TECHNICAL PAPER

JOURNAL OF THE SOUTH AFRICAN INSTITUTION OF CIVIL ENGINEERING
ISSN 1021-2019 (print) | ISSN 2309-8775 (online)
Vol 67 No 2, June 2025, Pages 26–37, Paper 1759

Prediction of the fundamental period of infilled reinforced concrete frame structures using advanced machine learning methods

A Yahiaoui, G Markou, N Bakas, S Dorbani

DR ASMA YAHIAOUI is a civil engineer and researcher. She was ranked first out of 200 in the structural engineering Master's programme, and recently earned her PhD from the University of Science and Technology Houari Boumediene (USTHB) with distinction. She gained experience in teaching several courses as a graduate research assistant at USTHB, published a paper in a Q1 journal, participated in three international conferences, and collaborated with several professors worldwide to establish the upcoming original work associated with this paper. Her research focuses primarily on applying computational methods in civil and earthquake engineering.

Contact details:
Department of Structures and Materials
Built Environment Research Laboratory, Civil Engineering Faculty
University of Science and Technology Houari Boumediene
Algiers, Algeria
E: ayahiaoui1@usthb.dz

DR GEORGE MARKOU is a civil engineer and academic with expertise in computational mechanics. He obtained his PhD from the National Technical University of Athens, receiving multiple awards and scholarships throughout his academic journey. As a professional civil engineer with extensive industry experience he has designed more than 50 projects. He also held academic positions in the UAE, Chile, the UK and South Africa. In 2024, he joined the Cyprus University of Technology where he is an assistant professor, while he still serves as research associate at the University of Pretoria in South Africa. His research spans FEM software development, machine learning, seismic assessment and soil-structure interaction.

Contact details:
Department of Civil Engineering
University of Pretoria
Hatfield Campus
Pretoria 0028, South Africa

Cyprus University of Technology
Department of Civil Engineering and Geomatics, Limassol 3036, Cyprus
E: george.markou@cut.ac.cy

DR NICOLAOS BAKAS has an extensive background in artificial intelligence, with more than 90 publications in areas including machine learning, numerical methods, optimisation, and large language models. He has served as principal investigator, researcher and coordinator for at least 20 projects in various research centres. Besides a PhD from the National Technical University of Athens, and more than ten years of university teaching experience, he also has in excess of 20 years of programming expertise in diverse languages and frameworks, resulting in his training seminars having reached more than 5 000 engineers so far.

Contact details:
Machine Intelligence Research and Engineering
School of Liberal Arts and Sciences
Department of Information Technology, Cybersecurity and Computer Science
The American College of Greece, Deree, Athens, Greece
E: nb@machine-intelligence.ai

PROF SAIDA DORBANI obtained her PhD from the University of Science and Technology Houari Boumediene (USTHB), Algeria. She served as head of USTHB's Department of Structures and Materials for many years, and is currently a member of the Scientific Committee of the department, the Scientific Council of the faculty, and the Scientific Council of the university. She is the research team leader for earthquake engineering, engineering seismology, structural dynamics, building acoustics, and natural risk management. Her research also includes the valorisation of local materials in the building sector, linear and nonlinear building behaviour (deterministic and probabilistic) under seismic load, and the use of AI techniques to solve civil engineering problems.

Contact details:
Department of Structures and Materials
Built Environment Research Laboratory
Civil Engineering Faculty
University of Science and Technology Houari Boumediene
Algiers, Algeria
E: s.dorbani.gc@gmail.com

The use of machine learning (ML) to solve civil engineering problems has increased remarkably during the last few decades due to its effectiveness in reliably approximating complex relationships. In this paper, a key parameter of seismic design is estimated using hyperparameter ML algorithms to develop predictive models that compute the fundamental period. Initially, the impact of the train-test split ratio was investigated using three different splits, where the best results were achieved with train-test split ratios equal to 90/10 for all metrics. By predicting the fundamental period with three ML methods, namely XGBoost-HYT-CV, DANN-MPIH-HYT, and RF-HYT, the best fit was acquired by XGBoost-HYT-CV (coefficient of determination R^2 = 99.994% and mean absolute error MAE = 0.00428). Although international literature agrees that building height is the primary factor influencing the fundamental period, feature engineering has revealed that the natural logarithm of the percentage of openings is the most significant parameter. This finding underscores the value of feature engineering in generating additional variables and uncovering their impact on output variables. Finally, an equation was derived from POLYREG-HYT that outperformed all existing formulae, deriving a final MAE of 0.0153, approximately three times smaller than the best-performing equations proposed in the international literature.

Keywords: fundamental period, tuned hyperparameters, feature importance, machine learning, reinforced concrete structures, numerical modelling

INTRODUCTION

The fundamental period (FP) is a crucial parameter in seismic design and the dynamics of structures since it allows the safe design of new buildings when computing the design seismic loads (Gallipoli *et al* 2023). This dynamic feature depends on the distribution of the mass and stiffness of the structures (Asteris *et al* 2015). In most cases, the mass of the nonstructural elements, such as masonry infills, is taken into consideration, whereas the lateral stiffness of these elements is neglected (Chennit *et al* 2022), leading to unsafe design, especially since infilled reinforced concrete (RC) structures are an integral part of buildings (Khelfi *et al* 2021).

Extensive research has been conducted on the effect of infill walls on the FP of buildings. Asteris *et al* (2015) showed that the FP decreases by 40% when the stiffness of infill panels increases from 2.25×10^5 to 25×10^5 kN/m. According to Amanat and Hoque (2006), the number of infill walls in a moment frame and the span length of the panel have a significant impact on the natural frequency of the building. Kose (2009) demonstrated that the FP is inversely proportional to the ratio of infilled panels to the total number of panels, and overlooking this percentage has a considerable influence on the value of the natural period. Furthermore, Ngab *et al* (2021) investigated the effect of various parameters affecting the period of vibration of RC buildings and concluded that the value of this dynamic characteristic is reduced by more than 50% in both solid and flat slab construction when infill walls are taken into account when computing FP. All these findings

Yahiaoui A, Markou G, Bakas N, Dorbani S. Prediction of the fundamental period of infilled reinforced concrete frame structures using advanced machine learning methods. J. S. Afr. Inst. Civ. Eng. 2025:67(2), Art. #1759, 12 pages. http://dx.doi.org/10.17159/2309-8775/2025/v67n2a3

underline the necessity of accounting for the presence of infill walls in the computation of the FP, particularly as the latter is proportional to the lateral displacement and inter-storey drift (Dorbani *et al* 2015).

In structural computations, many databases were created to solve different complex problems, and recently the most common method to develop solutions for specific problems has been the use of artificial intelligence and machine learning (ML) algorithms. AlHamaydeh *et al* (2022) investigated the behaviour of fibre-reinforced polymer RC deep beams without stirrups, where 93 finite element models were produced to train the higher-order stepwise regression. Markou and Bakas (2021) and Spijkerman *et al* (2021) estimated the shear capacity of RC slender beams without stirrups where two large databases were generated from nonlinear analyses through Reconan FEA software (Reconan FEA v2.00, User's Manual 2020), consisting of 35 849 and 10 000 data points, respectively. Chibaya *et al* (2023) used Reconan FEA (Reconan FEA v2.00, User's Manual 2020) to develop 300 models to extract a formula from the ML algorithms that compute the section's rotation of horizontally curved steel I-beams. Ababu *et al* (2022) created a database of 1 890 points to develop an equation for the deflection of horizontally curved steel I-beams, while Bakas *et al* (2021) investigated the database developed by Markou and Bakas (2021) to predict the shear strength of slender beams using different ML methods through high-performance computers.

Gravett *et al* (2021) developed a database containing 475 cases to estimate the FP of bare RC buildings, considering the soil structure interaction (SSI) phenomenon. The models were trained using higher-order nonlinear regression, and three equations were proposed. This dataset was expanded by Taljaard *et al* (2021) with the addition of 30 cases, and by Carstens *et al* (2022) where 315 unique points were included to get a database of 790 cases and three improved new expressions were proposed. Regarding steel structures with consideration of SSI, Van der Westhuizen *et al* (2022) proposed a dataset with 1 152 data points. Calitz *et al* (2023) utilised Reconan FEA (Reconan FEA v2.00, User's Manual 2020) to create various models, and by using the automated procedure, 10 368 FP results were produced. Van der Westhuizen *et al* (2023) constructed the largest dataset in the international literature, consisting of 98 308 models of steel moment frame buildings including the SSI effect. All the databases created by various researchers exhibit an impressive ability to solve different complex problems, demonstrating the capability of ML approaches to give reliable results.

One of the most well-known datasets for estimating the fundamental period (FP) of infilled reinforced concrete (RC) frame structures is the FP4026 Research Database (Asteris 2016), widely utilised by researchers applying various machine learning (ML) techniques. Asteris and Nikoo (2019) employed the artificial bee colony optimisation algorithm to train an artificial neural network (ANN), while Somala *et al* (2021) explored bagging and boosting approaches, two widely used ensemble learning techniques. The key difference between these methods lies in their data distribution – bagging introduces random variations in the training set, whereas boosting adaptively modifies the training data based on the performance of previously created classifiers (Kotsiantis & Pintelas 2004).

Mirrashid and Naderpour (2022) applied both ANN and a neuro-fuzzy approach, the latter combining fuzzy systems and ANN to leverage the strengths of both models (Yousefi & Hamilton-Wright 2014). Similarly, Latif *et al* (2022) used ANN alongside extreme gradient boosting (XGBoost), a boosting technique that iteratively improves weak learners to form a strong predictive model (Freund & Schapire 1997).

Charalampakis *et al* (2020) developed a formula for FP as a function of the number of storeys, span length, percentage of openings, and masonry wall stiffness. Their approach combined ANN, the M5 model tree algorithm (a hybrid of decision trees and linear regression) (Quinlan1992), and nonlinear regression. Verma and Sahoo (2022) applied a powerful ML method capable of capturing higher-order nonlinear relationships, known as the group method of data handling (GMDH), which is widely used in complex systems (Naderpour *et al* 2020).

More recently Yahiaoui *et al* (2023) estimated FP using three types of boosting techniques:

- Gradient Boosting Decision Trees (GBDT) – a fundamental boosting method that successively trains weak learners to correct previous errors (Friedman 2001).
- CatBoost – which differs from other boosting techniques by incorporating ordered boosting to prevent overfitting (Prokhorenkova *et al* 2018).
- LightGBM – an open-source ML library developed by Microsoft, which employs a leaf-wise node-splitting method instead of level-wise splitting, leading to faster convergence and improved model performance (Ke *et al* 2017).

Additionally, Yahiaoui *et al* (2023) developed an FP equation using multivariate adaptive regression splines (MARS), a white-box ML method that automatically detects nonlinear relationships and feature interactions (Friedman 1991). While these studies produced promising results, the majority of them were conducted without hyperparameter tuning, which could limit optimisation. The most accurate equation for FP estimation achieved a root mean square error (RMSE) of approximately 0.065, which is acceptable but leaves room for improvement. As demonstrated in this research, further enhancements in model accuracy can still be achieved.

This article aims to predict the FP of infilled RC structures by the application of three open-source ML approaches from the work of Markou *et al* (2024). These ML algorithms are the following:

- XGBoost-HYT-CV
- DANN-MPIH-HYT
- RF-HYT

By employing the feature engineering method, fifteen input features were generated instead of five (which were the initial input data). Moreover, using three different split ratios, the impact of the train-test split ratio on the performance of the predicted models was investigated. Feature importance was applied to get insight into the models to determine which feature has the primary effect on the FP through the use of the Asteris (2016) dataset. Finally, an equation was derived from the POLYREG-HYT predictive formula that was found to be the most accurate in the international literature.

RESEARCH SIGNIFICANCE

Several researchers have used ML techniques to predict the FP of infilled RC structures (Asteris & Nikoo 2019; Charalampakis *et al* 2020; Latif *et al* 2022; Mirrashid & Naderpour 2022; Somala *et al* 2021; Yahiaoui *et al* 2023), particularly using the dataset presented by Asteris (2016), where the majority of the models have shown good results. This paper aims to propose new models that have a significantly improved accuracy compared to the existing models found currently in the international literature. The comparison between the newly proposed predictive models and the existing models will be made by comparing the different error metrics reported in the literature. Hence, three ML algorithms with hyperparameter tuning proposed by Markou *et al* (2024) were employed to estimate the FP of the infilled RC buildings that form the data set in Asteris (2016). Additionally, the impact of the train-test split ratio on the performance of the predicted model was investigated using three different split ratios: 90/10%, 80/20% and 70/30%, in an attempt to derive the optimum train/test ratio for the understudy dataset. Furthermore, feature engineering is used to generate fifteen input variables, and the feature importance method is employed to determine the effect on the accuracy of derived predictive models, where the feature that had the most impact on the period of vibration was determined through this numerical investigation. Furthermore, the POLYREG-HYT (Markou *et al* 2024) white-box ML method is used to derive a formula for the natural period as a function of the fifteen inputs which is then compared to existing formulae. Its accuracy and robustness are found to be superior according to the numerical findings presented in this manuscript.

It is also important to note here that the dataset proposed by Asteris (2016) was used (*The FP4026 Research Database …*) on the fundamental period of infilled RC frame structures. This database contains 4 026 cases of infilled frame buildings modelled according to Eurocode 2 (Code 2005a) and Eurocode 8 (Code 2005b), and considers the variability of five features, namely:

- the number of storeys (N) varying from 1 to 22, with a constant storey height of 3 m
- the number of spans (NS) taking three values into account: 2, 4, and 6
- the span length (LS) from 3 m to 7.5 m with a step of 1.5
- the percentage of openings (rho) with five values from 0% to 100% with a range of 25%
- seven values for masonry wall stiffness (St) that were also introduced in the under-study dataset.

Therefore, the computational efficiency in the development of the predictive models through the use of this relatively large dataset will be presented.

FEATURE ENGINEERING

The objective of feature engineering is to create new features by modifying the existing ones using various transformations or by creating other significant features, as discussed in Verdonck *et al* (2021) and Van der Westhuizen *et al* (2022). Thus, to enhance the developed solution for the needs of this research work, the dataset is modified in such a way as to have fifteen features – the five initial inputs found in Asteris (2016) and an addition of ten modified parameters. To achieve this, each initial parameter (feature) is modified in the following ways: In(feature + 1), which will be referred to as **lnFeature** and $\frac{1}{1 + \text{feature}}$, which will be referred to as **1Feature**. Table 1 illustrates the first five rows of the produced data with fifteen inputs that resulted after implementing feature engineering. It is important to note here that the proposed ML algorithms (Markou *et al* 2024) are open-source and can be found through the link provided by Bakas *et al* (2023).

DEPLOYED MACHINE LEARNING ALGORITHMS

The benefit of using the three advanced ML algorithms proposed by Markou *et al* (2024) is the accuracy, advanced extendibility properties, and efficient use of computation time in developing predictive models. In order to further illustrate the computational efficiency of the adopted advanced ML algorithms, the Optuna framework (Akiba *et al* 2019) was also utilised. According to Yahiaoui *et al* (2023), the use of lightGBM took place for the hyperparameter tuning, where the algorithm required twelve hours to tune the hyperparameters. This time duration is the maximum time allowed to close the session on platforms such as Kaggle (Kaggle n.d.), which is found to be very time-consuming. On the other hand, when using the XGBoost-HYT-CV ML algorithm on the same dataset, the required time was just a few minutes, and for the cases of RF-HYT and DANN-MPIH-HYT a couple of hours. This numerical finding shows the computational efficiency of the proposed ML algorithms (Markou *et al* 2024), which will be demonstrated in the results section of this manuscript. For further details on the meaning of different ML-related terminology, Appendix B is provided (see page 36).

The following section briefly describes the three advanced ML algorithms proposed in Markou *et al* (2024), and which were used for this research work:

- **Extreme gradient boosting with hyperparameter tuning and cross-validation (XGBoost-HYT-CV)**
  XGBoost is a gradient boosting algorithm widely used among data scientists due to its scalability and speed. This algorithm also avoids a common problem in ML, namely overfitting, which occurs in the case of a model performing well in the training set but poorly in the case of unseen data (Ying 2019). For this reason, this algorithm includes a regularised model (Chen & Guestrin 2016). It is based on decision trees employed to solve regression and classification problems, and is available as an open-source library (XGBoost Documentation

Table 1 The first five generated data points by feature engineering

N	NS	LS (m)	rho	St × 10^5 kN/m	lnN	lnNS	lnLS	lnrho	lnSt	1N	1NS	1 LS	1rho	1St
1	2	3.0	0	2.25	0.693	1.099	1.386	0	1.179	0.5	0.333	0.25	1	0.308
1	2	3.0	1	2.25	0.693	1.099	1.386	0.693	1.179	0.5	0.333	0.25	0.5	0.308
1	2	3.0	1	4.50	0.693	1.099	1.386	0.693	1.705	0.5	0.333	0.25	0.5	0.182
1	2	3.0	1	7.50	0.693	1.099	1.386	0.693	2.140	0.5	0.333	0.25	0.5	0.118
1	2	3.0	1	11.25	0.693	1.099	1.386	0.693	2.506	0.5	0.333	0.25	0.5	0.082

n.d.). XGBoost-HYT-CV is an extension of XGBoost with hyperparameter tuning using grid search and cross-validation to improve accuracy and reduce required computational time during training and testing (Markou *et al* 2024).

- **Random forest with hyperparameter tuning and cross-validation (RF-HYT)** The random forest (RF) is a type of ensemble method that involves combining tree predictors so that each tree relies on the value of a random vector collected independently and with the same distribution (Breiman 2001). RF-HYT is an extension of RF with hyperparameter tuning using grid search (Markou *et al* 2024).
- **Deep neural network with hyperparameter tuning, MPI, and Horovod (DANN-MPIH-HYT)** The deep neural network (DNN) is an imitation of the human brain; it can be applied with both supervised and unsupervised learning methods. DNN is ANN with multiple hidden layers of connected units between the input and output layers. Even with the large applications of DNN in various fields, it is complex and computationally demanding (Sarvepalli 2015). The proposed method by Markou *et al* (2024) was found to perform well for both small and large datasets, but it is also the ML algorithm that derives the highest computational demand out of the three ML algorithms described in this section.

It is also interesting to note at this stage that tuning the hyperparameters in ML problems is crucial since it significantly modifies the results in a way that increases accuracy and reduces errors in the predictive models. According to Yahiaoui *et al* (2023), the most accurate results for predicting the FP of infilled RC frame buildings were achieved when the hyperparameter tuning was performed using the Optuna framework (Akiba *et al* 2019). The same numerical responses were achieved when predicting the concrete compressive strength without (Yahiaoui *et al* 2014) and with high-volume ground-granulated blast-furnace slag replacement (Rathakrishnan *et al* 2022) using random search to optimise the hyperparameters and the Optuna framework (Akiba *et al* 2019).

Therefore, according to the experience gained by Markou *et al* (2024), different assumptions were made to optimise the ML algorithmic performance during training and testing. The optimal hyperparameters for XGBoost-HYT-CV were obtained by setting the max_depth to 6 with a learning rate (eta) of 0.09 and n_estimators of 900. In addition to that, the colsample_bytree and subsample of 0.95 and 0.6 were selected, respectively. For the case of the RF-HYT algorithm, the tuned hyperparameters were set to 10 for the max_depth with the n_estimators set to be equal to 90. The min_samples split of 2 was adopted, and the max features and min_samples_leaf of 0.65 and 1 were chosen when executing the RF-HYT algorithm. Finally, the architecture of DANN-MPIH-HYT consists of two layers, where each layer consists of 100 neurons with a dropout rate equal to 0.01, a learning rate of 0.001, and a momentum of 0.75. The number of epochs and batch sizes were set to 100 and 453, respectively.

THE IMPACT OF TRAIN-TEST SPLIT RATIOS

The process of train-test split ratios consists of dividing the data into two parts – the training sets used to train the models and the test sets utilised to assess the performance of the derived predictive model from the training dataset (Naveen Venkatesh *et al* 2022). This step is crucial to ensure the optimum performance of any ML-generated model (Anifowose *et al* 2017). The general approach in ML training and testing is to divide the dataset into a training dataset that contains 80% of the total data points and the rest of the 20% of the dataset is used to test the numerically obtained model. Three different train-test split ratios were used to evaluate the effects on the accuracy and extendibility of the proposed predictive models, while the effects on the performance of the three ML algorithms were also investigated. The train-test split ratios that were studied for the needs of this work were: 90/10%, 80/20% and 70/30%.

Table 2 shows the metrics of the ML-generated models on the test sets for the three train-test split ratios. It is easy to see from Table 2 that the best performance derived for a train-test split ratio is equal to 90/10% for all error metrics. In addition to that, it can also be deduced that XGBoost-HYT-CV manages to derive near-zero error metrics, with mean absolute percentage error (MAPE), modified mean absolute percentage error (MAMPE), MAE and RMSE on the test data all being less than 0.01.

Figure 1 depicts the impact of the train-test split ratio on the performance of three ML models. The most relevant difference is noticed in the case of DANN-MPIH-HYT, where it is clear that the value of RMSE for the train-test split ratios of 90/10% is nearly 4.5 times less than for the case of 80/20% and 1.30 times less than for the case of 70/30%. This is attributed to the fact that the data points are selected randomly during the process of the train-test split ratios, leading to different performance, whereas the presence of outliers is more prominent in the case of 80/30% than in the case of 90/10%, which is also shown in Figure 2. In conclusion, choosing the optimum train-test split ratio leads to a significant effect related to the performance of the model. For the understudy dataset, the best split is 90% for training and 10% for testing, hence the results presented hereafter will be based on this split. The statistical properties of the training and testing sets for the fifteen input variables are presented in Table 3 for train-test split ratios equal to 90/10%, resulting in 3 623 cases for training and 403 for testing the ML-generated models.

Table 2 Performance of machine learning models on the test sets with different train-test split ratios

Train-test split ratios	Models	R^2 %	MAPE %	MAMPE %	MAE	RMSE
90/10%	DANN-MPIH-HYT	99.979	1.699	0.994	0.0107	0.0155
	XGBoost-HYT-CV	*99.994*	*0.694*	*0.399*	*0.0043*	*0.0081*
	RF-HYT	99.948	2.034	1.390	0.0149	0.0238
80/20%	DANN-MPIH-HYT	99.970	5.038	5.142	0.0544	0.0693
	XGBoost-HYT-CV	*99.991*	*0.925*	*0.514*	*0.0054*	*0.0103*
	RF-HYT	99.931	2.310	1.577	0.0167	0.0280
70/30%	DANN-MPIH-HYT	99.966	1.836	1.157	0.0123	0.0201
	XGBoost-HYT-CV	*99.985*	*0.994*	*0.563*	*0.0060*	*0.0132*
	RF-HYT	99.923	2.338	1.596	0.0170	0.0296

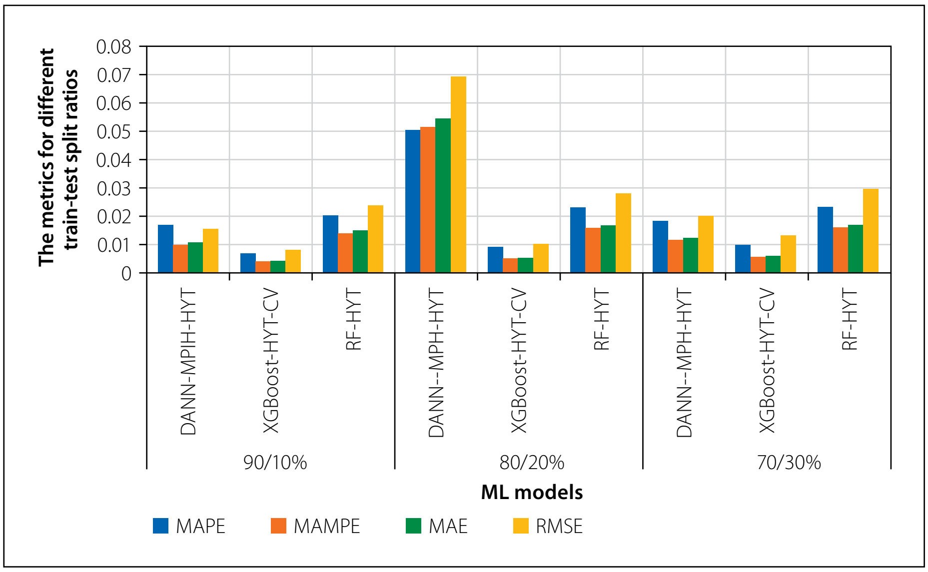

Figure 1 Impact of train-test split ratio on the performance of three machine learning algorithms

NUMERICAL RESULTS

Black-box ML methods

Black-box methods refer to ML algorithms that do not generate an equation that can be visually investigated by researchers and engineers. These methods are the DANN-MPIH-HYT, XGBoost-HYT-CV and RF-HYT (Markou *et al* 2024), which will be discussed in this section in terms of numerical response. As mentioned above, their performance will be evaluated through the use of the R^2, MAPE, MAMPE, MAE and RMSE metrics.

It is important to state at this point that several manuscripts have published results for this dataset where different ML methods were used to train and test

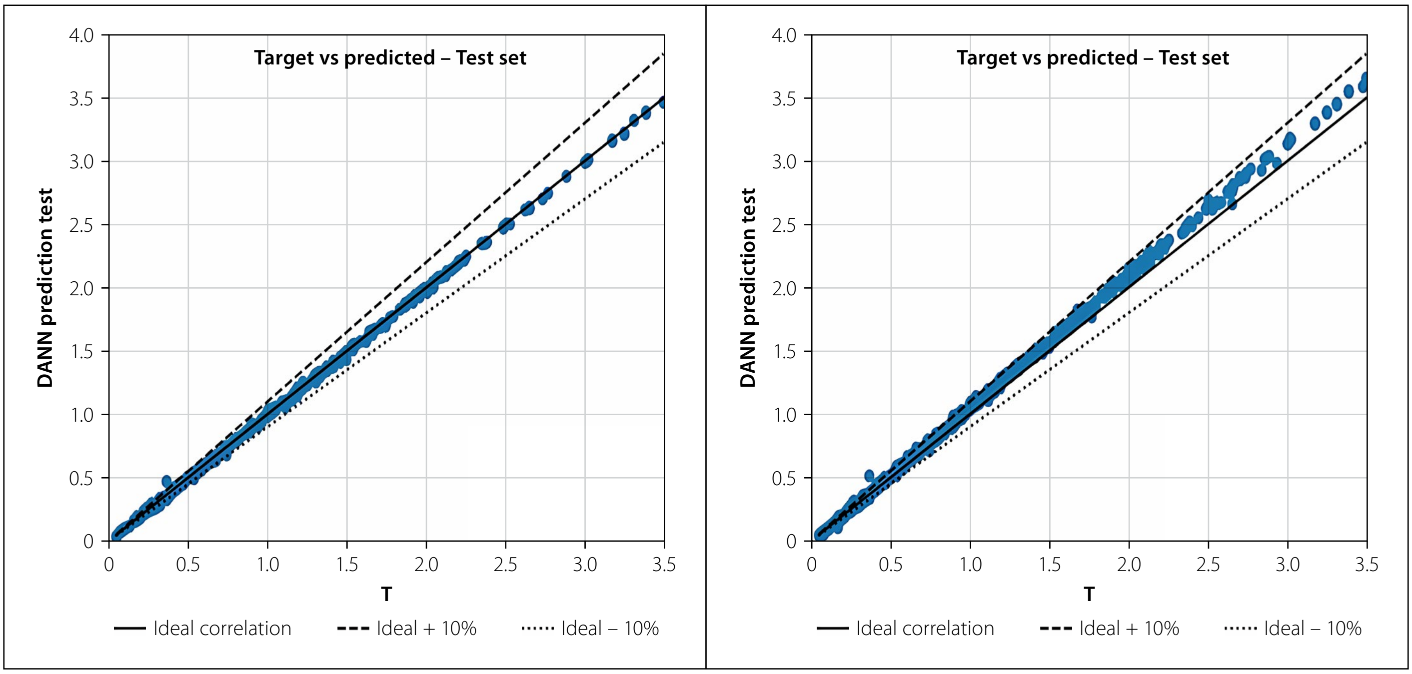

Figure 2 Comparison between the predicted and the numerically obtained periods for the case of the DANN-MPIH-HYT model for split ratios of (left) 90/10% and (right) 80/20%

Table 3 Statistical properties of the input features for the training and testing sets

Features	Training sets					Test sets				
	Mean	Median	Std	Min	Max	Mean	Median	Std	Min	Max
N	11.53	12.00	6.36	1.00	22.00	11.27	11.00	6.17	1.00	22.00
NS	4.95	6.00	1.55	2.00	6.00	4.96	6.00	1.52	2.00	6.00
LS	5.00	4.50	1.58	3.00	7.50	4.94	4.50	1.60	3.00	7.50
rho	0.63	0.75	0.40	0.00	1.00	0.64	0.75	0.40	0.00	1.00
St	11.74	11.25	7.75	2.25	25.00	12.00	11.25	8.09	2.25	25.00
LnN	2.35	2.56	0.67	0.69	3.14	2.34	2.48	0.63	0.69	3.14
LnNS	1.74	1.95	0.32	1.10	1.95	1.74	1.95	0.31	1.10	1.95
LnLS	1.76	1.70	0.27	1.39	2.14	1.74	1.70	0.27	1.39	2.14
Lnrho	0.45	0.56	0.27	0.00	0.69	0.46	0.56	0.27	0.00	0.69
LnSt	2.32	2.51	0.72	1.18	3.26	2.32	2.51	0.74	1.18	3.26
1N	0.12	0.08	0.11	0.04	0.50	0.12	0.08	0.10	0.04	0.50
1NS	0.19	0.14	0.07	0.14	0.33	0.19	0.14	0.07	0.14	0.33
1LS	0.18	0.18	0.05	0.12	0.25	0.18	0.18	0.05	0.12	0.25
1rho	0.66	0.57	0.19	0.50	1.00	0.65	0.57	0.19	0.50	1.00
1St	0.13	0.08	0.1	0.04	0.31	0.13	0.08	0.10	0.04	0.31

Table 4 Error metrics of the training and testing sets of the three ML algorithms

Models	Dataset	R^2 %	MAPE %	MAMPE %	MAE	RMSE	Time (s)
DANN-MPIH-HYT	Train	99.985	1.549	0.880	0.0098	0.0142	10887
	Test	99.979	1.699	0.994	0.0107	0.0155	0.001
XGBoost-HYT-CV	Train	*99.999*	0.273	0.155	0.0017	0.0030	2391
	Test	*99.994*	*0.694*	*0.399*	*0.0043*	*0.0081*	*0.009*
RF-HYT	Train	99.980	1.317	0.887	0.0098	0.0160	3714
	Test	99.948	2.034	1.390	0.0149	0.0238	0.015

for developing the most accurate predictive model (Charalampakis *et al* 2020; Latif *et al* 2022; Mirrashid & Naderpour 2022; Somala *et al* 2021; Verma & Sahoo 2022; Yahiaoui *et al* 2023). For researchers who used XGBoost to predict the natural period, the best published results for the case of the test dataset without tuning the hyperparameter were achieved by Somala *et al* (2021), where an R^2 equal to 99.9% and an RMSE of 0.018 were reported. For the case of RF, the same researchers reported a 0.026 as the optimum obtained RMSE and a 99.9% for the Pearson parameter R^2. In the case of the test dataset with a tuned hyperparameter, Thisovithan *et al* (2023) acquired through the use of RF an R^2 of 98.9% and a corresponding RMSE of 0.082, which is considered to be relatively high in comparison to the model of Somala *et al* (2021) derived from RF set without hyperparameter tunning.

Table 4 illustrates the performance of the three black-box ML approaches adopted for the needs of this research work when training and testing on the dataset. It is easy to depict that XGBoost-HYT-CV outperforms the two other methods in all metrics for both the training and testing sets, deriving for the test set an R^2 equal to 99.994%, and error values equal to 0.694%, 0.399%, 0.00428 and 0.00809 for MAPE, MAMPE, MAE and RMSE, respectively. The second most accurate predictive model was the one generated by DANN-MPIH-HYT, and finally RF-HYT, which had the lowest performance. Regarding the training computation time, it is evident that DANN-MPIH-HYT is computationally demanding (three hours) compared to the other two ML methods. Nevertheless, DANN-MPIH-HYT is found to be four times faster than the method proposed by Yahiaoui *et al* (2023). In addition, XGboost-HYT-CV is the fastest model, requiring a mere 40 minutes to complete the analysis and testing of the predictive model. This is 18 times faster than the time required to perform the hyperparameter tuning in Yahiaoui *et al* (2023).

It is important to note that the best results found in the literature for the Asteris (2016) database were published in Yahiaoui *et al* (2023), where it was reported that lightGBM tuned hyperparameters by the Optuna framework (Akiba *et al* 2019) derived an RMSE equal to 0.01129, a MAE of 0.00432, and 99.981% as a Pearson parameter R^2. According to the results presented here, it is easy to observe that XGBoost-HYT-CV outperforms the predictive model proposed by Yahiaoui *et al* (2023), as well as the one presented by Somala *et al* (2021). A similar numerical response was obtained through the DANN-MPIH-HYT that managed to derive minimal error metrics even in the case where deep learning performs optimally when the datasets are large in size, which was not the case with the dataset herein.

Figure 3 shows the performance of XGBoost-HYT-CV, demonstrating that this algorithm fits the dataset well with nearly no outliers. This reveals the numerical superiority of this method in reliably and robustly predicting the target variables of this database.

Feature importance

Feature importance is a method to determine which feature most affects the prediction of the target variable. It is useful especially for black-box ML-generated models since it allows researchers to investigate the sensitivity of the numerically obtained predictive models (Musolf *et al* 2022). In the case of computing the FP of infilled RC frame buildings, researchers agree that the building height, or the number of storeys, is the main parameter that has a higher effect on the prediction of this design parameter (Somala *et al* 2021; Verma & Sahoo 2022; Yahiaoui *et al* 2023).

Figure 4 illustrates the feature importance of XGBoost-HYT-CV, where it is found that the critical parameter that has

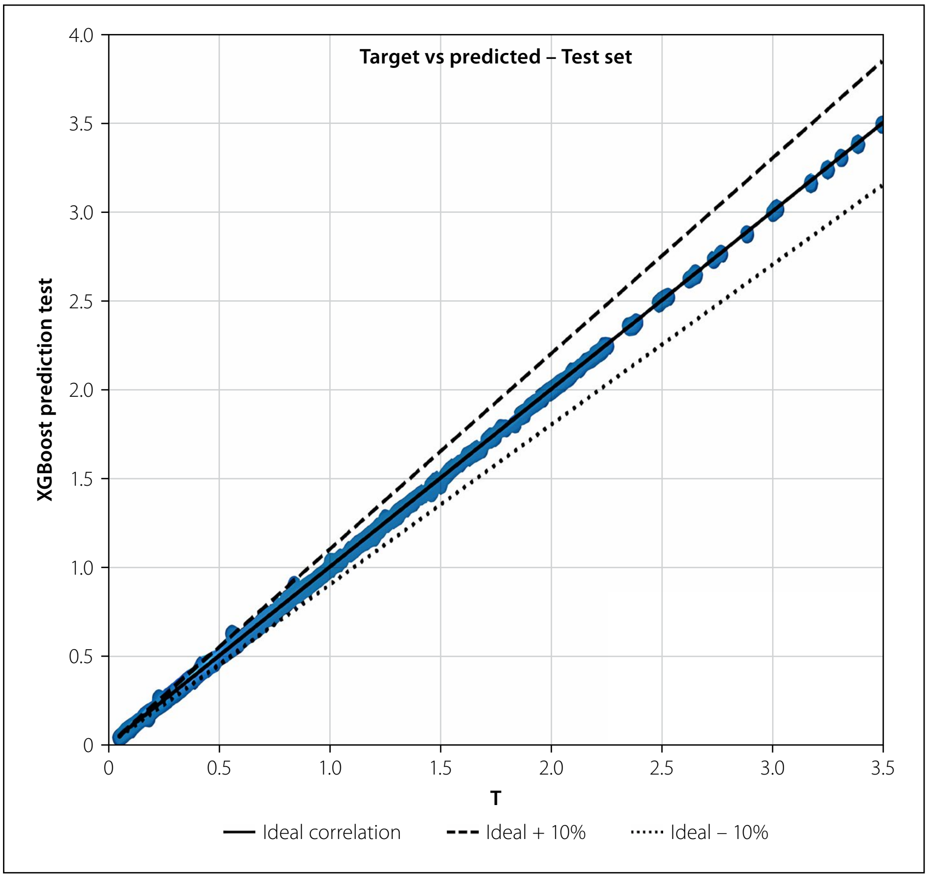

Figure 3 Comparison between the predicted period from XGBoost-HYT-CV and the numerically obtained period

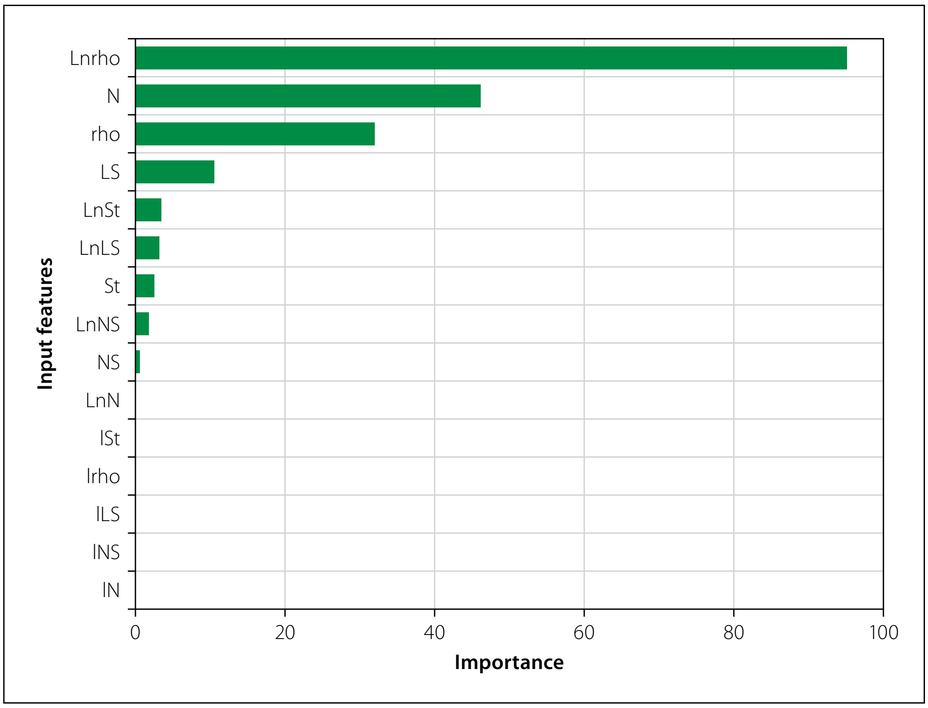

Figure 4 Feature importance of the XGBoost-HYT-CV for predicting the fundamental period

Table 5 The performance of POLYREG-HYT and multivariate linear regression

Models	Dataset	R^2 %	MAPE %	MAMPE %	MAE	RMSE	Time (s)
POLYREG-HYT	Train	99.962	3.039	1.358	0.015	0.022	251
	Test	99.957	3.363	1.428	0.015	0.022	0.001
MLR	Train	93.619	50.351	18.981	0.210	0.278	0.0345
	Test	92.630	48.244	18.668	0.200	0.278	0.001

the highest impact on the FP of infilled RC frames is the natural logarithm of the percentage of openings, followed by the number of storeys, the percentage of openings, and the length of spans. This finding highlights the importance of modifying the form of input features to explore their effect on the target and to point out the main parameter that has more impact than the others.

It is easy to see that five features have less effect on the prediction of the target variable, which are the natural logarithm of the masonry wall stiffness and the length of the spans, the masonry wall stiffness and the natural logarithm of the number of spans, and finally the number of spans. The six remaining ones do not influence the output prediction and should be omitted from the dataset. This does not mean that the stiffness of the walls is not important, since the natural logarithm of the wall stiffness is the fifth most important input feature, according to Figure 4. Furthermore, the proposed predictive model will only require the five initial input features to be defined by the user, where the prediction will be computed through the use of the proposed model.

Results obtained by the POLYREG-HYT algorithm

The benefit of using these types of ML-generated models is that the final output from the training/testing process is a closed-form formula (Adilkhanova *et al* 2022). Among the most popular white-box ML methods are multivariate linear regression (MLR), which captures the linear relationship between the independent and dependent variables (Maulud & Abdulazeez 2020), and polynomial regression (PR), which has the ability to capture the nonlinear relationship between the input and target variables. The adopted POLYREG-HYT algorithm enhances the problem of embedded optimisation of the polynomial feature selection of existing PR algorithms and can develop a closed-form prediction formula by automatically choosing the nonlinear features that provide the minimum error (Markou *et al* 2024). Furthermore, a polynomial degree of 3 was defined during the predictive model development, where the number of rounds was set to 1 000.

Table 5 presents the performance metrics of POLYREG-HYT and MLR algorithms. As can be seen, the POLYREG-HYT outperforms the MLR algorithm, especially in terms of error metrics, where for the case of MAPE, the POLYREG-HYT results in a 14.35 times lower value compared to that obtained by the MLR. Additionally, for the case of RMSE, a 12.89 times lower value is produced when using the POLYREG-HYT compared to the MLR algorithm. The explicit difference between these two methods underlines the numerical superiority of the proposed POLYREG-HYT (Markou *et al* 2024) in developing an accurate and robust closed-form formula that is able to compute the FP of RC frames with infill walls.

The optimum predictive formula obtained from POLYREG-HYT is given in Appendix A (page 36). The derived formula is large. It must be noted here that the POLYREG-HYT algorithm automatically selects the number of equation features that eventually form the final predictive formula. It must be noted at this point that, in addition to the equation developed by the software, an additional equation is constructed that can be used directly within an Excel sheet, which consists of the input features as they are found in the initial dataset. The predictive formula can then be used to predict new data that has not been used to train or test the POLYREG-HYT-generated model.

To further evaluate the numerical response of the proposed formula, Table 6 compares the derived equation by POLYREG-HYT with corresponding proposed formulae found in the literature, including building code equations. The comparison is performed through the use of reported RMSE and MAE on the same dataset published in Asteris (2016). Through observing the magnitudes of the error metrics, it is easy to conclude that the predicted FP achieved through the proposed POLYREG-HYT outperforms the other equations, confirming that the proposed formula offers the best fit for this dataset.

In order to further demonstrate the numerical capabilities of the proposed formula, Table 7 was developed which illustrates the normalisation of the RMSE and MAE error metrics for the best three equations found in the international literature, and how they compare to the

Table 6 Comparison between author and code formulae, and the derived equation from POLYREG-HYT

Author and code formulae	Training		Test	
	RMSE	MAE	RMSE	MAE
POLYREG-HYT	0.02170	0.01506	0.02159	0.01533
Crowley and Pinho (2006)	0.34619	0.27528	0.34813	0.27.317
Yahiaoui *et al* (2023)	0.07329	0.05082	0.06147	0.04411
Charalampakis *et al* (2020)	0.07337	0.05001	0.06114	0.04693
Asteris *et al* (2017)	0.35651	0.31783	0.31813	0.23339
Eurocode 8 (2005b)	0.52716	0.38381	0.52566	0.37996
Guler *et al* (2008)	0.75397	0.54543	0.75168	0.53352
Goel and Chopra (1997)	0.52636	0.38061	0.51202	0.37012
ASCE (2017)	0.50100	0.36599	0.49557	0.36223
RPA99version2003 (2003)	0.71726	0.50558	0.71421	0.49317

Table 7 Normalisation of the RMSE and MAE error metrics by assuming the POLYREG-HYT formula's RMSE = MAE = 1

Fundamental period of infilled RC frame structures formulae	Training		Test	
	RMSE	MAE	RMSE	MAE
POLYREG-HYT	1	1	1	1
Yahiaoui *et al* (2023	3.38	3.38	2.85	2.88
Charalampakis *et al* (2020)	3.38	3.32	2.83	3.06
Asteris *et al* (2017)	16.43	21.10	14.74	15.22

proposed POLYREG-HYT formula's numerical response. The results from the error metrics normalisation procedure demonstrate the superiority of the proposed predictive formula (see Appendix A on page 36), which also incorporates the impact of feature engineering. As can be seen from Table 7, the proposed POLYREG-HYT equation outperformed all predictive models for the test sets, where the RMSE and MAE of the proposed equation are more than three times lower than the equation proposed by Charalampakis *et al* (2020) and Yahiaoui *et al* (2023), and approximately fifteen times lower than the relationship proposed by Asteris *et al* (2017), which showed the lowest accuracy out of the four formulae in Table 7.

CONCLUSION

This article aimed to significantly enhance the accuracy of predictive models used to compute the FP of RC buildings with infill walls. In order to achieve this objective, three newly proposed ML algorithms (Markou *et al* 2024) were used to train and test predictive models using the Asteris (2016) dataset. In addition, the adopted ML algorithms were numerically investigated through the use of the same dataset, where their numerical responses were compared to other methods found in the international literature. The superiority of the proposed ML algorithms (Markou *et al* 2014) which can be found as open-source code (Bakas *et al* 2023) was demonstrated.

Initially, an investigation of the impact of train-test split ratios showed that the best split for this dataset is achieved when using 90% of the input data to train and 10% for testing. The finding emphasises the need to consider different splits since this affects the numerically obtained predictive model. The numerical investigation using three black-box ML methods (XGBoost-HYT-CV, DANN-MPIH-HYT and RF-HYT) to predict the FP of RC buildings with infill walls, showed that the XGBoost-HYT-CV method derived the optimum predictive model with a near perfect 99.994% R^2 score and with all error metric values being near zero (< 0.01). This is the most accurate predictive model currently found in the international literature.

According to the sensitivity analysis, it was found that the input variables, which were modified through feature engineering to generate fifteen features instead of five, the natural logarithm of the percentage of openings was the parameter that had the most influence on this design parameter, followed by the number of storeys. It was also found that XGBoost-HYT-CV was 18 times faster in training and testing the proposed predictive model (including hyperparameter tuning) compared to the algorithm proposed in Yahiaoui *et al* (2023), which was the most accurate predictive model prior to this current research work.

Finally, an equation was proposed by white-box ML that outperformed all proposed formulae found in the international literature. The proposed predictive FP formula was obtained through the use of the POLYREG-HYT ML method (Markou *et al* 2024) and was able to produce minimum error metrics on the test dataset: RMSE = 0.02159 and MAE = 0.01533. The proposed predictive equation resulted in an RMSE that was more than 285% improved compared to the two most accurate models found in the literature (Charalampakis *et al* 2020; Yahiaoui *et al* 2023), while a decrease of more than 1 500% was noted compared to the third most accurate model as had been proposed by Asteris *et al* (2017). These findings highlight the accuracy of the proposed predictive models that demonstrated numerical superiority compared to the most accurate predictive currently found in the international literature. Finally, the proposed predictive closed-form formula was compared to the results obtained by different design codes, where it was found that the design equations proposed in different building design codes derived extremely poor results. This highlights the need to update the design code formulae with more accurate ML-generated predictive models.

AUTHORS' STATEMENT

Competing interests

The authors declare that they have no conflict of interest and no financial interest.

Authors' contribution statement

- Authors Yahiaoui and Markou developed the models, interpreted the results, and wrote the manuscript.
- Authors Bakas and Dorbani conducted the analysis of the models and findings, and revised the manuscript.

Ethical and informed consent for data used

The research did not involve human participants and animals.

Data availability and access

The data, models, or code generated/used during the study are available in a repository online, in accordance with funder data retention policies.

The data used in this paper is available at the following URLs:

- https://www.sciencedirect.com/science/article/pii/S2352340916306291#s0035/
- https://doi.org/10.1016/j.dib.2016.10.002.

The code used in this paper is available at the following URL: https://machine-intelligence.ai/automl/.

REFERENCES

Ababu, E M, Markou, G & Bakas, N 2022. Using machine learning and finite element modelling to develop a formula to determine the deflection of horizontally curved steel I-beams. *Proceedings*, 14th International Conference on Agents and Artificial Intelligence (ICAART2022), Vol 3: 958–963. doi: 10.5220/0010982400003116.

Adilkhanova, I, Ngarambe, J & Yun, G Y 2022. Recent advances in black-box and white-box models for urban heat island prediction: Implications of fusing the two methods. *Renewable and Sustainable Energy Reviews*, 165: 112520. doi: 10.1016/j.rser.2022.112520.

Akiba, T, Sano, S, Yanase, T, Ohta, T & Koyama, M 2019. Optuna: A next-generation hyperparameter optimization framework. *Proceedings*, 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, pp 2623–2631. doi: 10.1145/3292500.3330701.

AlHamaydeh, M, Markou, G, Bakas, N & Papadrakakis, M 2022. AI-based shear capacity of FRP-reinforced concrete deep beams without stirrups. *Engineering Structures*, 264: 114441. doi: 10.1016/j.engstruct.2022.114441.

Amanat, K M & Hoque, E 2006. A rationale for determining the natural period of RC building frames having infill. *Engineering Structures*, 28(4): 495–502. doi: 10.1016/j.engstruct.2005.09.004.

Anifowose, F, Khoukhi, A & Abdulraheem, A 2017. Investigating the effect of training–testing data stratification on the performance of soft computing techniques: An experimental study. *Journal of Experimental and Theoretical Artificial Intelligence*, 29(3): 517–535. doi: 10.1080/0952813X.2016.1198936.

ASCE (American Society of Civil Engineers) 2017. *ASCE/SEI 7-16 Minimum Design Loads and Associated Criteria for Buildings and Other Structures*. American Society of Civil Engineers.

Asteris, P G 2016. The FP4026 Research Database on the fundamental period of RC infilled frame structures. *Data in brief*, 9: 704–709. doi: 10.1016/j.dib.2016.10.002.

Asteris, P G, Repapis, C C, Tsaris, A K, Di Trapani, F & Cavaleri, L 2015. Parameters affecting the fundamental period of infilled RC frame structures. *Earthquakes and Structures*, 9(5): 999–1028. doi: http://dx.doi.org/10.12989/eas.2015.9.5.999.

Asteris, P G, Repapis, C C, Repapi, E V & Cavaleri, L 2017. Fundamental period of infilled reinforced concrete frame structures. *Structure and Infrastructure Engineering*, 13(7): 929–941. doi: http://dx.doi.org/10.1080/15732479.2016.1227341.

Asteris, P G & Nikoo, M 2019. Artificial bee colony-based neural network for the prediction of the fundamental period of infilled frame structures. *Neural Computing and Applications*, 31(9): 4837–4847. doi: 10.1007/s00521-018-03965-1.

Bakas, N P, Markou, G, Charmpis, D C & Hadjiyiannakou, K 2021. Performance and scalability of deep learning models trained on a hybrid supercomputer: Application in the prediction of the shear strength of slender RC beams. In: *ECCOMAS Thematic Conference on Computational Methods in Structural Dynamics and Earthquake Engineering*, Vol 2021.

Bakas, N, Markou, G, Langousis, A, Lavdas, S & Chatzichristofis, S 2023. NBML: computer software for data analysis and predictive modelling with artificial intelligence algorithms. Online: https://machine-intelligence.ai/automl/.

Belyadi, H & Haghighat, A 2021. Chapter 5 – Supervised Learning. *Machine learning guide for oil and gas using Python*, pp 169–295. doi: 10.1016/B978-0-12-821929-4.00004-4.

Breiman, L 2001. Random forests. *Machine learning*, 45: 5–32.

Cai, Z, Long, Y & Shao, L 2019. Classification complexity assessment for hyper-parameter optimization. *Pattern Recognition Letters*, 125: 396–403. doi: 10.1016/j.patrec.2019.05.021.

Calitz, D, Markou, G, Bakas, N P & Papadrakakis, M 2023. Developing fundamental period formulae for steel framed structures through machine learning and automated algorithms. In: *ECCOMAS Thematic Conference on Computational Methods in Structural Dynamics and Earthquake Engineering*.

Charalampakis, A E, Tsiatas, G C & Kotsiantis, S B 2020. Machine learning and nonlinear models for the estimation of fundamental period of vibration of masonry infilled RC frame structures. *Engineering Structures*, 216: 110765. doi: 10.1016/j.engstruct.2020.110765.

Carstens, N, Markou, G & Bakas, N 2022. Improved predictive fundamental period formula for reinforced concrete structures through the use of machine learning algorithms. In: *ICAART* 2: 647–652. doi: 10.5220/0010984500003116.

Chen, T & Guestrin, C 2016. XGBoost: A scalable tree boosting system. *Proceedings.* 22nd Acm Sigkdd International Conference on Knowledge Discovery and Data Mining, pp 785–794. doi: 10.1145/2939672.2939785.

Chennit, M, Ahmed-Chaouch, A, Saidani, M & Bourzam, A 2022. Periods and mode shapes for uniform shear wall buildings: Importance of selecting the appropriate dynamic behavior. *International Journal of Structural Stability and Dynamics*, 22(11): 2250114. doi: 10.1142/S0219455422501140.

Chibaya, VT, Markou, G & Bakas, N 2023. Development of formulae for the section rotations due to bending of curved steel I-beams through AI and Ml algorithms. *Proceedings of the Compdyn* (ECCOMAS Thematic Conference on Computational Methods in Structural Dynamics and Earthquake Engineering).

Code, P 2005a. Eurocode 2: Design of Concrete Structures Part 1–1: General Rules and Rules for Buildings. *British Standard Institution, London*, 668: pp 659–668.

Code, P 2005b. Eurocode 8: Design of Structures for Earthquake Resistance Part 1: General Rules, Seismic Actions and Rules for Buildings. *Brussels: European Committee for Standardization*, 10.

Crowley, H & Pinho, R 2006. Simplified equations for estimating the period of vibration of existing buildings. In: *First European conference on earthquake engineering and seismology*, 1122: 1–10.

Dorbani, S, Badaoui, M & Benouar, D 2015. Impact of natural period and epicentral distance on storey lateral displacements. *spectrum*, 11: 14.

Freund, Y & Schapire, R E 1997. A decision-theoretic generalization of on-line learning and an application to boosting. *Journal of Computer and System Sciences*, 55(1): 119–139. doi: 10.1006/jcss.1997.1504.

Friedman, J H 1991. Multivariate adaptive regression splines. *The Annals of Statistics*, 19(1): 1–67.

Friedman, J H 2001. Greedy function approximation: A gradient boosting machine. *Annals of Statistics*, 29(5): 1189–1232. doi: 10.1214/AOS/1013203451.

Gallipoli, M R, Petrovic, B, Calamita, G, Tragni, N, Scaini, C, Barnaba, C, Vona, M & Parolai, S 2023. Towards specific T–H relationships: FRIBAS database for better characterization of RC and URM buildings. *Bulletin of Earthquake Engineering*, 21(4): 2281–2307. doi: 10.1007/s10518-022-01594-7.

Goel, R K & Chopra, A K 1997. Period formulas for moment-resisting frame buildings. *Journal of Structural Engineering*, 123(11): 1454–1461. doi: https://doi.org/10.1061/(ASCE)0733-9445(1997)123:11(1454).

Gravett, D Z, Mourlas, C, Taljaard, V L, Bakas, N, Markou, G & Papadrakakis, M 2021. New fundamental period formulae for soil-reinforced concrete structures interaction using machine learning algorithms and ANNs. *Soil Dynamics and Earthquake Engineering*, 144: 106656. doi: 10.1016/j.soildyn.2021.106656.

Guler, K, Yuksel, E & Kocak, A 2008. Estimation of the fundamental vibration period of existing RC buildings in Turkey utilizing ambient vibration records. *Journal of Earthquake Engineering*, 12(S2): 140–150. doi: 10.1080/13632460802013909.

Ilemobayo, J A, Durodola, O, Alade, O, Awotunde, O J, Olanrewaju, A T, Falana, O, Ogungbire *et al* 2024. Hyperparameter tuning in machine learning: A comprehensive review. *Journal of Engineering Research and Reports*, 26(6): 388–395. doi: 10.9734/jerr/2024/v26i61188.

Kaggle (n.d.). Kaggle: Your Home for Data Science. Available at: https://www.kaggle.com/.

Ke, G, Meng, Q, Finley, T, Wang, T, Chen, W, Ma, W, Ye, Q & Liu, T Y 2017. Lightgbm: A highly efficient gradient boosting decision tree. *Advances in neural information processing systems*, 30: 3149–3157.

Khelfi, M, Bourahla, N & Remki, M 2021. Performance evaluation of masonry infilled RC frame structures under lateral loads. *Građevinar*, 73(3): 219–234. doi: 10.14256/JCE.2647.2019.

Kose, M M 2009. Parameters affecting the fundamental period of RC buildings with infill walls. *Engineering Structures*, 31(1): 93–102. doi: 10.1016/j.engstruct.2008.07.017.

Kotsiantis, S & Pintelas, P 2004. Combining bagging and boosting. *International Journal of Computational Intelligence*, 1(4): 324–333. doi: 10.1103/PhysRevD.77.085025.

Latif, I, Banerjee, A & Surana, M 2022. Explainable machine learning aided optimization of masonry infilled reinforced concrete frames. *Structures*, 44: 1751–1766. doi: 10.1016/j.istruc.2022.08.115.

Markou, G & Bakas, N P 2021. Prediction of the shear capacity of reinforced concrete slender beams without stirrups by applying artificial intelligence algorithms in a big database of beams generated by 3D nonlinear finite element analysis. *Computers and Concrete*, 28(6): 433–447. doi: 10.12989/cac.2021.28.6.533.

Markou, G, Bakas, N P, Chatzichristofis, S A & Papadrakakis, M 2024. A general framework of high-performance machine learning algorithms: Application in structural mechanics. *Computational Mechanics*, 73(4): 705–729. doi: 10.13140/RG.2.2.14962.89287.

Maulud, D & Abdulazeez, A M 2020. A review on linear regression comprehensive in machine learning. *Journal of Applied Science and Technology Trends*, 1(2): 140–147. doi: 10.38094/jastt1457.

Mirrashid, M & Naderpour, H 2022. Computational intelligence-based models for estimating the fundamental period of infilled reinforced concrete frames. *Journal of Building Engineering*, 46: 103456. doi: 10.1016/j.jobe.2021.103456.

Musolf, A M, Holzinger, E R, Malley, J D & Bailey-Wilson, J E 2022. What makes a good prediction? Feature importance and beginning to open the black box of machine learning in genetics. *Human Genetics*, 141(9): 1515–1528. doi: 10.1007/s00439-021-02402-z.

Naderpour, H, Eidgahee, D R, Fakharian, P, Rafiean, A H & Kalantari, S M 2020. A new proposed approach for moment capacity estimation of ferrocement members using Group Method of Data Handling. *Engineering Science and Technology, an International Journal*, 23(2): 382–391. doi: 10.1016/j.jestch.2019.05.013.

Naveen Venkatesh, S, Arun Balaji, P, Elangovan, M, Annamalai, K, Indira, V, Sugumaran, V & Mahamuni, V S 2022. Transfer learning-based condition monitoring of single point cutting tool. *Computational Intelligence and Neuroscience*, 2022(1): 3205960. doi: 10.1155/2022/3205960.

Ngab, A S, Attia, N J & Shahin, M S 2021. Effects of reinforced concrete building characteristics on the fundamental natural period using the Rayleigh Method. *International Journal of Engineering Research & Technology*, 10(11): 269–278.

Nikoo, M, Zarfam, P & Nikoo, M 2012. Determining displacement in concrete reinforcement building with using evolutionary artificial neural networks. *World Applied Sciences Journal*, 16(12): 1699–1708.

Optuna – A hyperparameter optimization framework (n.d.). Available at: https://optuna.org/.

Prokhorenkova, L, Gusev, G, Vorobev, A, Dorogush, A V & Gulin, A 2018. CatBoost: Unbiased boosting with categorical features. *Advances in Neural Information Processing Systems*, 31: 1–23.

Quinlan, J R 1992. Learning with continuous classes. In: *5th Australian Joint Conference on Artificial Intelligence*, 92: 343–348.

RandomForestRegressor – SCIKIT-LEARN 1.6.1 documentation (n.d.). Available at: https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.RandomForestRegressor.html.

Rathakrishnan, V, Beddu, S B & Ahmed, A N 2022. Predicting compressive strength of high-performance concrete with high volume ground granulated blast-furnace slag replacement using boosting machine learning algorithms. *Scientific Reports*, 12(1): 9539. doi: 10.1038/s41598-022-12890-2.

Reconan FEA v2.00 User's Manual 2020. Available at: https://www.researchgate.net/publication/342361609_ReConAn_v200_Finite_Ele- ment_Analysis_Software_User's_Manual.

RPA99version2003 (2003). Algerian Seismic Regulations RPA99 Version 2003 – DTR BC 2.48. Earthquake Engineering National Research Centre (CGS).

Salehin, I & Kang, D K 2023. A review on dropout regularization approaches for deep neural networks within the scholarly domain. *Electronics*, 12(14): 3106. doi: 10.3390/electronics12143106.

Sarvepalli, S K 2015. Deep learning in neural networks: The science behind an artificial brain. Liverpool Hope University, Liverpool. doi: 10.13140/RG.2.2.22512.71682.

Somala, S N, Karthikeyan, K & Mangalathu, S 2021. Time period estimation of masonry infilled RC frames using machine learning techniques. *Structures*, 34: 1560–1566 (online). doi: 10.1016/j.istruc.2021.08.088.

Spijkerman, Z, Bakas, N P, Markou, G & Papadrakakis, M 2021. Predicting the shear capacity of reinforced concrete slender beams without stirrups by applying artificial intelligence algorithms. In: *International Conference on Computational Methods in Structural Dynamics and Earthquake Engineering*. doi: 10.7712/120121.8749.18602.

Taljaard, V L, Gravett, D Z, Mourlas, C, Markou, G, Bakas, N P & Papadrakakis, M 2021. Development of a new fundamental period formula by considering soil-structure interaction with the use of machine learning algoritms. In: *International Conference on Computational Methods in Structural Dynamics and Earthquake Engineering* (vol 2021). doi: 10.7712/120121.8748.18534.

Thisovithan, P, Aththanayake, H, Meddage, D P P, Ekanayake, I U & Rathnayake, U 2023. A novel explainable AI-based approach to estimate the natural period of vibration of masonry infill reinforced concrete frame structures using different machine learning techniques. *Results in Engineering*, 19: 101388. doi: 10.1016/j.rineng.2023.101388.

Verma, A & Sahoo, D R 2022. Application of GMDH model for predicting the fundamental period of regular RC infilled frames. *Steel and Composite Structures, an International Journal*, 42(1): 123–137.

Verdonck, T, Baesens, B, Óskarsdóttir, M & Van den Broucke, S 2024. Special issue on feature engineering editorial. *Machine learning*, 113(7): 3917–3928. doi: 10.1007/s10994-021-06042-2.

Van der Westhuizen, A M, Markou, G & Bakas, N P 2022. Development of a new fundamental period formula for steel structures considering the soil-structure interaction with the use of machine learning algorithms. In: *International Conference on Agents and Artificial Intelligence* (vol 3): 952–957. doi: 10.5220/0010978400003116.

Van der Westhuizen, A M, Bakas, N & Markou, G 2023. Developing an artificial neural network model that predicts the fundamental period of steel structures using a large dataset. *Proceedings*, ECCOMAS Thematic Conference on Computational Methods in Structural Dynamics and Earthquake Engineering.

XGBoost Documentation (n.d.). Available at: https://xgboost.readthedocs.io/en/stable/.

XGBoost Parameters – XGBoost 2.1.3 documentation (n.d.). Available at: https://xgboost.readthedocs.io/en/stable/parameter.html.

Yahiaoui, A, Dorbani, S & Yahiaoui, L 2023. Machine learning techniques to predict the fundamental period of infilled reinforced concrete frame buildings. *Structures*, 54: 918–927. doi: 10.1016/j.istruc.2023.05.052.

Yahiaoui, A, Matos, J C & Dorbani, S 2024. Probabilistic forecast of concrete compressive strength using ML. In: *International Probabilistic Workshop* (pp 281–286). Cham: Springer Nature Switzerland. doi: **https://doi.org/10.1007/978-3-031-60271-9_25**.

Ying, X 2019. An overview of overfitting and its solutions. *Journal of Physics: Conference Series*, 1168: 022022). doi: **10.1088/1742-6596/1168/2/022022**.

Yousefi, J & Hamilton-Wright, A 2014. Characterizing EMG data using machine-learning tools. *Computers in Biology and Medicine*, 51: 1–13. doi: **10.1016/j.compbiomed.2014.04.018**.

Zhang, X 2010. Regularization. *Encyclopedia of Machine Learning*, Sammut, Cl, Springer US, Boston, MA, pp 845–849. doi: **10.1007/978-0-387-30164-8_712**.

APPENDIX A

$$
\begin{aligned}
T = {} & 4.11397LnN \times LnNS ++ 1.82087LnNS \times 1LS \times 1rho ++ 9.28386St + 329340LnN - 5.75649LnN^2 \times LnLS - 0.200372N \times NS - 6623.5Lnrho \\
& \times 1LS \ast 1rho - 4.24726LS \times 1rho^2 - 25346.7NS \times LnNS \times 1rho - 15.6833N \times 1LS \times 1rho - 5.05368 \times 10^{-4} \times St^2 \times 1LS + 0.514668rho \\
& \times LnN \times LnNS - 0.0460130N \times LS \times 1rho + 2.82722 \times 10^{-4}NS \times LnSt \times 1NS + 0.831854LS \times LnLS \times 1N + 7.94948LnSt \times 1NS \times 1LS \\
& + 0.0628509N \times St \times 1rho - 12.7391St \times LnNS - 0.0210832N \times rho \times St - 4.62920Lnrho \times LnSt \times 1St - 130025 \times 1N^2 - 174223LnN \times \\
& 1N - 0.170341 \times rho^2 \times LnN +10.3640St \times Lnrho^2 + 0.0739743N \times St \times Lnrho + 35635.6LnNS \times 1NS^2 - 0.195877LS \times LnN - 6.48996 \times \\
& 10^{-3}N \times LS \times LnLS + 0.0189N \times rho \times LnSt - 7.61595 \times 10^{-3}N \times NS \times LS - 2.51269NS \times LnNS \times LnLS + 52506.4rho \times 1rho + 11.9337N \\
& \times LnN \times 1LS - 8.03561rho \times LnSt \times 1LS - 1.47328St \times LnNS \times 1rho - 35.4988N \times rho + 0.608619LS \times Lnrho \times 1N + 16111.1NS \times rho \times \\
& 1NS + 34.9003N \times Lnrho + 22.7390N \times rho \ast Lnrho - 688613N \times 1N + 10.2640LnN \times LnLS^2 - 0.309246LnN \times 1St^2 - 0.0241779N \times LnSt \\
& \times 1rho + 4.29668 \times 10^{-4}LS^2 \times LnNS - 4.72420N \times rho^2 - 0.384954NS \times LnN \times 1LS - 0.505677NS \times 1N^2 - 0.139586N^2 \times 1LS + 1.06919 \\
& \times 1rho \times 1St + 10.5917Lnrho \times LnSt \ast 1LS - 0.180778LnSt \times 1N \times 1LS + 0.138427 \times N \times 1St - 4.17222 \times 10^{-4} \times rho \times St^2 - 1446.54rho^2 \\
& \times Lnrho - 40262.1rho \times LnNS \times LnSt +1.56319LnLS \times LnSt \times 1NS +40301LS \times LnNS \times 1LS + 4.73870LnSt \times 1rho^2 - 0.0140509LnN \times \\
& LnSt \times 1rho + 21513.2rho \times 1LS \ast 1rho + 6832.99rho \times LnSt \times 1NS - 2.59493LnN \times 1rho^2 - 0.0699796N \times 1NS \times 1St + 1.54376LS \times LnSt \\
& \times 1LS - 1123.32rho \times Lnrho \times 1LS + 5.80491St \times LnNS^2 - 68.5523rho^2 \times \mathrm{LnLS} + 0.0246112LS \times LnNS \times Lnrho + 7.07735 \times 10^{-4}LS \times St \\
& \times LnSt - 673.335NS^2 \times LnSt + 40311.3LnNS \times 1LS + 0.657393LnN \times LnLS \times Lnrho + 16.8726LS \times rho \times Lnrho - 1.44555NS \times rho \times St \\
& + 5.08601rho \times 1N \times 1LS + 5.46822LnLS^2 \times 1NS - 10.7416 \times 1LS^2 \times 1rho + 0.418503N \times LnNS \times LnLS + 1.52144LS \times 1NS^2 - 20.2278N \\
& \times Lnrho \times 1LS + 8.55766Lnrho \times LnSt - 0.0186501N \times LnNS \times 1St - 3.16835LnN \times LnNS \times LnLS + 73912.LnNS \times 1rho - 338782LnN \times \\
& 1N^2 + 0.432278N \times LnLS \times 1NS + 7.85246NS \times LnLS - 1290.83NS \times LnNS \times Lnrho +22529.1N \times LnN - 24.5144St \times 1NS^2 - 0.557845rho \\
& \times 1N \times 1St + 0.0121602N \times NS \times 1rho - 0.0188587St \times LnLS + 6.74112N \times rho \times 1LS + 1.5089LnLS \times LnSt \times 1LS - 3.09527St \times 1rho^2 \\
& - 2.69044 \times 10^{-3}LS \times St \times 1NS + 2.3641LnN \times Lnrho \times 1NS + 6286.25NS^2 \times 1rho - 3.38294 \times 10^{-3}LS \times LnSt^2 + 15.6265LnNS \times 1N \times \\
& 1LS - 4.45551LnN \times 1NS - 2.18681LnLS \times 1N \times 1NS - 6.53108Lnrho \times 1rho \times 1St +413347LnN^2 \times 1N - 0.0618923N \times St - 2104.39N \times \\
& LnN^2 + 14958.5LS \times Lnrho \times 1LS + 12894.5NS \times rho \times LnSt - 622.301rho \times LnLS \times 1LS + 3376.29NS \times Lnrho \times 1NS + 0.301038LS \times LnN \\
& \times LnNS + 5124.11Lnrho^2 \times 1LS - 0.332967N \times 1LS \times 1St - 0.0129903LS \times St \times Lnrho - 4211.12 \ast LnNS^3 + 310041N^2 \times 1N - 0.0458706N \\
& \times LnN \times 1rho - 378609N + 6701.17rho \times LnNS \times 1rho + 0.0123205N \times NS^2 + 34924.1rho^2 \times 1rho - 39.5669rho \times LnLS^2 + 986188N \ast \\
& 1N^2 + 1.30686N \times LnLS + 27.7497N \times LS \times 1LS - 6.03934rho \times LnN \times 1rho + 53996.8 \times 1NS \times 1rho - 34.2356LnN \times LnLS + 5.8867 \times \\
& 10^{-6}St^2 \times LnLS + 0.0963686LS \times LnN \times 1N - 25257.4LnNS \times Lnrho^2 + 1.41640 \times 10^{-3}LS \times LnN \times LnSt + 8.53747 \times 10^{-3} LS \times rho \times St - \\
& 1.37322LnN \times Lnrho \times 1LS - 2.21568rho \times 1N^2 + 19.2185N^2
\end{aligned}
$$

APPENDIX B

A glossary of acronyms and terms related to machine learning

- **Regularised model:** The problem of overfitting is usually controlled by including a regularisation, or in other words, a penalty. Regularisation is essential in all machine learning techniques and is also employed to reduce the model complexity (Zhang 2010).
- **Kaggle session platform:** Kaggle is a platform that provides powerful tools and resources regarding the data science field, for 12 hours every day (Kaggle n.d.).
- **Optuna framework:** Optuna is one of several open-source frameworks for tuning the hyperparameters, provided under the MIT licence (Akiba *et al* 2019; Optuna – A hyperparameter optimization framework n.d.).
- **Hyperparameters**: Hyperparameters are the parameters that control the training and structure of machine learning models. In contrast to the parameters learned during the training, hyperparameters are determined before the training starts (Ilemobayo *et al* 2024).
- **Hyperparameter tuning:** Hyperparameter tuning is a critical step in machine learning. The aim is to find the optimal combination of hyperparameters that leads to the best model performance (Cai *et al* 2019).
- **max_depth:** The max_depth refers to the maximum depth of the tree in all boosting and tree machine learning types (Belyadi & Haghighat 2021).
- **Learning rate:** The learning rate ranges between 0 and 1, and it specifies the step size, meaning that in all trees, the weight will be multiplied by this hyperparameter (Belyadi & Haghighat 2021).
- **n_estimators:** The **n_estimators** refer to the number of trees (Belyadi & Haghighat 2021).
- **Colsample_bytree:** The Colsample_bytree denotes the subsampling ratio of columns utilised during the construction of each tree; it happens once for each tree built (XGBoost Parameters – XGBoost 2.1.3 Documentation n.d.).
- **Subsample:** This value between 0 and 1 helps control the common overfitting problem. It defines the training instances ratio, meaning that if this ratio is configured to 0, XGBoost will randomly select fifty percent of the training data before constructing trees (XGBoost Parameters – XGBoost 2.1.3 Documentation n.d.).
- **min-samples split**: Minimum number of samples needed for dividing an internal node (RandomForestRegressor – scikit-learn 1.6.1 Documentation n.d.).
- **max_features**: The number of features to take into account when searching for the ideal split (RandomForestRegressor – scikit-learn 1.6.1 Documentation n.d.).
- **min_samples_leaf:** This specifies the minimum number of samples that must

exist at a leaf node. By default it is 1 (RandomForestRegressor – scikit-learn 1.6.1 Documentation n.d.).

- **Artificial neural network architecture:** In a neural network, there are several layers, with each layer containing several neurons. Each neuron in a layer receives as input the outputs of all the other neurons in the previous layer. The output layer can have several outputs.
- **Dropout rate:** This is one of the most known regularisation techniques, mainly employed in neural networks to control overfitting (Salehin & Kang 2023).
- **Momentum:** This algorithm is an improved version of the gradient descent, where the weights are updated after each batch (Nikoo *et al* 2012).
- **Number of epochs:** The number of iterations in a neural network (Kose 2009).
- **Batch size:** The batch size determines the number of samples utilised to calculate the gradient descent at each iteration to update the neural network's weights (Naveen Venkatesh *et al* 2022).