamic responses of the system. Specifically, the dynamic responses include the maximum vertical displacements of the bridge girder at $L/4$, $L/2$, and $3L/4$ positions, where L denotes the span length of the bridge under consideration.

4. Building a machine learning model to accurately predict train loads when passing over bridges

4.1. Algorithm XGBoost (Extreme Gradient Boosting)

The XGBoost algorithm was proposed by Chen and Guestrin [13] based on the structure of the Gradient Boosted

Decision Trees (GBDT) ensemble learning algorithm, and is used in supervised learning situations where the training data with multiple input attributes to predict an output variable can be a continuous value. This algorithm has attracted much attention due to its outstanding results in Kaggle machine learning competitions. Unlike GBDT, the objective function of XGBoost includes a regularization term in the loss function to avoid overfitting. The objective function of the XGBoost algorithm is in the form:

$$obj(\theta) = \sum_{i=1}^{n} L(\hat{y_i}, y_i) + \sum_{i=1}^{n} \Omega(f_k) + C \quad (5)$$

where L is the loss function measuring the difference between the predicted value $\hat{y_i}$ and the target value y_i, $\Omega(f_k)$ is the normalized value at the kth iteration, and C is a constant that can be selectively discarded.

Normalized quantity function $\Omega(f_k)$:

$$\Omega(f_k) = \gamma T + \frac{1}{2}\lambda \sum_{j=1}^{T} \omega_j^2 \quad (6)$$

where $\gamma$is the leaf complexity, $T$is the number of leaves, $\lambda$is the penalty variable, and ω_j is the predicted output at each leaf node.

Furthermore, XGBoost uses a Taylor series second-order approximation of the loss function instead of the first derivative. If the loss function is the mean square error (MSE), then the objective function can be written as:

$$obj(\theta) = \sum_{i=1}^{n} \left[g_i f_t(x_i) + \frac{1}{2} h_i f_t^2(x_i) \right] + \Omega(f_t) \quad (7)$$

where g_i and h_i are the first-order and second-order gradient statistics on the loss function.

The final loss value is calculated by summing all the loss values of the leaf nodes at step t. Therefore, the objective function has the form of the equation:

$$obj(\theta) = \sum_{j=1}^{T} \left[\left(\sum_{i \epsilon I_j} g_i \right) \omega_j + \frac{1}{2} \left(\sum_{i \epsilon I_j} h_i + \lambda \right) \omega_j^2 \right] + \gamma T \quad (8)$$

where I_j is the total number of samples in the leaf node j.

XGBoost – a machine learning algorithm with high accuracy and fast training speed, with many hyperparameters. The optimizable hyperparameters of the proposed method are as follows:

Number of estimators: The maximum number of sub-models built in an ensemble learning model. In general, if the value is too small, the model mismatch problem will occur, and if the value is too large, the amount of computation will increase greatly. Moreover, after reaching a certain value, the benefit of improving the model is negligible.

Maximum depth: The maximum depth of a tree limits the number of nodes from root to leaf. Increasing this value will make the model more complex and more likely to cause overfitting.

Minimum child weight: Defines the minimum total weight required in a leaf node. If the total weight of a node is less than this value, the split stops and the node becomes a leaf node.

Colsample bylevel: The proportion of columns randomly sampled at each level of the decision tree. Subsampling occurs once for each new level of depth reached in the tree. Columns are subsampled from the set of columns selected for the current tree.

Subsample: Subsampling will occur once after each boosting iteration, helping to control the input data sampling process.

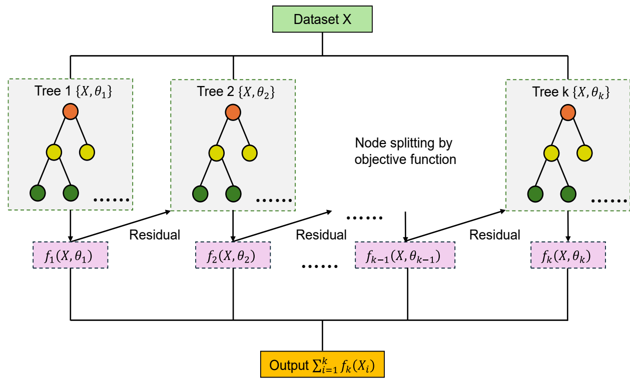


Figure 6. *XGBoost model structure*

In summary, the XGBoost model optimizes the objective function by adding a regularization term to minimize overfitting in the process of estimating predicted values. The structure of the XGBoost algorithm is shown in Figure 6.

4.2. Deploy training model

In this section, the performance of the developed XGBoost model is presented. To develop a train load estimation model when crossing a bridge, the dataset of 1000 samples is divided into small datasets, including a training set and a testing set, with proportions of 80% and 20% corresponding to the processing stages of the XGBoost model to avoid overfitting during the model training process. The input parameters are also selected during the model evaluation process, including the vertical displacement of the bridge girder at the positions $L/4, L/2,$ and $3L/4$ (where L is the calculated span length) and the train speed. In this study, the coefficient of determination (R^2), root mean square error (RMSE), and mean absolute error (MAE) are used to evaluate the performance of the model. To control the learning process, Bayesian hyperparameter optimization based on the probabilistic model is selected to be used in this study [14]. The influence of the input variables on the train weight is also examined. In the XGBoost model, gradient boosting constructs boosting trees to determine the scores of input variables. The scores of each input variable give the importance of each input variable in the training model, and their importance is evaluated based on the coefficients: "gain", "frequency" and "cover". In this study, the Gain coefficient is used to determine the scores [15] of the input variables.

$$gain = \frac{1}{2}\left[\frac{\left(\sum_{i\epsilon I_L} g_i\right)^2}{\left(\sum_{i\epsilon I_L} h_i + \lambda\right)} + \frac{\left(\sum_{i\epsilon I_R} g_i\right)^2}{\left(\sum_{i\epsilon I_R} h_i + \lambda\right)} - \frac{\left(\sum_{i\epsilon I} g_i\right)^2}{\left(\sum_{i\epsilon I} h_i + \lambda\right)} \right] \quad (9)$$

where, $I_L$and I_R are the samples of the left node and right node after branching, $I = I_L + I_R$. The higher the score Gain, the higher the importance score of the variable, indicating that the corresponding parameter variable is more important and more effective.

4.3. Results and model evaluation

To minimize the MSE value, Bayesian optimization is performed with a maximum of 100 iterations, and each iteration includes a maximum of 20 function evaluations. The hyperparameters that generate the smallest MSE on the validation dataset are then used to evaluate the model's performance on the test dataset. Figure 7 illustrates the convergence curve of the Bayesian hyperparameter optimization process applied to the XGBoost model across the specified parameter ranges. Notably, the model achieved a preliminary MSE of approximately 0.629 (T^2) after 10 iterations, which further decreased to 0.555 (T^2) at the 75th iteration, representing the minimum MSE achieved during the optimization process. These results confirm that the optimization procedure significantly enhanced the model's predictive accuracy. The optimal hyperparameter values determined through Bayesian optimization are presented in Table 6.

Table 6. *Results of optimal hyperparameter values using the Bayesian optimization method*

Hyperparameter	Upper limit	Lower limit	Optimal value
N_estimators	100	500	126
learning_rate	0	1	0.16
max_depth	3	10	3
min_child_weight	2	10	3.69
subsample	0.5	1	0.55
Reg_gamma	0	1	0.68

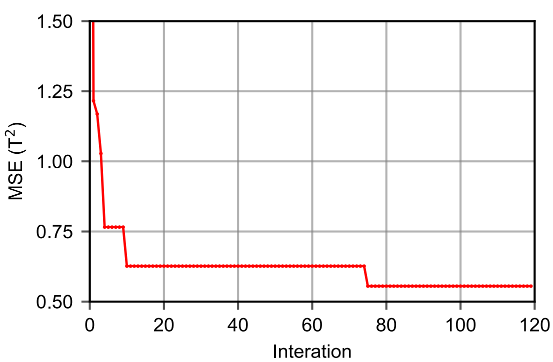
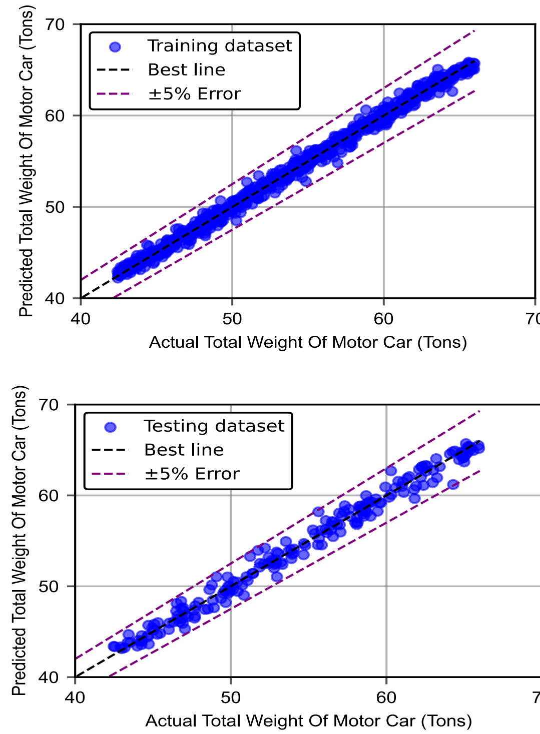


Figure 7. *The minimum MSE convergence curve of the hyperparameter optimization process*

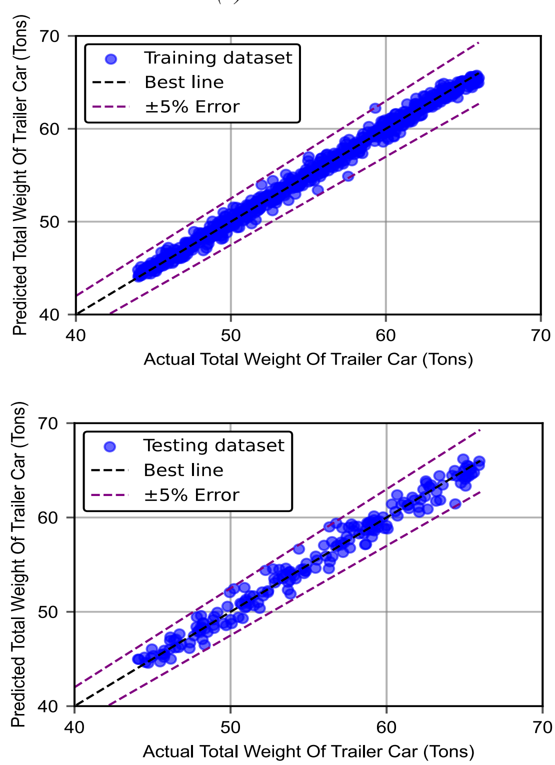
Figure 8 presents the regression plots comparing the predicted and actual values for both the training and testing datasets. Additionally, Table 7 provides the training results of the model.

Table 7. *Model performance results*

Parameter	Total weight of the motor car		Total load of trailer	
	Training set	Test set	Training set	Test set
R^2	0.996	0.983	0.993	0.979
RMSE	506.6	877.9	543.9	897.1
MAE	385.9	672.1	405.6	686.3



(a) Motor car



(b) Trailer car

Figure 8. *Estimated results of total train weight based on the XGBoost model*

During the training process, the analysis results of the trained model show that the model has high prediction performance with $R^2 = 0.996$, $RMSE = 506.6$, $MAE = 385.9$ for the training set and $R^2 = 0.983$, $RMSE = 887.9$, $MAE = 672.1$ for the test set of the prediction of the total load of the motor car. For the total load of the trailer car, the optimized model still maintains high accuracy with the results $R^2 = 0.993$, RMSE = 543.9, MAE = 405.6 of the training set and $R^2 = 0.979$, RMSE = 897.1, MAE = 686.3 for the test set. Overall, although the model has shown high performance before optimization, the optimized model shows better performance than the original model. The results show that most of the calculated car weights are within the range of $\pm 5\%$ real values.

The importance of a parameter depends on whether the predictive performance changes significantly when the feature is replaced by random noise. With the analysis process, Figure 9 shows the F-score of the four input variables based on the Gain score of the XGBoost model. As shown in the figure, the train speed and the vertical displacement values of the bridge girder at positions L/2, L/4, and 3L/4 are important factors affecting the train car load prediction. In addition, it is observed that the vertical displacement value of the bridge girder is the most important parameter.

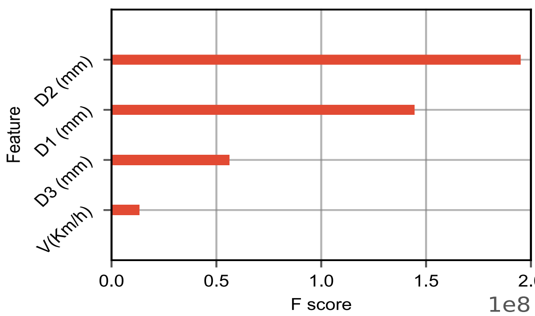


Figure 9. *Feature importance analysis using F-score (Gain metric) from the XGBoost model*

5. Conclusions

This study developed an integrated machine learning-based framework for estimating in-motion train weights on railway bridges, combining a finite element-based train–rail–bridge interaction model with the XGBoost machine learning algorithm. The Gouhe Bridge, under the operation of Pioneer EMU trains, was used as a case study to validate the proposed approach. Based on the results, the following conclusions can be drawn:

A comprehensive dataset of 1000 samples was generated using Latin Hypercube Sampling to reflect uncertainties in key parameters, including train speed, motor car weight, and trailer car weight, along with corresponding bridge girder displacements at L/4, L/2, and 3L/4 positions.

The XGBoost regression model was trained using this dataset to predict the total load of trains during bridge crossings. After Bayesian hyperparameter optimization, the model demonstrated high predictive accuracy, achieving:

For motor cars: R^2 = 0.996, RMSE = 506.6 kg, MAE = 385.9 kg (training set) and R^2 = 0.983, RMSE = 877.9 kg, MAE = 672.1 kg (test set).

For trailer cars: R^2 = 0.993, RMSE = 543.9 kg, MAE = 405.6 kg (training set) and R^2 = 0.979, RMSE = 897.1 kg, MAE = 686.3 kg (test set).

Feature importance analysis indicated that the vertical displacements of the bridge girder, particularly at L/2, L/4, and 3L/4, are the most influential factors in predicting train loads, while train speed also plays a significant role.

The Bayesian optimization process successfully reduced the MSE to 0.555 (T^2) after 75 iterations, identifying optimal hyperparameters (e.g., n_estimators = 126, learning_rate = 0.16, max_depth = 3). This confirms the effectiveness of Bayesian optimization in enhancing model performance while ensuring computational efficiency.

Overall, the proposed framework demonstrates that machine learning models can be effectively integrated with physics-based dynamic interaction simulations for accurate, real-time estimation of in-motion train loads on high-speed railway bridges. The model's ability to maintain predictions within ±5% of actual values highlights its potential application in bridge management, maintenance planning, and operational safety monitoring in modern railway systems.

REFERENCES

[1] R. Pimentel, D. Ribeiro, L. Matos, A. Mosleh, and R. Calçada, "Bridge Weigh-in-Motion system for the identification of train loads using fiber-optic technology", *Structures*, vol. 30: pp. 1056-1070, 2021.

[2] B. Pintão, A. Mosleh, C. Vale, P. Montenegro, and P. Costa, "Development and validation of a weigh-in-motion methodology for railway tracks", *Sensors,* vol. 22, no. 5, pp. 1976, 2022.

[3] F. Marques, C. Moutinho, W.-H. Hu, Á. Cunha, and E. Caetano, "Weigh-in-motion implementation in an old metallic railway bridge", *Engineering Structures,* vol. 123, pp. 15-29, 2016.

[4] A. Žnidarič, J. Kalin, M. Kreslin, P. Favai, and P. Kolakowski, "Railway bridge Weigh-in-Motion system", *Transportation Research Procedia,* vol. 14, pp. 4010-4019, 2016.

[5] H. Xia, N. Zhang, and W. Guo, *Dynamic interaction of train-bridge systems in high-speed railways*. Springer: Berlin/Heidelberg, Germany, 2018.

[6] D. Cantero, T. Arvidsson, E. OBrien, and R. Karoumi, "Train–track–bridge modelling and review of parameters", *Structure and Infrastructure Engineering,* vol. 12, no. 9, pp. 1051-1064, 2016.

[7] T. L. A. Duc, P. H. Nam, N. Van My, T. D. Quang, and F. Paolacci, "2D train-track-bridge interaction analysis considering the effect of ballasted tracks", *The University of Danang - Journal of Science and Technology*, vol 22, no. 11B, pp. 1-7, 2024.

[8] P. Lou, "Finite element analysis for train–track–bridge interaction system", *Archive of Applied Mechanics,* vol. 77, no. 10, pp. 707-728, 2007.

[9] D.V. Nguyen, K.D. Kim, and P. Warnitchai, "Simulation procedure for vehicle–substructure dynamic interactions and wheel movements using linearized wheel–rail interfaces", *Finite Elements in Analysis and Design,* vol. 45, no. 5, pp. 341-356, 2009.

[10] K. Du Kim and P. Warnitchai, "Dynamic analysis of three-dimensional bridge–high-speed train interactions using a wheel–rail contact model", *Engineering Structures,* vol. 31, no. 12, pp. 3090-3106, 2009.

[11] R. Ferrara, "*A numerical model to predict train induced vibrations and dynamic overloads*", Doctoral dissertation, Université Montpellier II, Mende, France, 2013.

[12] M. D. McKay, R. J. Beckman, and W. J. Conover, "A comparison of three methods for selecting values of input variables in the analysis of output from a computer code", *Technometrics,* vol. 42, no. 1, pp. 55-61, 2000.

[13] T. Chen and C. Guestrin, "Xgboost: A scalable tree boosting system", in *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, San Francisco, California, USA, 2016, pp. 785-794.

[14] M. H. Nguyen and H. N. Phan, "A Bayesian-optimized neural network model for shear capacity of a perfobond strip connector in various types of composite structures", *Advanced Steel Construction,* vol. 20, no. 4, pp. 376–384, 2024.

[15] L. Breiman, J. Friedman, R. A. Olshen, and C. J. Stone, *Classification and regression trees*. Routledge: Oxfordshire, England, 2017.