Journal of Building Engineering 44 (2021) 102977

Contents lists available at ScienceDirect

Journal of Building Engineering

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

StructuresNet and *FireNet*: Benchmarking databases and machine learning algorithms in structural and fire engineering domains

M.Z. Naser [a,b,1,*], Venkatesh Kodur [c], Huu-Tai Thai [d], Rami Hawileh [e], Jamal Abdalla [e], Vitaliy V. Degtyarev [f]

[a] *Glenn Department of Civil Engineering, Clemson University, Clemson, SC, 29634, USA*
[b] *AI Research Institute for Science and Engineering (AIRISE), Clemson University, Clemson, SC, 29634, USA*
[c] *Department of Civil and Environmental Engineering, Michigan State University, East Lansing, MI, USA*
[d] *Department of Infrastructure Engineering, The University of Melbourne, Parkville, VIC, 3010, Australia*
[e] *Department of Civil Engineering, American University of Sharjah, Sharjah, United Arab Emirates*
[f] *New Millennium Building Systems, LLC, 3700 Forest Dr. Suite 501, Columbia, SC, 29204, USA*

ARTICLE INFO

Keywords:
Machine learning
Artificial intelligence
Validation
Databases
Structures
Fire

ABSTRACT

Machine learning (ML) continues to rise as an effective and affordable method of tackling engineering problems. Unlike other disciplines, the integration of ML into structural and fire engineering domains remains deficient. This is due in part to the lack of benchmark databases to compare the effectiveness of ML models. In order to bridge this knowledge gap, this paper presents a benchmark examination of common supervised learning ML algorithms that can be easily deployed into structural and fire engineering problems. The selected algorithms include; Decision Trees (DT), Random Forest (RF), Extreme Gradient Boosted Trees (ExGBT), Light Gradient Boosted Trees (LGBT), TensorFlow Deep Learning (TFDL), and Keras Deep Residual Neural Network (KDP), and are used with their default values to establish a proper benchmark against six databases. The compiled datasets have been thoroughly tested and span two domains, structural engineering; 1) elemental response of concrete-filled steel tubular (CFST) circular columns at ambient conditions, 2) shear response of cold-formed steel (CFS) channels with slotted webs, 3) compressive strength of concrete, 4) fatigue life data, 5) shear strength of reinforced concrete (RC) beams and FRP-strengthened RC beams; and fire engineering, 6) fire behavior of RC concrete columns in terms of spalling occurrence and fire resistance. This study also investigates a variety of commonly used performance metrics that are applicable to regression and classification-based ML problems. We invite ML users to apply their models to the presented databases to establish a benchmark by mean of external validation and then extend their models to other problems and databases. Collectively, the presented work establishes the first step towards a unified framework that can be used to accelerate the adoption of ML into structural and fire engineering domains.

1. Introduction

Engineers tend to design experiments to examine problems [1]. Naturally, such experiments are limited in their number of test specimens and parameters, and are also articulated to fit the available testing equipment and facilities. To ensure comparability of tests, testing standards (i.e., ASTM, ISO etc.) were established. Such standards provide a unified reference for test methods, equipment, and specifications for various testing scenarios and environments. Regardless of the test in question, the notion of a testing standard is to ensure compliance to a certain procedure that allows duplication of results among different stakeholders. Within the structural and fire engineering domains, testing standards primarily exist for material property (e.g. testing for compressive strength of concrete etc.), and examination of elemental behavior (to some extent) [2–4].

In lieu of experiments, engineers may also utilize advanced numerical tools such as finite element (FE) methods to model material and elemental response at ambient or fire conditions. Advancements in such

* Corresponding author. Glenn Department of Civil Engineering, Clemson University, Clemson, SC, 29634, USA.
E-mail addresses: mznaser@clemson.edu (M.Z. Naser), kodur@egr.msu.edu (V. Kodur), tai.thai@unimelb.edu.au (H.-T. Thai), rhaweeleh@aus.edu (R. Hawileh), jabdalla@aus.edu (J. Abdalla), vitaliy.degtyarev@newmill.com, vitdegtyarev@yahoo.com (V.V. Degtyarev).
[1] www.mznaser.com.

https://doi.org/10.1016/j.jobe.2021.102977
Received 29 April 2021; Received in revised form 7 July 2021; Accepted 11 July 2021
Available online 17 July 2021

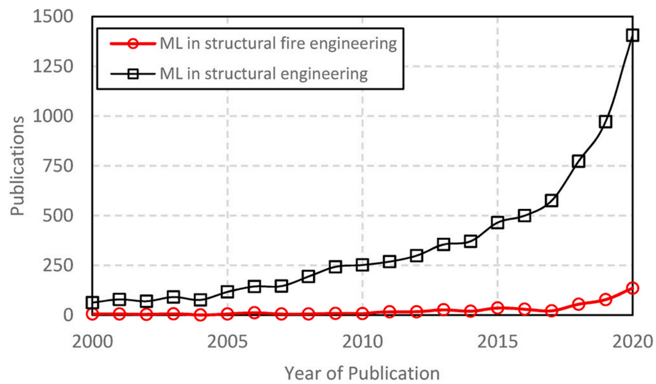

Fig. 1. ML-based publication trends during 2000–2020.

numerical tools were made possible as a result of developing improved computing workstations with fast processing units that can be obtained conveniently. In a way, numerical methods provide users with affordable and "logical" means to predict structural and fire engineering phenomena – noting how FE is founded upon solving partial differential equations which are formulated through functional minimization techniques [5].

A common practice of utilizing FE methods is to first validate predictions from a developed FE model against that obtained from a real test. Such validation is often displayed in terms of a chart with two series; thereby comparing predictions from FE model and measurements from experiments [6,7]. The open literature seems to agree that a "good validation" is that which has a 5–20% variation between FE predictions and test measurements [8–13]. However, establishing a good agreement is often subjective as we continue to lack a standardized method to establish such validation. Despite the lack of a standardized procedure not only to develop a FE model, or carry out a simulation but also to validate such a simulation, the use of FE modeling is regarded as a cornerstone within the civil engineering industry [14–18].

Still, a few questions might arise that may question the suitability of FE simulation as a tool. For example, what are the recommended element types to be used in a specific FE model aimed to explore a given phenomenon? What material models are considered proper to model such a phenomenon? What convergence criteria and solution technique should one use in a modeling a specific problem? Moreover, what constitutes a good FE model? And how can we ensure that a developed FE model can be safely applied beyond its intended use or range of applicability? In a way, and at this day and age, the use of FE modeling seems to be treated as propriety information, and often regarded as an art with a complementary scientific component [19].

The above also brings in a few questions in light of the rise of Machine learning (ML) as a potential new method for tackling structural and fire engineering problems [20–23]. Simply put, ML can be thought of ML is a universal method that can be applied to virtually any engineering problem. In a way, ML can be thought of as a software that can be applied to explore the observations noted in our databases. Thus, the following questions may primarily be of importance, especially to researchers and practicing engineers interested in adopting ML. For example: What algorithms can one use? Are all algorithms the same? Do algorithms need to be developed from scratch? Or can existing algorithms (pre-developed) be used as is? Where to get data to develop ML models? How to validate such models? As one can see, these questions mirror those outlined above and others not mentioned herein for brevity (e.g., what coding language to use?). The authors of this work believe that addressing the above questions early and during the current rise of ML will not only be beneficial to this community but will help facilitate the adoption of ML as a new method of choice. As such, the primary motivation behind this work builds upon similar calls for developing validation and benchmarking procedures for FE models [24–27] and aims to set the stage toward a unified ML procedure within structural and fire engineering domains and by structural and fire engineers.

A closer look into the open literature shows that publications with a ML theme in structural and/or fire engineering continue to steadily rise (see Fig. 1). Noting how parallel fields have embraced ML indicates that ML will continue this positive trend. In addition to our examination of global trends, a deep dive into the open literature highlights how ML has been successfully used in a variety of problems. For example, Behnood and Golafshani [28,29] developed a series of ML models to examine properties of concrete derivatives (traditional concrete, concrete with waste foundry sand, and high-performance concrete), and asphalt materials with notable success and have led to creating new and simple models that can predict the properties of concrete and asphalt materials. In addition, the works of Mangalathu et al. [30,31] showcased how ML models can be used to predict the seismic and structural response of concrete shear walls and bridges, which has also led to developing open-source classification models. Degtyarev [32,33] successfully developed a database and an Artificial Neural Network (ANN) to examine the response of shear strength of CFS channels with slotted webs with high accuracy exceeding 95%. The collaboration of Lopes and Bobadilha [34,35] has resulted in novel ML models that were applied to evaluating the quality of timber materials. These models achieved notable accuracy exceeding 75%, and were shown to be convertible into mobile phone applications. Additional works that applied ML into this domain were also identified by other research groups [36–44], and carried out by the authors [45–52].

This work aims to present a benchmarking study that applies commonly used supervised learning algorithms (with default settings) to publicly available databases with a goal to establish a first documentation and examination for validation of ML models. The selected algorithms include; Decision Trees (DT), Random Forest (RF), Extreme Gradient Boosted Trees (ExGBT), Light Gradient Boosted Trees (LGBT), TensorFlow Deep Learning (TFDL), and Keras Deep Residual Neural Network (KDP). Furthermore, this paper showcases commonly used validation and performance metrics that can be applied to regression and classification problems by examining six large datasets covering concrete, and steel materials and structures, and conveniently named StructuresNet and FireNet.

The intend of this paper is to outline a systematic procedure to maintain repeatability and benchmarking of commonly used ML models within the structural and fire engineering domains. We would like to emphasize that the goal of the shown analysis is not to finetune algorithms to report upon the best performing algorithm to examine a particular problem, nor to guarantee tuning of algorithms to chase high metrics, but rather to apply the selected algorithms in their default settings to allow interested readers from repeating this work to compare the performance of their algorithms and then aim to develop improved models (both of which may encompass similar or other types of algorithms to that used herein). By using algorithms in their default settings, the attained performance of these algorithms is then "benchmarked" and documented. Such benchmarking will allow future ML users from also benchmarking newly developed ML models or ensembles and compare their performance against that of the most commonly used models reported in our domains on the presented databases. We hope that this paper founds an approach that can be further massaged by the collective works in our domains to establish a uniform, and possibly standardized, mean to apply ML models to fully harness the positive potential of this technology in the near future. The message of this work aligns with that proposed by other researchers that focused on FE models [24–27].

1.1. Description of databases

This section describes the examined six databases of varying size and scope within the structural and fire engineering disciplines in details. These databases cover re-occurring problems of simple and complex nature and fall under regression and classification problems.

Table 1
Details on StructuresNet and FireNet.

Database	Domain	Application	Category	No. of data points	Basis	References
Thai database	Structural engineering (StructuresNet)	Design of CFST columns	Regression	3103	170 tests	[58,59]
Degtyarev & Degtyareva database		Shear strength of CFS channels with slotted webs		3512	FE simulations	[32]
Yeh database		Compressive strength of high-performance concrete		1030		[57]
Abdalla & Hawileh database		Fatigue life data		59	Tests	[48]
Abdalla et al. database		Shear strength of RC and FRP-strengthened beams		290	Tests	[55,56]
Naser & Kodur database	Fire engineering (FireNet)	Fire resistance and spalling response of reinforced concrete columns	Regression and Classification (binary and multi-class)	306	140 tests and 169 FE simulations	[53,54]

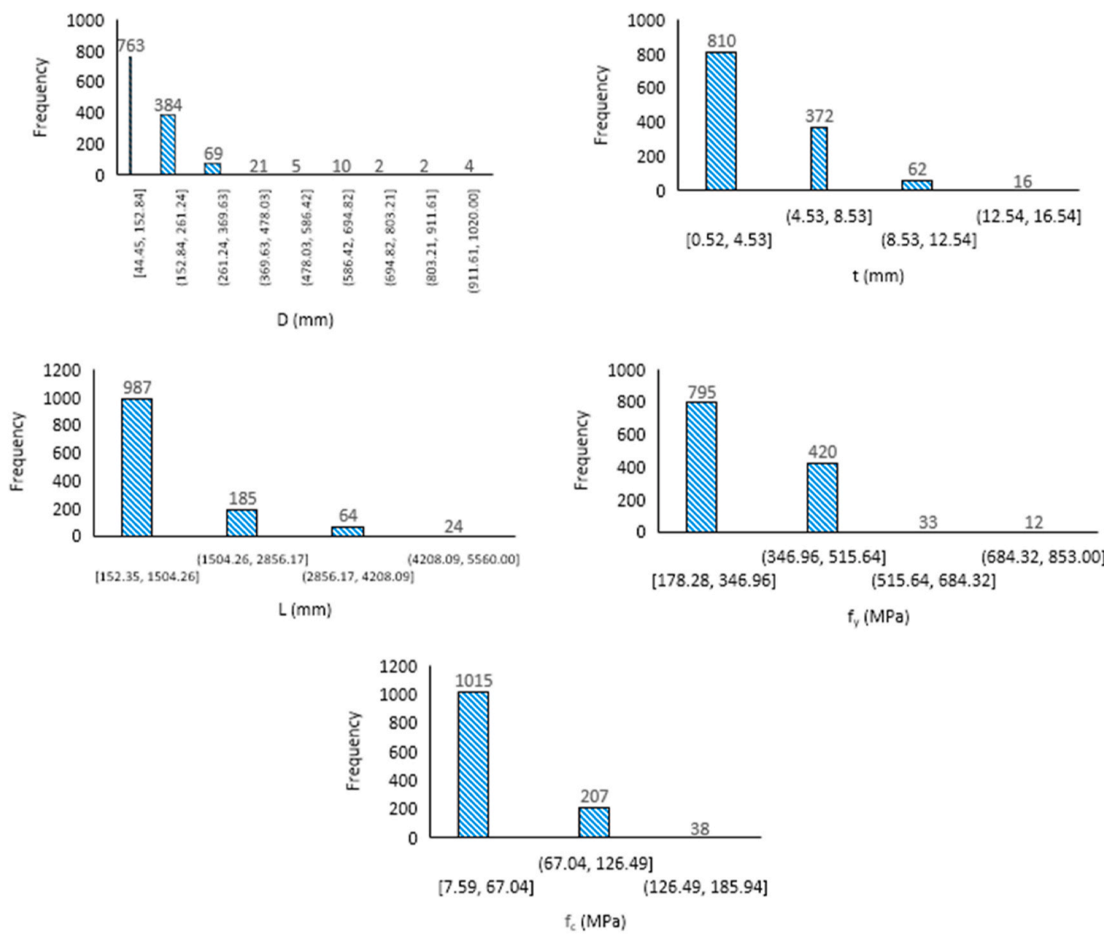

Fig. 2. Frequency of identified features of selected CFST in the compiled database.

Information with regard to the history of each database, and to statistical distribution are provided herein for brevity. Please note that a more in-depth analysis on each database can be found in their respective references [32,48,53–59]. Table 1 gives insights into the presented databases. One should also note that there is very limited data on fire-exposed structural elements and structures; which is reflected by the smaller size of FireNet as opposed to StructuresNets. All of the presented databases are hosted online on Mendeley public repositories, as well as original papers (and complete links to these databases are shown herein: Database 1 [59]. Database 2 [122]. Database 3 [123]. Database 4 [63,

Table 2
Key statistics from CFST database.

Section	Features	D (mm)	t (mm)	D/t	L_e (mm)	f_y (MPa)	f'_c (MPa)
Circular (concentric loading)	Min	44.45	0.52	7.42	152.35	178.28	9.17
	Max	1020.00	16.54	220.93	5560.00	853.00	193.30
	Average	158.52	4.31	44.28	1060.53	336.35	50.21
	Standard deviation	105.42	2.45	32.37	1005.28	90.89	31.57
	Median	127.3	4.00	33.33	662.00	325.00	41.00
	Skewness	3.71	1.58	2.86	1.98	2.18	2.06
Parameter		*D*	f_c	f_y	L_e	*t*	*N*
D		1.000					
f_c		-0.003	1.000				
f_y		0.072	0.030	1.000			
L_e		0.201	-0.154	0.080	1.000		
t		0.478	-0.022	0.238	0.216	1.000	
N		**0.911**	0.126	0.145	0.109	**0.549**	1.000

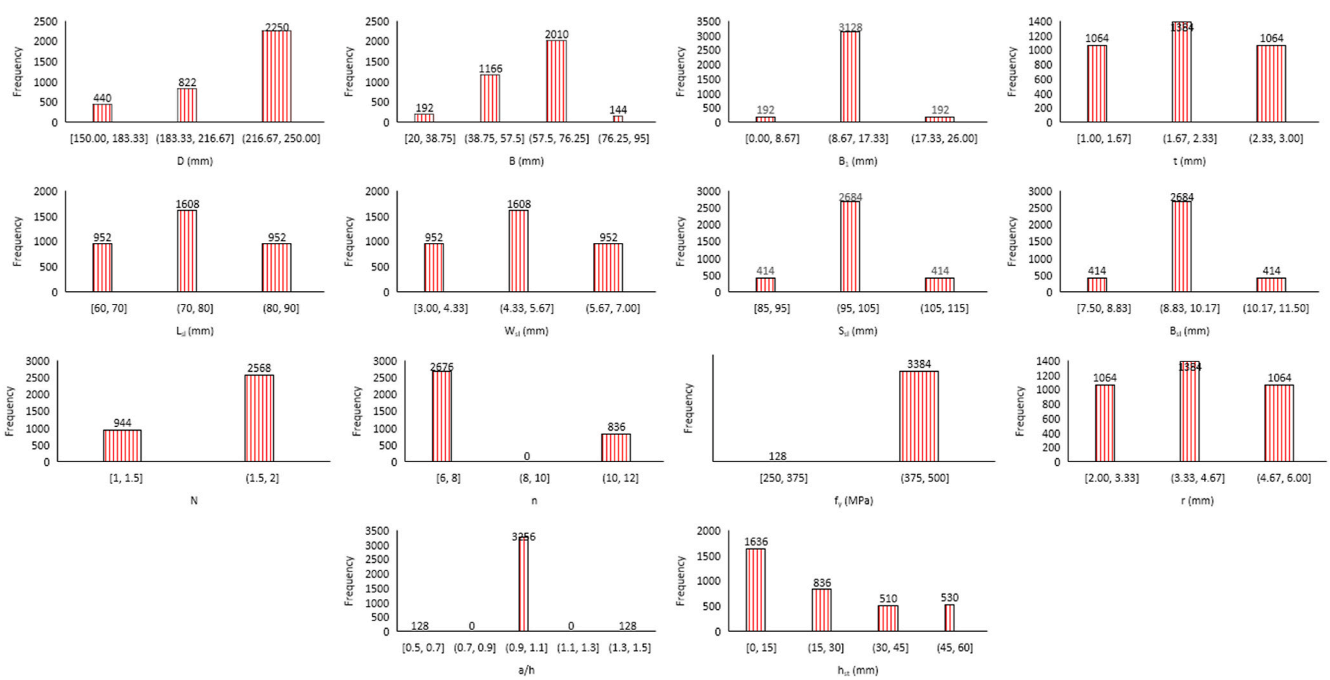

Fig. 3. Frequency of identified features of selected channels in the compiled database.

64]. Database 5 [55,67]. Database 6 [54]).[2]

1.2. *Database on concrete-filled steel tubular (CFST) columns (StructuresNet 1: Thai database)*

A comprehensive database that covers four types of concrete-filled steel tubular (CFST) columns was developed by Thai et al. [58,59]. This database collected 3103 notable tests on CFST columns collected from over 170 studies and falls under a regression database. The selected columns cover a range of configurations (short, slender, circular, square, and rectangular sections) that were tested under concentric and eccentric loading.

This database documents geometric features in terms of: 1) effective length, L_e, 2) tube thickness, t, 3) tube diameter, D, and material properties in terms of 4) yield stress, f_y, 5) compressive strength, f_c, of in-

[2] A historical, rough and traditional rule of thumb is that a minimum set can be 10 cases per predictor per van Smeden et al. [124]. However, this rule of thumb has been associated with some limitations (as noted by Riley et al. [125]) and hence a revised 23 cases per predictor criteria is proposed. As of this moment, a series of investigation continue to be carried out to better answer the question of the minimum size of databases needed for a ML analysis. In all cases, our databases satisfy both, the traditional criterion, and newly proposed criterion.

Table 3
Statistics from collected database.

	D (mm)	B (mm)	B_1 (mm)	t (mm)	L_{sl} (mm)	W_{sl} (mm)	S_{sl} (mm)	B_{sl} (mm)	N	n	f_y (MPa)	BC	r (mm)	a/h	h_{st} (mm)	V_{cr} (N)
Minimum	150.0	20.0	0.0	1.0	60.0	3.0	85.0	7.5	1.0	6.0	250.0	0.0	2.0	0.5	0.0	401.9
Maximum	250.0	95.0	26.0	3.0	90.0	7.0	115.0	11.5	2.0	12.0	500.0	-	6.0	1.5	60.0	309322.4
Average	225.8	57.8	13.0	2.0	75.0	5.0	100.0	9.5	1.7	8.0	490.9	-	4.0	1.0	19.6	32107.1
Standard deviation	35.4	13.5	4.3	0.8	11.0	1.5	7.3	1.0	0.4	2.4	46.9	-	1.6	0.1	22.0	37675.0
Skewness	-1.1	0.2	0.0	0.0	0.0	0.0	0.0	0.0	-1.0	0.8	-4.9	-	0.0	0.0	0.7	2.3
Parameter	D (mm)	B (mm)	B_1 (mm)	t (mm)	L_{sl} (mm)	W_{sl} (mm)	S_{sl} (mm)	B_{sl} (mm)	N	n	f_y (MPa)	BC	r (mm)	a/h	h_{st} (mm)	V_{cr} (N)
D (mm)	1.000															
B (mm)	0.648	1.000														
B_1 (mm)	0.000	0.000	1.000													
t (mm)	0.000	0.000	0.000	1.000												
L_{sl} (mm)	0.000	0.000	0.000	0.000	1.000											
W_{sl} (mm)	0.000	0.000	0.000	0.000	0.000	1.000										
S_{sl} (mm)	0.000	0.000	0.000	0.000	0.000	0.000	1.000									
B_{sl} (mm)	0.000	0.000	0.000	0.000	0.000	0.000	0.000	1.000								
N	0.218	0.131	0.000	0.000	0.000	0.000	0.000	0.000	1.000							
n	0.370	0.248	0.000	0.000	0.000	0.000	0.000	0.000	0.012	1.000						
f_y (MPa)	-0.133	-0.103	0.000	0.000	0.000	0.000	0.000	0.000	-0.008	-0.079	1.000					
BC	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	1.000				
r (mm)	0.000	0.000	0.000	1	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	1.000			
a/h	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	1.000		
h_{st} (mm)	0.311	0.200	0.000	0.000	0.000	0.000	0.000	0.000	0.539	0.131	-0.003	0.000	0.000	0.000	1.000	
V_{cr} (N)	-0.058	-0.0298	0.004	**0.726**	-0.306	-0.130	0.061	0.078	0.020	-0.118	-0.031	0.0981	**0.726**	-0.1474	0.012	1.000

filled concrete of CFST columns. Other features were also included such as modulus of concrete and steel, ultimate strength of steel section, load eccentrics etc. For sake of this study, 1245 circular CFST columns that were tested under concentric loading are examined herein. A graphical distribution of all features in this database is plotted in Fig. 2 and Table 2 summarizes the main attributes of the collected database in terms of material and geometric features.

Table 2 also shows that this database covers a practical range of CFST columns. For example, the minimum and maximum diameters of circular columns range between 44.45 mm and 1020.00 mm. The thickness range of the same columns varies between 0.52 and 16.54 mm. The range of yield strength of steel tubes and compressive strength of concrete filling is from 9.17 MPa to 193.30 MPa for concrete, and from 115.00 MPa to 853.00 MPa for steel. A sensitivity analysis was carried out to identify a correlation between all features compiled in this database. The outcome of this analysis shows that of all features, geometric features (D, and t) are of the highest importance. One should note that this sensitivity analysis is independent of the used ML model.

1.3. Database on shear strength of cold-formed steel (CFS) channels with slotted webs (StructuresNet 2: Degtyarev & Degtyareva database)

The second database falls under a regression database and compiles observations taken from 3512 FE simulations aimed to investigate the elastic shear buckling loads and the ultimate shear strengths of CFS channels with slotted webs as carried out by Degtyarev & Degtyareva [60–62], and recently published at [32]. In this database, the ultimate shear strengths of the CFS channels were determined from FE models that account for material and geometric nonlinearities, as well as initial geometric imperfections – thereby making this dataset rich with realistic information.

Overall, this database accounts for 15 features: 1) channel depth, D, 2) channel flange width, B, 3) channel flange stiffener length, B_1, 4) channel thickness, t, 5) length of slots, L_{sl}, 6) height of slots, W_{sl}, 7) spacing of slots in the longitudinal direction, S_{sl}, 8) spacing of slots in the transverse direction, B_{sl}, 9) number of perforated regions, N, 10) number of slot rows, n, 11) yield stress of steel, f_y, 12) type of boundary conditions: realistic and test setup (designated as 1 and 2, respectively), 13) inside bend radius, r, 14) the aspect ratio, a/h, and 15) height of the longitudinal stiffener, h_{st}, that can be used to predict the elastic shear buckling load, V_{cr}, and/or the ultimate shear strength, V_n (see Fig. 3). In this benchmark study, the elastic shear buckling load, V_{cr} will be solely used. The outcome of the sensitivity analysis is listed in Table 3 and shows strong correlation between channel thickness and inside bend radius, and elastic shear buckling load. It is worth noting that the inside bend radius was taken as 2t in all models, and hence the strong correlation (noting that channel thickness has a strong correlation with V_{cr}).

1.4. Database on compressive strength of high-performance concrete (StructuresNet 3: Yeh database)

The third database falls under a regression database and compiles 1030 data points taken from tests that determined compressive strength (f'_c) of high-performance concrete (HPC) as a function of: 1) Cement, C, 2) Blast Furnace Slag, B, 3) Fly Ash, F, 4) Water, W, 5) Superplasticizer, S, 6) Coarse Aggregate, CA, 7) Fine Aggregate, FA, and 8) Age, A. This database was published by Yeh [57] and has been extensively used in ML studies. Fig. 4 and Table 4 show the distribution of all features comprising this database. The outcome of the correlation matrix shows the highest positive correlation to be between cement and compressive strength, followed by age and compressive strength, and flay ash and superplasticizer. A negative correlation exists between water and superplasticizer and water and fine aggregates.

1.5. Database on low cycle fatigue (StructuresNet 4: Hawileh & Abdalla database)

The fourth database falls under a regression database and compiles real observation of around 60 data points taken from strain-controlled low-cycle fatigue tests that were carried out on steel reinforcing bars

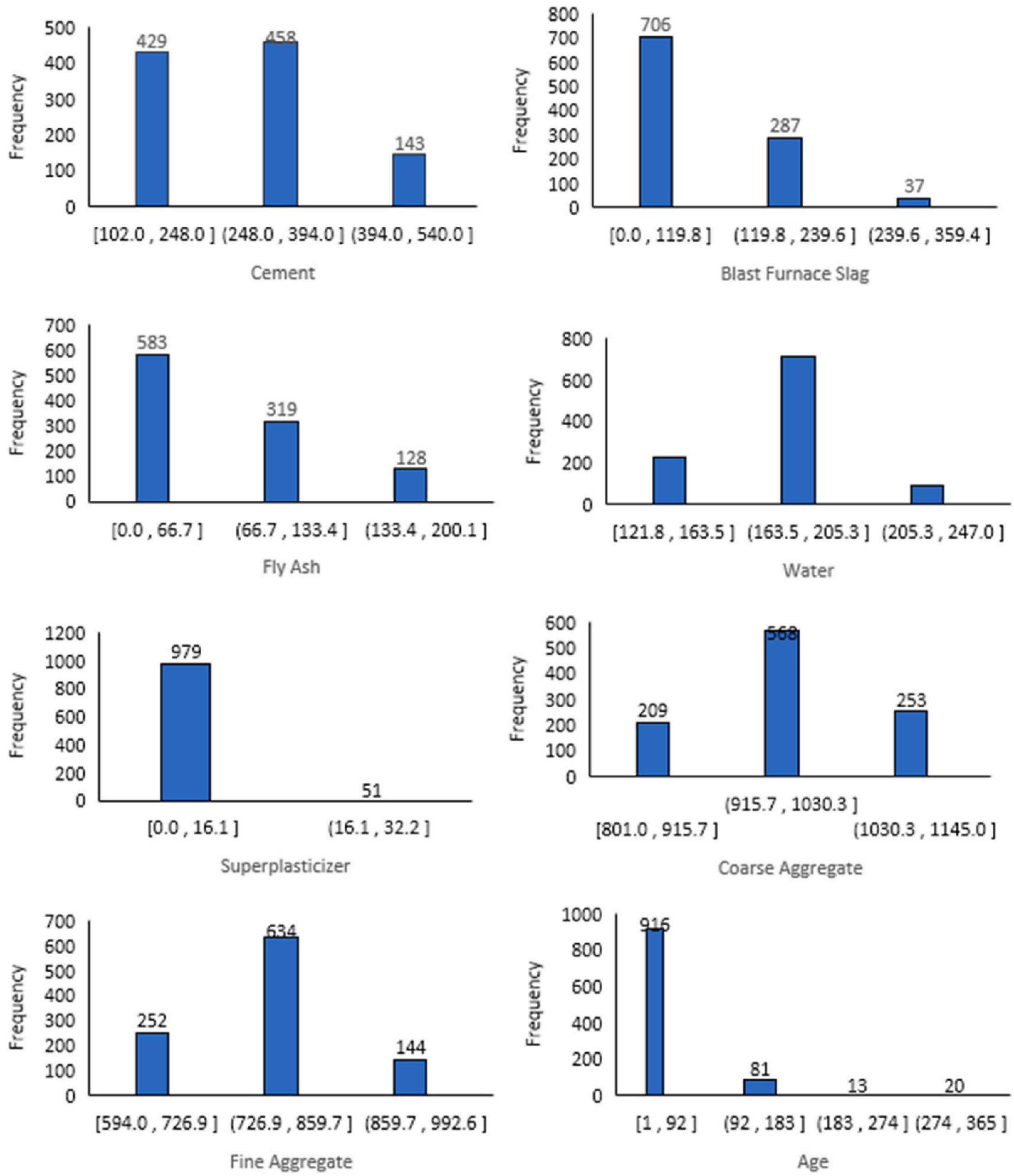

Fig. 4. Frequency of identified features of concrete mix designs in the compiled database.

under cyclic load with a frequency of 0.05 Hz. The tests determined the low-cycle fatigue life by measuring the number of reversals ($2N_f$) to fatigue failure of steel reinforcement bars of grade BS 460B and BS B500B. The database also contains generated data of energy dissipated in the first cycle, average cycles and total cycles of loading using numerical integration of area enclosed by the stress-strain hysteresis loops. These experimental and generated output parameters are function of: 1) Amplitudes of loading, A, and 2) strain ratio, R, for steel grade of BS460B, BS B500B. This database was generated by Abdalla et al. [63], Hawileh et al. [64] and has been used in ML studies to predict the fatigue life of steel reinforcing bars [48]. Fig. 5 and Table 5 show the distribution of all features comprising this database. The outcome of the correlation matrix shows the highest positive correlation to be between fatigue life (2Nf) and the total energy (W_{fT}), followed by R. A negative correlation exists between 2Nf and energy dissipated in the first cycle (W_1), energy dissipated in average cycles (W_A). Other low-cycle fatigue of steel reinforcing bars databases were generated as a result of experimental tests [65,66]. A sensitivity analysis was carried out to identify the correlation between all features compiled in this database. The outcome of this analysis shows that all features seem to be of high importance.

1.6. Database on shear strength of reinforced concrete (RC) beams and FRP-strengthened RC beams (StructuresNet 5: Abdalla et al. database)

A comprehensive database that covers two types of reinforced concrete beams strengthened in shear with steel stirrups [55] and with externally-bonded carbon fiber reinforced polymer sheets [67]. The two databases collected 290 notable test results from several experimental programs to measure the shear strength of RC beams. These regression databases were used to predict the shear strength of RC beams using different ML techniques.

The first database documents geometric features in terms of: 1) beam width, b, 2) beam effective depth, d, 3) span-to-depth ratio, a/d, 4) shear

Table 4
Statistics on collected database.

Section	Features	C	B	F	W	S	CA	FA	A	f'c
Compressive strength of HPC	Min	102.0	0.0	0.0	121.8	0.0	801.0	594.0	1.0	2.3
	Max	540.0	359.4	200.1	247.0	32.2	1145.0	992.6	365.0	82.6
	Average	281.2	73.9	54.2	181.6	6.2	972.9	773.6	45.7	35.8
	Standard deviation	104.5	86.3	64.0	21.4	6.0	77.8	80.2	63.2	16.7
	Median	0.5	0.8	0.5	0.1	0.9	0.0	-0.3	3.3	0.4
	Skewness	102.0	0.0	0.0	121.8	0.0	801.0	594.0	1.0	2.3

Parameter	*C*	*B*	*F*	*W*	*S*	*CA*	*FA*	*A*	*f'c*
C	1.000								
B	-0.275	1.000							
F	-0.397	-0.324	1.000						
W	-0.082	0.107	-0.257	1.000					
S	0.093	0.043	**0.377**	**-0.657**	1.000				
CA	-0.109	-0.284	-0.010	-0.182	-0.266	1.000			
FA	-0.223	-0.282	0.079	**-0.451**	0.223	-0.179	1.000		
A	0.082	-0.044	-0.154	0.278	-0.193	-0.003	-0.156	1.000	
f'c	**0.498**	0.135	-0.106	-0.290	0.366	-0.165	-0.167	**0.329**	1.000

reinforcement ratio, ρ_v, 5) concrete compressive strength, f_c, 6) flexural reinforcement ratio, ρ_w and 7) shear strength, V_n.. A graphical distribution of all features in this database is plotted in Fig. 6 and Table 6 summarizes the main attributes of the collected database in terms of material and geometric features. The second database documents geometric features in terms of: 1) beam width, b_w, 2) beam effective depth, d_{eff}, 3) beam span, L, 4) span-to-depth ratio, a/d, 5) concrete compressive strength, f_c, 6) steel yield strength of stirrup, f_y, 7) shear reinforcement per length, A_v/S, 8) steel yield strength of longitudinal reinforcement, f_y, 9) area of longitudinal reinforcement, A_{st}, 10) thickness of the fiber, t_f, 11) width of the fiber, B_f, 12) height of the fiber, H_f, 13) width of the fiber over the spacing ratio, W_f/S_f, 14) stress in the fiber, f_f, 15) modulus of elasticity of the fiber, E_f and 16) shear strength of the beam, V_f. A graphical distribution of all features in this database is plotted in Fig. 7 and Table 7 summarizes main attributes of the collected database in terms of material and geometric features. The outcome of the correlation matrix shows the highest positive correlation to be between the shear strength (V_n) and beam width (b) and between the shear strength (V_n) and beam depth (d). A negative correlation exists between shear strength (V_n) and span-to-depth ratio (a/d) and shear strength (V_n) and shear reinforcement ratio (ρ_v).

Sensitivity analyses were carried out to identify the correlation between all features compiled in these databases. The outcome of these analyses shows that all features, d_{eff}, A_v/S, f_f, H_f, and E_f are of the highest importance.

1.7. Database on fire resistance of reinforced concrete columns (FireNet: Naser & Kodur database)

The sixth database falls under a classification database and compiles real observations taken from over 140 fire resistance tests (including spalling phenomenon) and 169 FE simulations on reinforced concrete columns [68–81], and was compiled in Ref. [53]. This database contains information on binary incidents of fire-induced spalling (i.e., column spalled/does not spall), multi-class classification on fire rating of columns (e.g., in an hourly basis), and a regression-based data (i.e., fire resistance duration). The identified features in the database include: 1) column width, W, 2) steel reinforcement ratio, r, 3) column length, L, 4) concrete compressive strength, f_c, 5) steel yield strength, f_y, 6) restraint conditions, K (fixed-fixed, fixed-pinned, and pinned-pinned), 7) concrete cover to reinforcement, C, 8) eccentricity in applied loading in two axes (e_x and e_y), 9) the magnitude of applied loading, P, and 10) fire failure time, FR.

Fig. 8 and Table 8 present additional details into the range of each of the selected features. Similar to the other databases, this database also covers a practical range of columns often used in the construction industry. For example, all columns are of a square cross-section with a minimum and maximum width between 203 mm and 601 mm. The steel reinforcement ratio ranges between 0.9 and 4.4% and a length of 2.1–5.7 m. The range of yield strength of steel reinforcement and compressive strength of concrete filling is from 354.0 MPa to 591.0 MPa, and from 24.0 MPa to 138.0 MPa, respectively. The used concrete cover spans 25.0–64.0 mm and eccentric between 0 and 150 mm finally, the applied loading ranges between 0.0 and 5373.0 kN.

A sensitivity analysis was carried out to identify the correlation between all features compiled in this database. The outcome of this analysis shows a primarily weak correlation between the features and fire resistance except for the case of boundary conditions which displayed a medium negative correlation, a positive correlation attained by the concrete cover. In addition, a few interesting observations can also be made from this correlation analysis. For example, a high positive correlation appears to be between compressive strength and applied loading, and a medium correlation arises between column width and loading level.

1.8. Selected machine learning algorithms

As mentioned earlier, the primary goal of this work is to benchmark commonly used ML algorithms (in their default settings) against structural and fire engineering problems. In this pursuit, a review of recent works [82–84] identified the following six algorithms as the most commonly used algorithms in structural and fire engineering domains: Decision Trees (DT), Random Forest (RF), Extreme Gradient Boosted Trees (ExGBT), Light Gradient Boosted Trees (LGBT), TensorFlow Deep

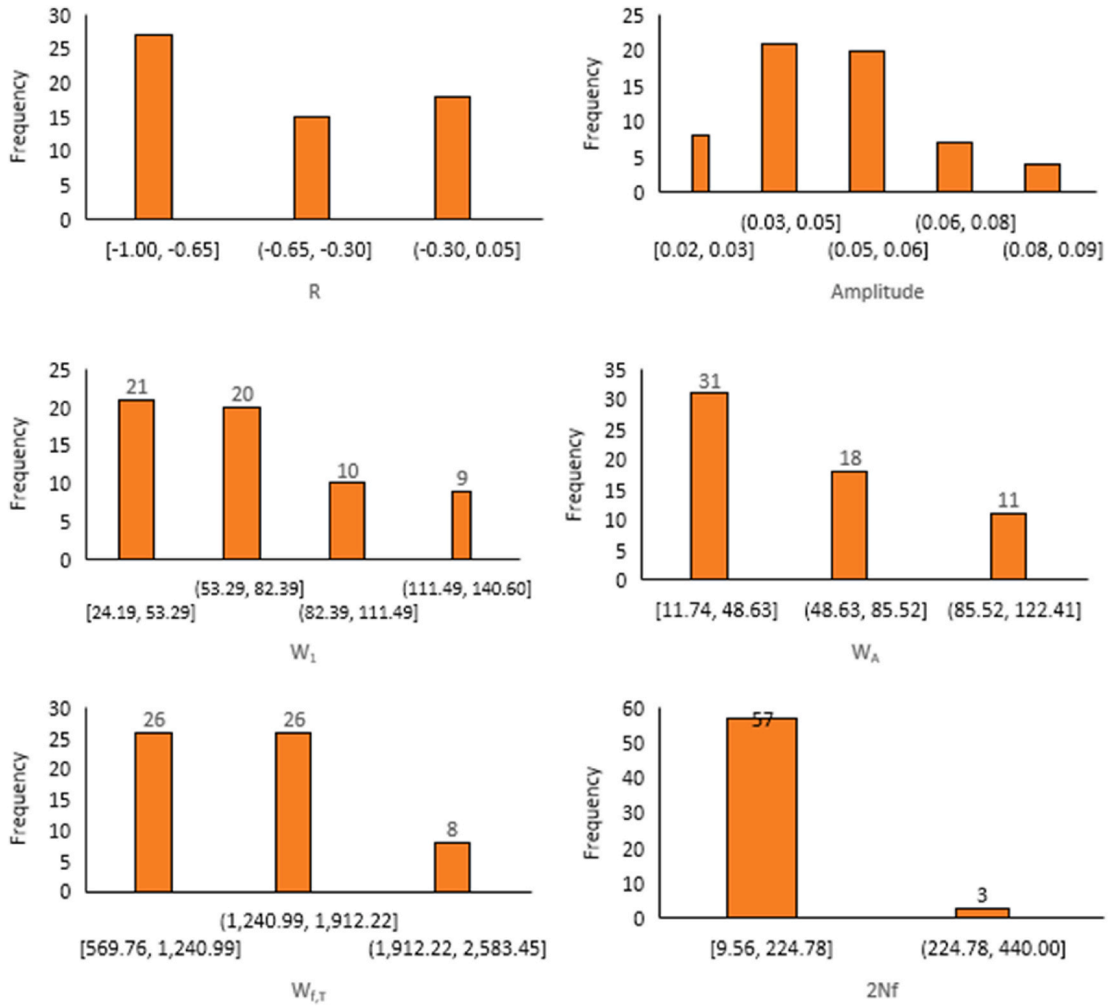

Fig. 5. Frequency of identified features of concrete mix designs in the compiled database.

Table 5
Statistics on collected database.

Section	Features	A	R	W_1	W_A	$W_{f,T}$	$2N_f$
Compressive strength of HPC	Min	0.0	-1.0	24.2	11.7	569.8	9.6
	Max	0.1	0.0	140.6	122.4	2583.4	440.0
	Average	0.0	-0.6	72.9	55.2	1324.9	75.9
	Standard deviation	0.0	-0.6	30.4	27.8	482.7	75.7
	Median	0.0	0.4	0.6	0.6	0.6	2.6
	Skewness	0.4	0.3	24.2	11.7	569.8	9.6
Parameter		A	R	W_1	W_A	$W_{f,T}$	$2N_f$
A		1.000					
R		-0.511	1.000				
W_{p1}		0.964	-0.344	1.000			
$\Delta W_{p,avg}$		0.964	-0.561	0.958	1.000		
W_{fT}		-0.826	0.691	-0.763	-0.869	1.000	
$2N_f$		-0.699	0.566	-0.638	-0.716	0.863	1.000

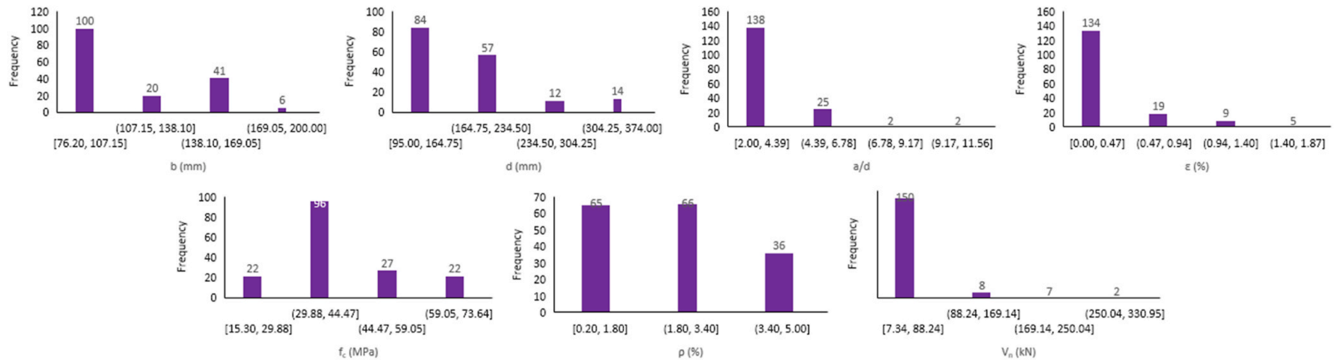

Fig. 6. Frequency of identified features of selected RC columns in the compiled database.

Table 6
Statistics on collected database for RC beams.

Section	Features	b (mm)	d (mm)	a/d	ρ_v	f_c	ρ	V_n
Shear strength of RC beams	Min	76.2	95.0	2.0	0.0	15.3	0.2	7.3
	Max	200.0	374.0	11.6	1.9	73.6	5.0	330.9
	Average	111.5	176.6	3.5	0.3	41.3	2.2	47.5
	Standard deviation	34.3	72.4	1.4	0.4	13.3	1.2	49.3
	Median	0.6	1.3	2.2	2.0	0.9	0.6	3.1
	Skewness	76.2	95.0	2.0	0.0	15.3	0.2	7.3
Parameter		b	d	a/d	ρ_v	f_c	ρ_w	V_n
b		1.000						
d		**0.634**	1.000					
a/d		-0.165	-0.223	1.000				
ρ_v		-0.501	-0.401	-0.070	1.000			
f_c		0.016	-0.080	0.078	0.040	1.000		
ρ_w		-0.309	-0.171	0.111	0.295	0.254	1.000	
V_n		**0.527**	**0.506**	-0.326	-0.191	0.181	0.353	1.000

Learning (TFDL), and Keras Deep Residual Neural Network (KDP), and these are briefly discussed herein. Most of these algorithms can be used in regression, and classification problems which are expected to cover the majority of structural and fire engineering problems.

1.9. Decision Trees (DT)

The DT algorithm has the capability to generate a schematic representation of all possible decisions and consequences, which can be visualized by dividing the database into branch-like arrangements [85]. In general, a DT is generated and starts at a root node and then grows into tree-like components (i.e., leaves etc.). The developed algorithm was obtained in its default setting from Scikit platform [86]. This algorithm has a maximum depth of “none”, minimum leaf size and maximum size for split equals to 1 and “none”, respectively [87,88]. This DT algorithm utilized Gini impurity to facilitate the quality of a split and processing of datapoints. For example, for a node t, the Gini index g (t) is defined as [89]:

$$g(t) = \sum_{j \neq i} p(j|t)p(i|t) \tag{1}$$

where i and j are target field categories, and p is for probability.

$$p(j,t) = \frac{p(j,t)}{p(t)}; p(j,t) = \frac{\pi(j)N_j(t)}{N_j}; \text{and} \cdot p(t) = \sum_j p(j,t) \tag{2}$$

1.10. Random Forest (RF)

This algorithm integrates multiple DTs via ensemble learning to form a more powerful prediction model; hence, a forest of trees [90]. In RF, all individual DTs reach a predictive outcome. Then, this outcome is processed depending on the type of problem (i.e., regression vs. classification). For a regression problem, the average result of all trees is calculated to arrive at a final outcome. On the other hand, in a classification problem, the majority voting method is used to consolidate the final outcome. A typical formulation of RF is presented herein:

$$Y = \frac{1}{J}\sum_{j=1}^{J} C_{j,full} + \sum_{k=1}^{K}\left(\frac{1}{J}\sum_{j=1}^{J} contribution_j(x,k)\right) \tag{3}$$

where, J is the number of trees in the forest, k represents a feature in the observation, K is the total number of features, C_{full} is the average of the entire dataset (initial node). The used algorithm can be found herein [91] and has the following default settings; number of trees = 500, Gini impurity to facilitate quality of a split, a maximum depth of “none”, minimum leaf size, and maximum size for split equals to 5 and “none”, respectively.

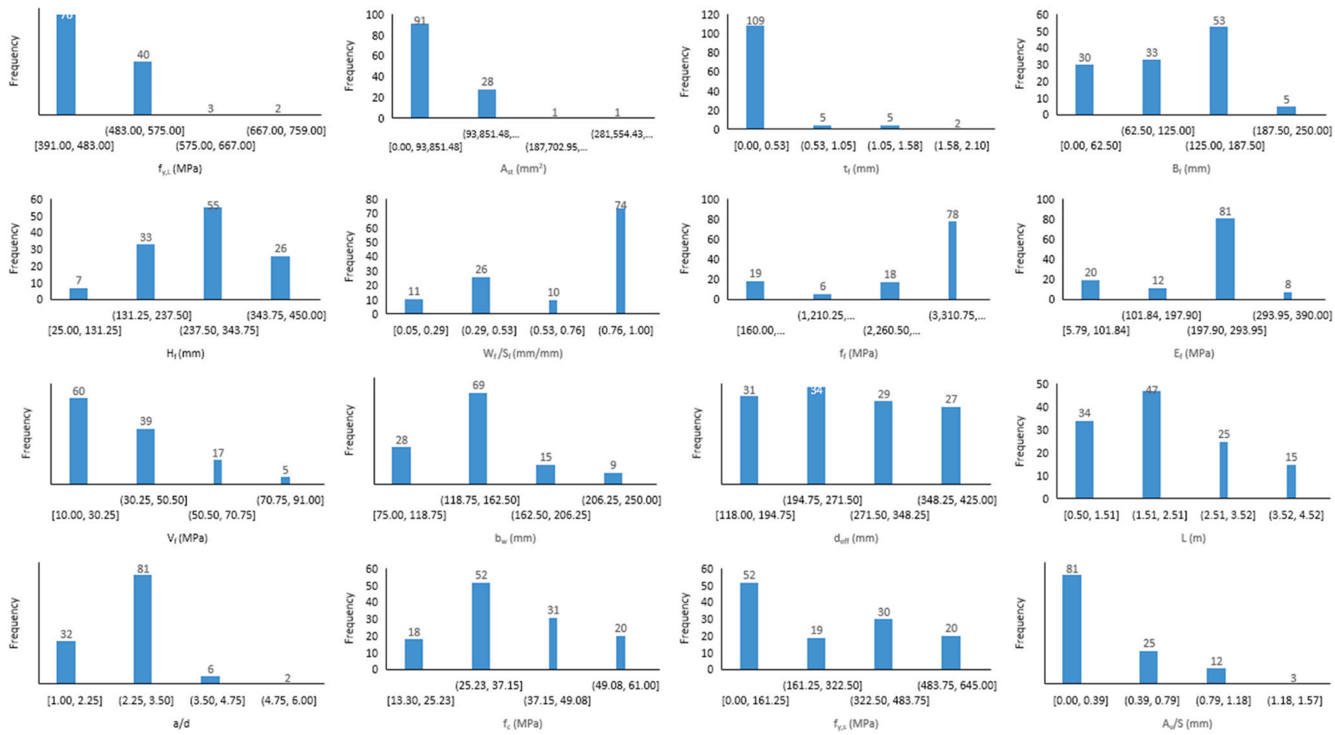

Fig. 7. Frequency of identified features of selected RC columns in the compiled database.

Table 7
Statistics on collected database for FRP-strengthened RC beams.

	b_w (mm)	d_{eff} (mm)	L (m)	a/d	f_c (MPa)	$f_{y,s}$ (MPa)	A_v/S (mm²/mm)	$f_{y,L}$ (MPa)	A_{st} (mm²)	t_f (mm)	B_f (mm)	H_f (mm)	W_f/S_f	f_f (MPa)	E_f (MPa)	V_f (kN)
Minimum	75.0	118.0	0.5	1.0	13.3	0.0	0.0	391.0	0.0	0.0	0.0	25.0	0.1	160.0	5.8	10.0
Maximum	250.0	425.0	4.5	6.0	61.0	645.0	1.6	759.0	375405.9	2.1	250.0	450.0	1.0	4361.0	390.0	91.0
Average	145.5	263.5	2.2	2.7	35.9	236.4	0.3	473.3	45958.4	0.3	104.7	262.9	0.8	2940.9	198.1	33.4
Standard deviation	41.9	84.7	1.0	0.8	10.3	222.8	0.4	65.6	66745.8	0.4	68.6	89.3	0.3	1223.2	92.8	17.9
Skewness	0.8	-0.2	0.8	1.3	0.4	0.1	1.4	1.5	1.7	2.8	-0.4	-0.6	-0.8	-1.4	-0.6	1.0
Parameter	b_w	d_{eff}	L	a/d	f_c	$f_{y,s}$	A_v/S	$f_{y,L}$	A_{st}	t_f	B_f	H_f	W_f/S_f	f_f	E_f	V_f
b_w	1.000															
d_{eff}	0.241	1.000														
L	-0.207	**0.538**	1.000													
a/d	-0.150	-0.131	0.108	1.000												
f_c	0.050	-0.059	-0.215	-0.076	1.000											
$f_{y,s}$	-0.174	0.141	0.108	-0.032	-0.025	1.000										
A_v/S	-0.239	-0.031	0.069	-0.006	-0.034	0.657	1.000									
$f_{y,L}$	-0.056	0.185	-0.089	0.117	0.127	0.005	-0.107	1.000								
A_{st}	-0.196	-0.011	0.154	0.114	-0.076	0.723	**0.882**	-0.031	1.000							
t_f	-0.070	-0.301	-0.135	-0.047	0.151	0.041	0.075	-0.072	0.051	1.000						
B_f	0.107	0.148	0.106	0.178	0.119	0.100	0.101	0.122	0.134	-0.186	1.000					
H_f	0.258	**0.857**	0.357	-0.185	0.161	0.138	-0.149	0.199	-0.107	-0.376	0.108	1.000				
W_f/S_f	-0.244	-0.189	0.343	0.218	-0.267	0.245	0.194	-0.112	0.299	-0.087	0.086	-0.290	1.000			
f_f	0.179	0.351	0.281	-0.002	-0.187	-0.106	-0.216	0.032	-0.137	-0.521	0.168	0.490	-0.112	1.000		
E_f	0.239	0.359	0.206	0.017	-0.161	-0.095	-0.257	0.174	-0.144	-0.471	0.221	0.434	-0.132	**0.819**	1.000	
V_f	0.311	0.224	-0.023	-0.039	0.146	-0.155	-0.061	0.178	-0.152	-0.037	0.297	0.340	-0.260	0.189	0.128	1.000

1.11. *Extreme Gradient Boosted Trees (ExGBT)*

The ExGBT algorithm is an improved form of the Adaboost algorithm [92]. ExGBT re-samples the collected data points into a tree-like format, where each tree sees a bootstrap sample of the database in each iteration. ExGBT fits each successive tree to previous residual errors obtained from previous trees; thereby focusing each iteration on the observations that are most difficult to predict, which becomes a good practice for the algorithm to yield high prediction accuracy [93]. The code of the used ExGBT can be found online at [94,95]. This algorithm incorporates default settings of a learning rate of 0.1, maximum tree depth of 3, subsample feature of 1.0, and 100 for the number of boosting stages.

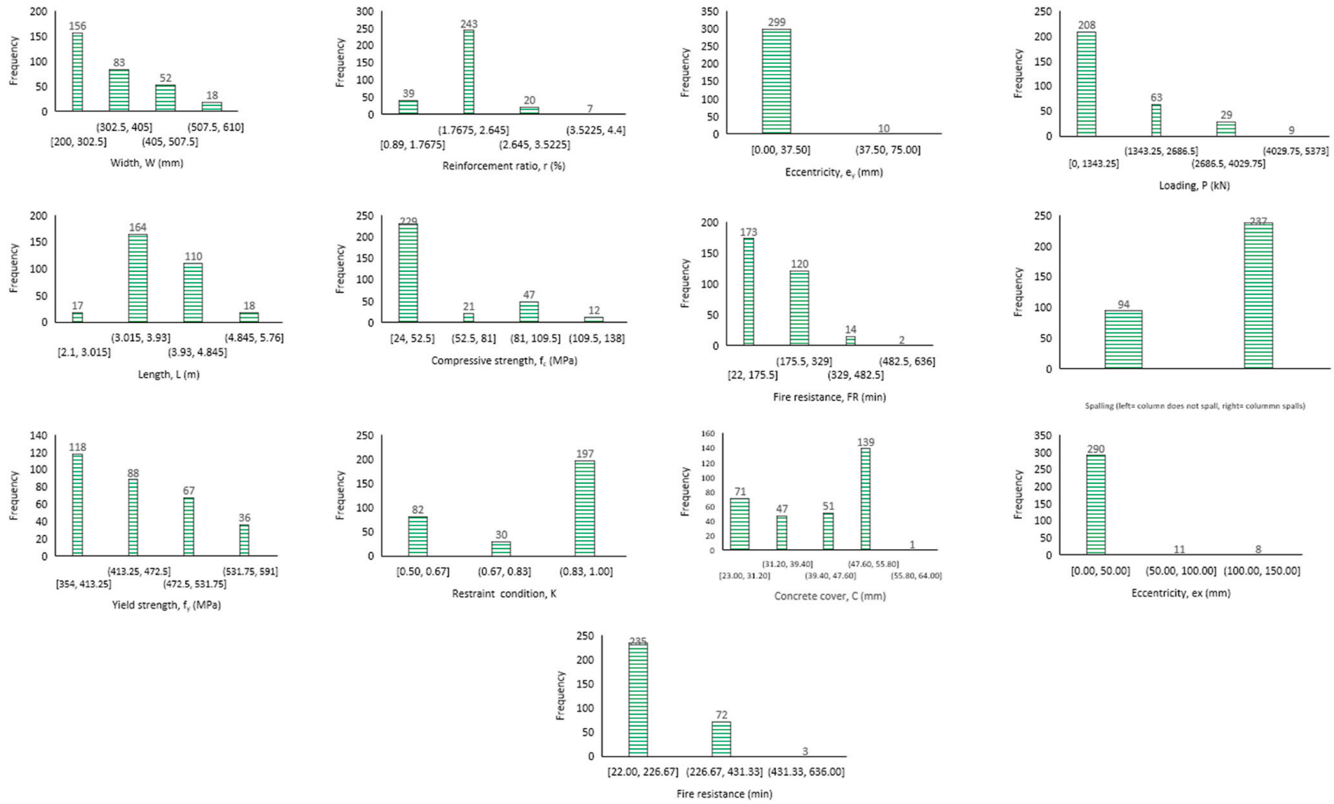

Fig. 8. Frequency of identified features of selected RC columns in the compiled database.

1.12. Light Gradient Boosted Trees (LGBT)

Light gradient boosted trees is a light algorithm that requires little processing and is a generalization of their parent algorithm (Adaboost) [96]. This algorithm is very much similar to the RF algorithm with the main exception is that it does not fit the trees in parallel but rather it fits the trees in a successive manner and fits the residual errors from all the previous trees combined. This is advantageous, as the model focuses each iteration on the examples that are most difficult to predict. The used algorithm can be found at [97] with the following default settings: learning rate = 0.02, maximum depth = "none", number of boosting stages = 500 etc.

1.13. TensorFlow Deep Learning (TFDL)

This is a neural network-based model that uses Deep Learning as the primary method of analysis. A TFDL algorithm mimics the topology of the brain and comprises of a minimum of three layers. The first layer receives the database and forward it to the second set of layer(s). These layers use a nonlinear activation function which enables the algorithm of generating an approximation form that permits gradient-based optimization (see Eqs. (4) and (5)). The used algorithm in its default settings (neurons in each layer = 55, number of training examples = 128, optimizer = Adam, learning rate = 0.001, early stopping window = 10 etc.) can be found at [98].

$$net_j = \sum_{i=1}^{n} In_i w_{ij} + b_j \tag{4}$$

$$Y = f\left(net_j\right) \tag{5}$$

where, In_i and b_j are the ith input signal and the bias value of jth neuron, respectively, w_{ij} is the connecting weight between ith input signal and jth neuron, and f is an activation function such as Relu.

1.14. Keras Deep Residual Neural Network (KDP)

Keras is a high-level library for developing neural networks [99]. In a residual network, a direct connection exists linking data points to the outputs. Such a connection smoothens out the loss function and enables better optimization of the network. In the used KDP, default settings of a learning rate of 0.03 was used, along with a Prelu activation function, two layers containing 512 neurons. KDP can be readily found at [100].

1.15. Selected performance metrics

The adequacy of ML models in predicting engineering phenomena is often established through a comparison against performance metrics. Such metrics are defined as logical and/or mathematical constructs intended to measure the closeness of test measurements to that predicted by a ML model [101–103]. There exists a large body of literature covering a variety of metrics [104,105]. In this work, a focus is to provide the reader with a set of metrics that can be suitable for the majority of engineering applications. These metrics cover two domains, regression, and classification, as listed below.

In this study, four regression metrics and four classification metrics are presented (see Table 9). These metrics are commonly used in structural and fire engineering literature [32,106–108]. On the regression front, the metrics include; Mean Absolute Error (MAE), Mean Absolute Percentage Error (MAPE), Root Mean Squared Error (RMSE), and Coefficient of Determination (R^2). Both MAE and MAPE measure the difference between continuous variables in terms of the same scale or as a percentage, respectively. MAPE tends to suffer when applied to predictions with zero values. On the other hand, the RMSE describes the errors in a scale-independent fashion, where lower values indicating favorable prediction capability. One should note that RMSE is sensitive to outliers and to the fraction of the data used. R^2 is also used herein, and this metric is the square of the coefficient of correlation (r); which measures the degree of association between observed and predicted

Table 8
Statistics on collected database.

		W (mm)	*r (%)*	*L (m)*	*f_c (MPa)*	*f_y (MPa)*	*K*	*C (mm)*	*e_x (mm)*	*e_y (mm)*	*P (kN)*	*FR (min)*
Fire resistance analysis	Minimum	200.0	0.9	2.1	24.0	354.0	-	23.0	0.0	0.0	0.0	22.0
	Maximum	610.0	4.4	5.8	138.0	591.0	-	64.0	150.0	75.0	5373.0	636.0
	Average	324.3	2.1	4.0	49.3	449.4	-	40.2	15.8	2.0	1204.8	161.0
	Standard deviation	99.2	0.6	0.7	28.1	60.1	-	8.7	29.7	10.1	1031.6	97.6
	Skewness	1.9	0.6	0.3	1.4	0.7	-	-0.6	2.9	5.3	1.7	0.9
Spalling Analysis	Minimum	152.0	0.7	-	16.0	-	-	25.0	-	-	0.0	-
	Maximum	514.0	4.9	-	126.5	-	-	64.0	-	-	5373.0	-
	Average	325.3	2.5	-	54.3	-	-	37.6	-	-	1556.9	-
	Standard deviation	69.4	0.8	-	27.9	-	-	4.4	-	-	1109.1	-
	Skewness	0.7	1.0	-	1.1	-	-	0.6	-	-	1.4	-

Parameter	*W*	*r*	*L*	*f_c*	*f_y*	*K*	*C*	*e_x*	*e_y*	*P*	*FR*
W	1.000										
r	-0.120	1.000									
L	-0.172	0.256	1.000								
f_c	0.244	0.055	-0.110	1.000							
f_y	-0.250	-0.346	-0.078	-0.478	1.000						
K	0.022	-0.283	0.326	-0.079	0.169	1.000					
C	0.319	0.312	-0.224	0.279	**-0.641**	-0.362	1.000				
e_x	-0.088	0.046	0.356	-0.230	0.154	0.278	-0.257	1.000			
e_y	0.156	-0.047	-0.001	-0.136	-0.144	0.145	0.160	0.181	1.000		
P	**0.670**	0.121	-0.206	**0.559**	-0.384	-0.214	0.283	-0.213	0.035	1.000	
FR	0.381	0.081	-0.440	0.221	-0.277	**-0.604**	**0.558**	-0.370	-0.043	0.365	1.000

values with r closer to +1 indicates a positive and perfect linear relationship. Higher and positive values of R^2 indicate strong and positive prediction capability.

On the classification front, four metrics are also presented, including; Accuracy (ACC), Balanced accuracy (BACC), Area under the ROC curve (AUC), and Log Loss Error (LLE). Unlike their regression counterparts, these metrics are used to evaluate the prediction capability of a ML algorithm in terms of categorial outputs of binary (i.e., spalling occurs/spalling does not occur), or multi-output classes (e.g., 60 min fire rating/120 min fire rating/180 min fire ratings etc.). For instance, ACC evaluates the ratio of the number of correct predictions to the total number of samples used in the analysis, and as such, assumes equal penalty for errors. BACC is useful for databases with imbalanced data and multi-classes; where on class has relatively larger occurrences than other classes. This metric is a normalized version of ACC and calculates accuracy on a per-class basis, then averaging the per-class accuracies. The AUC measures the area under the Receiver Operating Characteristic (ROC) curve; with a higher area (close to 1.0) reflecting an accurate prediction capability. The Log Loss error measures the performance of a classification model whose output is a probability value between 0 and 1; thereby, a prefect model would have a log loss of 0.0.

The above discussion shows that while all selected metrics have been used in engineering and computer science benchmarking, they still tend to have some limitations, and hence it is advisable to use a collection of metrics when evaluating ML algorithms in problems in our domains. Comparing model performance across multi-metrics is seen of merit (as opposed a sole metric) since this practice brings in a whole view to the performance of ML models. The ML user is also advised as to apply due diligence in selecting proper metrics for the problem on hand. For example, the use of regression-based metrics may not yield a proper exploration of classification-based problems and vice versa. The above discussion covers key ideas behind some of the most commonly used metrics and a more in-depth discussion on the provided metrics, along with others such as Mean Squared Error (MSE), Reference index (RI), Confusion Matrix (CM), and Cohen's kappa (CK) etc., can be found elsewhere [104,105].

A: actual measurements, P: predictions, n: number of data points, E = A-P, P (denotes number of real positives), N (denotes number of real negatives), TP (denotes true positives), TN (denotes true negatives), FP (denotes false positives), and FN (denotes false negatives).

1.16. Benchmarking of selected algorithms

This section details the benchmarking of all selected algorithms at the six compiled databases. As mentioned above, all algorithms were used given their default settings to allow a raw evaluation of their performance against structural and fire engineering data/problems. Table 10 lists the outcome of the carried-out benchmark analysis in terms of performance metrics under training, validation and testing regimes. All analyses adopted a five-fold cross-validation procedure. The best performing algorithms are shown in bold in Table 10. For simplicity and to negate the notion of chasing accuracy as it is beyond the objective

Table 9
List of common performance metrics.

Problem	Name	Metric	Remarks
Regression	Mean Absolute Error (MAE)	Measures the difference between two continuous variables, as $MAE = \frac{\sum_{i=1}^{n}\lvert E_i\rvert}{n}$	• Uses a similar scale to input data [109]. • Can be used to compare data points of different scales.
	Mean Absolute Percentage Error (MAPE)	Measures the extent of error in percentage terms, as $MAPE = \frac{100}{n}\sum_{i=1}^{n}\lvert E_i\rvert/\lvert A_i\rvert$	• Cannot be used if there are actual zero values. • Non-symmetrical (adversely affected if a predicted value is larger or smaller than the corresponding actual value) [110].
	Root Mean Squared Error (RMSE)	Measures the square root of the average of squared errors $RMSE = \sqrt{\frac{\sum_{i=1}^{n}{E_i}^2}{n}}$	• Scale dependent. • A lower value for RMSE is favorable. • Sensitive to outliers. • Highly dependent on fraction of data used (low reliability) [111].
	Coefficient of Determination (R^2)	Measures the goodness of fit of a mode $R^2 = 1 - \sum_{i=1}^{n}(P_i - A_i)^2 / \sum_{i=1}^{n}(A_i - A_{mean})^2$	• R^2 values close to 1.0 indicate strong correlation. • The square of correlation.
Classification	Accuracy (ACC)	Evaluates the ratio of number of correct predictions to the total number of samples. $ACC = \frac{TP+TN}{P+N} = \frac{TP+TN}{TP+TN+FP+FN}$	• Presents performance at a single class threshold only. • Assumes equal cost for errors [39].
	Balanced accuracy (BACC)	$BACC = \frac{1}{M}\sum_{m=1}^{M}\frac{r_m}{n_m}$ where, M = number of classes, n_m = data size belongs to class m, r_m = number of data accurately predicted belonging to class m.	• Balanced accuracy is a metric that one can use when evaluating how good a binary or multi-classifier is. • Useful for imbalanced and multi-classification databases.
	Area under the ROC curve (AUC)	Measures the two-dimensional area underneath the entire ROC curve. $AUC = \sum_{i=1}^{N-1}\frac{1}{2}(FP_{i+1} - FP_i)(TP_{i+1} - TP_i)$	• Not dependent on a single class threshold. • Associated with increased training times.
	Log Loss Error (LLE)	Measures the where the prediction input is a probability value. $LLE = -\sum_{c=1}^{M}A_i logP,$ where, M: number of classes, c: class label, y: binary indicator (0 or 1) if c is the correct classification for a given observation.	• Penalizes for being too confident in wrong prediction. • Has probability between zero and 1. • A log loss of zero indicates a perfect model.

of this work, all results were rounded for two decimal places.

As one can see and as expected, not a single algorithm was found to be dominant in all of the carried-out examinations, nor in all three testing regimes. This highlights the need for adopting multiple algorithm search, and multiple performance metrics in a given ML analysis. One should still note that of all algorithms, ExGBT and LGBT seem to outperform all other algorithms, with ExGBT leading. For instance, ExGBT managed to score the best metrics in Database 1, 3, 4 and 5, while LGBT performed comfortably well in Database 2. On the contrary, the DT algorithm performed the poorest of all algorithms in the majority of the tested databases, followed by RF and TFDL.

A note to remember is that the outcome of this analysis only reflects upon the selected six algorithms and does not imply that other algorithms may not perform better than those used herein. The same also goes for the selected performance metrics. As mentioned earlier, the notion of this work is not to start an "accuracy chase", especially since, as the conducted analysis shows, accurateness is not only a complex metric to realize and achieve but is also subjective and requires a deep dive into multi-metrics and domains.

1.17. Future directions

This benchmarking study alludes to the notion that intentional, or unintentional cherry-picking in a given ML analysis is likely; given that not all users are familiar with all different ML modeling techniques, nor there is a requirement to attempt to try to examine all possible ML techniques. In the majority of scenarios, and rightly so, a user may in fact favor algorithms that s/he familiar with over others. As such, properly benchmarking ML model is needed now noting how the use of ML into our domains is expected to continue to rise and hence works targeting benchmarking will set the foundation towards a reliable and safe integration of this new technology. Early attempts in this area will help overcome existing issues related to standardization and validation of FE models, among other methods [7,112]. In addition, future attempts will continue to overcome some of the current limitations of ML especially those with regard to limited number of data points, selection of tuning parameters, different coding languages, need for improved inference performance etc. [82,83].

One could argue that modeling (in general) may not be suited for technicians, like testing standards for materials, and hence a FE model should only be implemented by trained engineers. However, a trained engineer is also required to follow/adhere to a procedure. To ensure compatibility, such a procedure is to be unified, generally accepted, or standardized for repeatability and transparency. We, then, argue that the modeling, whether is to be deployed by technicians or engineers, also needs to follow a commonly accepted procedure. In a way, a move towards a unified procedure will facilitate both inclusivity and diversity into our domains. Such a procedure can start by benchmarking commonly used ML models as it is customary in the computer science domain [113–115]. The message of this work also aligns with that proposed by other researchers that focused on unifying FE modeling procedures [24–27].

This paper focuses on benchmarking commonly available ML algorithms against structural and fire engineering phenomena by analyzing six notable databases that have been properly documented and examined in the open literature. As such, the primary goal of this paper is not to chase high accuracy scores but rather establishes a benchmark for the following ML models DT, RF, ExGBT, LGBT, TFDL, and KDP against structural and fire engineering problems. Similar to other works [116–119], we hope that StructuresNet and FireNet can accelerate the use of ML into the structural engineering and fire engineering domains. In the future to come, new works are encouraged to cross-check their ML models' predictive power against findings from this benchmarking study. We expect finetuned upcoming ML models to achieve improved performance than what we displayed herein. Interested works are also invited to continue progress in this area as a mean to capitalize upon the attractiveness of ML.

There are three sub-domains to benchmarking: 1) number and types

Table 10
Outcome of benchmark analysis.

	Metric	DT			RF			ExGBT			LGBT			TFDL			KDP		
		Training	Validation	Testing	Training	Validation	Testing	Training	Validation	Testing	Training	Validation	Testing	Training	Validation	Testing	Training	Validation	Testing
Database 1 (Regression)	R^2	0.96	0.96	0.87	0.82	0.94	0.83	*0.99*	*0.99*	0.92	0.91	0.96	*0.99*	0.57	0.56	0.39	0.86	0.93	0.88
	MAE	291.48	280.57	437.52	331.19	331.41	425.11	*115.18*	*132.13*	235.20	197.91	161.74	*116.06*	835.06	926.84	881.29	369.90	298.70	366.28
	MAPE	12.69	13.27	13.31	12.99	15.77	12.77	*6.29*	*6.95*	6.73	7.00	6.72	*5.65*	29.58	36.52	26.26	11.99	11.37	10.50
	RMSE	707.966	691.71	1665.5	1005.55	881.29	1929.42	*260.81*	*309.48*	1293.29	1091.33	646.86	*254.28*	2433.55	2493.91	3654.07	1374.04	928.01	1608.60
Database 2 (Regression)	R^2	0.95	0.95	0.94	0.92	0.93	0.94	0.99	0.99	*0.99*	*0.99*	*0.91*	*0.99*	0.33	0.73	0.81	0.99	0.93	*0.99*
	MAE	3966.50	3964.98	3699.48	5399.59	4838.66	4321.92	1844.14	1695.84	1625.74	*1528.76*	*1519.08*	*1552.55*	8326.67	9069.17	7703.48	2242.65	2200.97	1977.25
	MAPE	12.09	11.78	11.50	20.27	19.16	18.67	5.25	5.0	4.87	*4.47*	*4.60*	*4.48*	26.08	28.38	26.75	6.55	6.63	6.57
	RMSE	8688.51	8803.70	8861.78	11547.00	9965.00	922.20	4165.88	3727.08	4012.00	*3492.27*	*3489.72*	*3756.76*	16313.00	17212.00	16263.00	4799.37	4770.15	4235.65
Database 3 (Regression)	R^2	0.88	0.83	0.82	0.89	0.88	0.89	*0.93*	*0.92*	*0.94*	0.69	0.64	0.67	0.48	0.46	0.39	0.88	0.89	0.90
	MAE	4.44	5.05	4.70	4.18	4.18	4.03	*3.10*	*3.11*	*2.66*	7.56	8.056	7.65	10.57	10.73	10.82	3.85	3.77	3.43
	MAPE	15.83	17.85	15.01	15.27	14.94	13.64	*10.90*	*10.59*	*8.91*	35.34	34.42	29.36	29.93	31.97	30.86	12.69	12.14	11.46
	RMSE	5.97	6.81	6.67	5.46	5.70	5.25	*4.23*	*4.59*	*3.7*	9.42	10.08	9.08	12.37	12.39	12.43	5.88	5.64	4.98
Database4 (Regression)	R^2	-1.00	0.42	0.69	0.56	0.72	0.88	*0.59*	*0.78*	*0.90*	-1.00	0.22	0.62	-3.02	-0.54	0.58	-0.96	0.27	0.71
	MAE	24.50	20.16	24.28	15.17	18.27	17.84	*11.05*	*18.15*	*14.02*	28.01	26.49	31.21	54.31	42.92	36.78	25.60	27.40	22.88
	MAPE	41.93	24.26	4099	48.65	27.32	35.19	*26.29*	*22.05*	*28.04*	237.59	97.59	133.92	74.33	37.63	49.84	54.81	37.76	38.44
	RMSE	44.29	37.29	37.33	20.47	34.78	22.87	*19.63*	*30.10*	*21.32*	44.67	48.55	41.56	68.29	63.18	43.76	43.15	46.87	36.25
Database5 (Regression)	R^2	0.49	0.72	0.47	0.75	0.80	0.35	0.96	0.95	0.91	*0.98*	*0.96*	0.80	0.81	0.80	0.54	0.98	0.95	*0.94*
	MAE	17.39	13.90	12.72	13.58	12.57	13.88	5.75	5.70	5.71	*3.13*	*5.11*	7.18	12.18	13.63	12.76	3.46	5.13	*4.44*
	MAPE	66.19	37.72	37.32	47.79	34.72	51.72	18..64	14.36	19.16	8.06	11.67	27.36	36.32	40.00	43.79	*7.69*	*10.61*	*15.90*
	RMSE	28.70	23.32	24.09	20.42	20.39	26.60	7.62	*9.14*	9.93	*4.83*	9.23	14.78	17.53	19.46	22.39	5.28	9,97	*7.97*
Database5 (Regression)	R^2	-0.1	0.01	0.12	0.32	0.26	0.06	*0.50*	0.28	0.29	0.37	0.22	0.21	-2.99	-2.17	-2.00	0.34	*0.28*	*0.31*
	MAE	12.32	*13.19*	14.64	10.18	11.68	13.97	*8.48*	11.79	12.13	9.29	11.79	12.69	25.81	25.49	28.26	9.61	12.11	*12.10*
	MAPE	50.48	50.16	71.03	40.99	45.18	67.62	*34.50*	*44.11*	54.49	39.74	44.05	*63.27*	84.95	75.26	85.53	38.12	45.25	62.30
	RMSE	14.96	17.02	19.71	11.73	14.65	17.96	*10.11*	*1438*	15.63	11.32	14.96	16.47	28.53	30.14	32.57	11.67	14.44	*15.39*
Database 6 (Multi-class Classification)	AUC	0.78	00.74	0.82	0.82	*0.80*	0.84	*0.83*	*0.80*	*0.87*	0.81	0.78	0.80	0.81	0.79	0.83	0.81	0.76	*0.80*
	Accuracy	0.30	0.42	0.67	0.52	0.46	0.57	*0.47*	0.42	*0.71*	*0.47*	0.43	0.42	0.43	*0.5*	0.42	0.43	0.46	0.46
	Balanced Accuracy	0.36	0.38	0.62	*0.49*	0.44	0.51	0.48	0.43	*0.67*	0.45	0.42	0.42	0.42	*0.48*	0.37	0.42	0.44	0.42
	Log Loss	4.01	4.04	4.38	1.16	*1.20*	1.09	*1.15*	*1.20*	*0.97*	*1.15*	1.24	1.17	1.19	1.31	1.18	1.21	1.48	1.43
Database 6 (Binary Classification)	AUC	0.69	0.65	0.78	0.76	0.82	0.86	0.77	*0.85*	*0.87*	*0.79*	0.77	0.85	0.76	0.73	0.74	0.76	0.80	0.76
	Accuracy	0.67	0.70	0.76	0.77	0.76	*0.84*	*0.78*	0.79	*0.84*	0.77	*0.78*	0.81	0.74	0.76	0.79	*0.78*	*0.78*	0.79
	Balanced Accuracy	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
	Log Loss	0.61	0.88	0.52	0.60	0.51	*0.43*	0.64	*0.48*	0.48	*0.52*	0.52	0.48	0.54	0.56	0.58	0.65	0.55	0.71
Database 6 (Regression)	R^2	0.44	0.58	0.32	0.54	0.69	*0.50*	*0.73*	*0.77*	0.45	0.57	0.64	0.49	0.51	0.51	0.41	0.48	0.65	0.29
	MAE	45.15	43.20	45.93	41.00	36.54	34.42	*30.76*	*30.51*	*33.26*	39.21	40.36	38.72	41.18	46.47	50.01	41.79	37.70	42.88
	MAPE	27.15	32.15	32.92	27.13	27.21	29.19	*20.16*	*22.42*	*21.53*	28.54	30.55	34.20	30.04	37.93	44.42	28.01	26.57	28.92
	RMSE	67.63	60.25	91.1	61.47	50.80	*77.43*	*47.29*	*43.86*	81.38	59.49	56.10	78.27	63.33	64.66	84.37	64.81	54.78	92.30
Leaderboard	Recurrence	0	1	0	1	2	3	*24*	*22*	*17*	*10*	9	9	0	2	0	1	2	8
	%	0	2	0	2	5.6	8.3	*66.7*	*61.2*	*47*	30.5	19.4	25	0	5.6	0	2	5.6	22

of databases, 2) used performance metrics, and 3) repeatability of predictability [120,121]. This paper covers the first two sub-domain, and as such, work is needed to benchmark the latter by examining derivates of feature selection techniques, model tuning parameters (in terms of the learning rate, loss functions, activation functions, hyperparameter tuning etc.), use of optimizers, hybrid and ensemble modeling approaches. For example, Degtyarev [32] showed how finetuning some of the noted parameters above can result in large improvements (whether in terms of shorter processing time, or attaining higher accuracy metrics). In addition, one must not forget benchmarking hardware or cloud services associated with ML modeling as well. Such benchmarking may lead to developing eco-friendly or green ML models that do not require intense energy resources to solve structural or fire engineering problems.

2. Conclusions

This paper presents a framework for developing, benchmarking, and validating commonly adopted supervised learning ML algorithms against databases compiled for structural and fire engineering problems. These presented datasets cover six domains, 1) elemental response of CFST circular CFST columns at ambient conditions, 2) shear response of CFS channels with slotted webs, 3) compressive strength of concrete, 4) fatigue life data, 5) shear strength of RC and FRP-strengthened beams and fire engineering; and 6) fire behavior of RC concrete columns in terms of spalling occurrence and fire resistance. In total, six algorithms were benchmarked including; Decision Trees (DT), Random Forest (RF), Extreme Gradient Boosted Trees (ExGBT), Light Gradient Boosted Trees (LGBT), TensorFlow Deep Learning (TFDL), and Keras Deep Residual Neural Network (KDP). Holistically, the presented paper establishes the first step towards a unified framework that can be used to accelerate the adoption of ML into structural and fire engineering domains.

The following list of inferences can also be drawn from the findings of this study:

- All selected algorithms in their default settings seem to properly capture the structural and fire engineering phenomena examined herein (with satisfactory and varying levels of success). This implies that structural and fire engineers can adopt raw algorithms as is, as opposed to developing complex ML models or undergo painful programming exercises. This also implies that complications arising due to engineers' historically limited knowledge on ML coding (given the lack of ML presentations into structural and fire engineering curriculum) can be easily overcome.
- Of all algorithms showcased herein, both Extreme Gradient Boosted Trees (ExGBT), Light Gradient Boosted Trees (LGBT) seem to rank the highest on the carried-out tests.
- As expected, out of all examined algorithms, not a single algorithm was found to be dominant in all of the carried-out examinations. This highlights the need for adopting multiple algorithm search and multiple performance metrics in a given ML analysis.

- Benchmarking efforts are encouraged to continue to develop accepted databases and performance evaluations of ML algorithms since the integration of ML into our domains is on the horizon. Early efforts will not only ensure a smooth transition into automation within our historically slow-adapting fields but will also negate existing hurdles observed in attempting to unified FE simulation methods.

Data availability

Some or all data, models, or code that support the findings of this study are available from the corresponding author upon reasonable request. All of the presented databases are hosted online on public repositories (and complete links to these databases are shown herein).

Links to databases:

Database 1 [59]. Database 2 [122]. Database 3 [123]. Database 4 [63,64]. Database 5 [55,67]. Database 6 [54].

Author statement

All authors contributed equally to this manuscript.

Declaration of competing interest

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

References

[1] B.T. Christensen, C.D. Schunn, The relationship of analogical distance to analogical function and preinventive structure: the case of engineering design, Mem. Cognit. (2007), https://doi.org/10.3758/BF03195939.
[2] É. Stauffer, J.J. Lentini, ASTM standards for fire debris analysis: a review, Forensic Sci. Int. (2003), https://doi.org/10.1016/S0379-0738(02)00459-0.
[3] V.K.R. Kodur, S. Banerji, R. Solhmirzaei, Test methods for characterizing concrete properties at elevated temperature. Fire Mater., 2020, pp. 381–395, https://doi.org/10.1002/fam.2777.
[4] M. Safiuddin, N. Hearn, Comparison of ASTM saturation techniques for measuring the permeable porosity of concrete, Cement Concr. Res. (2005), https://doi.org/10.1016/j.cemconres.2004.09.017.
[5] O.C. Zienkiewicz, R.L. Taylor, The Finite Element Method Volume 1 : the Basis, Methods, 2000.
[6] A.E. Tekkaya, A guide for validation of FE-simulations in bulk metal forming, Arabian J. Sci. Eng. (2005).
[7] J. Ferreira, T. Gernay, J. Franssen, O. Vassant, Discussion on a systematic approach to validation of software for structures in fire - romeiro Ferreira Joao Daniel, SiF 2018 10th Int. Conf. Struct. Fire, Belfast, accessed, http://hdl.handle.net/2268/223208, 2018. (Accessed 1 April 2020).
[8] J.G. Teng, S.T. Smith, J. Yao, J.F. Chen, Intermediate crack-induced debonding in RC beams and slabs. Constr. Build. Mater., 2003, https://doi.org/10.1016/S0950-0618(03)00043-6.
[9] M.Z. Naser, R.A. Hawileh, H.A. Rasheed, Performance of RC T-beams externally strengthened with CFRP laminates under elevated temperatures, J. Struct. Fire Eng. 5 (2014) 1–24, https://doi.org/10.1260/2040-2317.5.1.1.
[10] A. Abu-Obeidah, R.A. Hawileh, J.A. Abdalla, Finite element analysis of strengthened RC beams in shear with aluminum plates, Comput. Struct. (2015), https://doi.org/10.1016/j.compstruc.2014.10.009.
[11] A. Ahmed, V.K.R. Kodur, Effect of bond degradation on fire resistance of FRP-strengthened reinforced concrete beams, Compos. B Eng. (2011), https://doi.org/10.1016/j.compositesb.2010.11.004.
[12] M.F. Hassanein, V.I. Patel, M. Elchalakani, H.T. Thai, Finite element analysis of large diameter high strength octagonal CFST short columns, Thin-Walled Struct. (2018), https://doi.org/10.1016/j.tws.2017.11.007.
[13] Z. Huang, Y. Tu, S. Meng, N. Bagge, J. Nilimaa, T. Blanksvärd, Validation of a numerical method for predicting shear deformation of reinforced concrete beams, Eng. Struct. (2019), https://doi.org/10.1016/j.engstruct.2019.109367.
[14] A. Elkady, D.G. Lignos, Improved seismic design and nonlinear modeling recommendations for wide-flange steel columns, J. Struct. Eng. (2018), https://doi.org/10.1061/(asce)st.1943-541x.0002166.
[15] E.I.S. Flores, K. Saavedra, J. Hinojosa, Y. Chandra, R. Das, Multi-scale modelling of rolling shear failure in cross-laminated timber structures by homogenisation and cohesive zone models, Int. J. Solid Struct. (2016), https://doi.org/10.1016/j.ijsolstr.2015.11.027.
[16] J. He, F. Pan, C.S. Cai, F. Habte, A. Chowdhury, Finite-element modeling framework for predicting realistic responses of light-frame low-rise buildings under wind loads, Eng. Struct. (2018), https://doi.org/10.1016/j.engstruct.2018.01.034.
[17] M. Hajjar, E. Hantouche, A. El Ghor, Shear tab connection with composite beam subjected to transient-state fire temperatures: a rational model for design, J. Struct. Fire Eng. (2019), https://doi.org/10.1108/JSFE-11-2018-0037.
[18] V.K.R. Kodur, M.Z. Naser, Approach for shear capacity evaluation of fire exposed steel and composite beams, J. Constr. Steel Res. 141 (2018), https://doi.org/10.1016/j.jcsr.2017.11.011.
[19] B. Szabó, I. Babuška, Introduction to Finite Element Analysis: Formulation, Verification and Validation, 2011, https://doi.org/10.1002/9781119993834.
[20] M.Z. Naser, Deriving temperature-dependent material models for structural steel through artificial intelligence, Construct. Build. Mater. 191 (2018) 56–68, https://doi.org/10.1016/J.CONBUILDMAT.2018.09.186.
[21] A. Dexters, R.R. Leisted, R. Van Coile, S. Welch, G. Jomaas, Testing for knowledge: maximising information obtained from fire tests by using machine learning techniques, Inter (2019), 2019, http://hdl.handle.net/1854/LU-8622485.
[22] M.Z. Naser, AI-based cognitive framework for evaluating response of concrete structures in extreme conditions, Eng. Appl. Artif. Intell. 81 (2019), https://doi.org/10.1016/j.engappai.2019.03.004.
[23] V. Zope, T. Dadlani, A. Matai, P. Tembhurnikar, R. Kalani, IoT sensor and deep neural network based wildfire prediction system. Proc. Int. Conf. Intell. Comput. Control Syst. ICICCS 2020, Institute of Electrical and Electronics Engineers Inc., 2020, pp. 205–208, https://doi.org/10.1109/ICICCS48265.2020.9120949.
[24] P.J. Roache, Verification and validation in computational science and engineering, Comput. Sci. Eng. (1998).
[25] I. Babuska, J.T. Oden, Verification and validation in computational engineering and science: basic concepts, Comput. Methods Appl. Mech. Eng. (2004), https://doi.org/10.1016/j.cma.2004.03.002.
[26] B.H. Thacker, S.W. Doebling, F.M. Hemez, M.C. Anderson, J.E. Pepin, E. a. Rodriguez, Concepts of Model Verification and Validation, Concepts Model Verif. Valid, 2004.
[27] J.S. Carson, Model verification and validation. Winter Simul. Conf. Proc., 2002, https://doi.org/10.1007/978-1-4615-0863-2_17.
[28] A. Behnood, E.M. Golafshani, Machine learning study of the mechanical properties of concretes containing waste foundry sand, Construct. Build. Mater. (2020), https://doi.org/10.1016/j.conbuildmat.2020.118152.
[29] A. Behnood, E. Mohammadi Golafshani, Predicting the dynamic modulus of asphalt mixture using machine learning techniques: an application of multi biogeography-based programming, Construct. Build. Mater. (2021), https://doi.org/10.1016/j.conbuildmat.2020.120983.
[30] S. Mangalathu, H. Jang, S.H. Hwang, J.S. Jeon, Data-driven machine-learning-based seismic failure mode identification of reinforced concrete shear walls, Eng. Struct. (2020), https://doi.org/10.1016/j.engstruct.2020.110331.
[31] S. Mangalathu, S.H. Hwang, E. Choi, J.S. Jeon, Rapid seismic damage evaluation of bridge portfolios using machine learning techniques, Eng. Struct. (2019), https://doi.org/10.1016/j.engstruct.2019.109785.
[32] V.V. Degtyarev, Neural networks for predicting shear strength of CFS channels with slotted webs, J. Constr. Steel Res. (2021), https://doi.org/10.1016/j.jcsr.2020.106443.
[33] V. Degtyarev, N. Degtyareva, Elastic Shear Buckling Loads and Ultimate Shear Strengths of CFS Channels with Slotted Webs: FE Simulation Results, 2020. https://doi.org/10.17632/t9hhkngdwv.1.
[34] G.S. Bobadilha, C.E. Stokes, D.J. Verly Lopes, Artificial neural networks modelling based on visual analysis of coated cross laminated timber (CLT) to predict color change during outdoor exposure, Holzforschung (2020), https://doi.org/10.1515/hf-2020-0193.
[35] D.J. Verly Lopes, G. dos S. Bobadilha, K.M. Grebner, A fast and robust artificial intelligence technique for wood knot detection, BioResources (2020), https://doi.org/10.15376/biores.15.4.9351-9361.
[36] Y. Xie, M. Ebad Sichani, J.E. Padgett, R. DesRoches, The promise of implementing machine learning in earthquake engineering: a state-of-the-art review, Earthq. Spectra (2020), https://doi.org/10.1177/8755293020919419.
[37] Y.J. Cha, W. Choi, O. Büyüköztürk, Deep learning-based crack damage detection using convolutional neural networks, Comput. Civ. Infrastruct. Eng. (2017), https://doi.org/10.1111/mice.12263.
[38] Y. Zhang, H.V. Burton, H. Sun, M. Shokrabadi, A machine learning framework for assessing post-earthquake structural safety, Struct. Saf. (2018), https://doi.org/10.1016/j.strusafe.2017.12.001.
[39] H. Huang, H.V. Burton, Classification of in-plane failure modes for reinforced concrete frames with infills using machine learning, J. Build. Eng. (2019), https://doi.org/10.1016/j.jobe.2019.100767.
[40] J. Wäldchen, P. Mäder, Machine learning for image based species identification, Methods Ecol. Evol. (2018), https://doi.org/10.1111/2041-210X.13075.
[41] Q. Zhang, J. Xu, L. Xu, H. Guo, Deep Convolutional Neural Networks for Forest Fire Detection, 2016, https://doi.org/10.2991/ifmeita-16.2016.105.
[42] A.H. Alavi, A.H. Gandomi, Prediction of principal ground-motion parameters using a hybrid method coupling artificial neural networks and simulated annealing, Comput. Struct. (2011), https://doi.org/10.1016/j.compstruc.2011.08.019.
[43] M. Flah, A.R. Suleiman, M.L. Nehdi, Classification and quantification of cracks in concrete structures using deep learning image-based techniques, Cement Concr. Compos. (2020), https://doi.org/10.1016/j.cemconcomp.2020.103781.
[44] W. Ben Chaabene, M. Flah, M.L. Nehdi, Machine learning prediction of mechanical properties of concrete: critical review, Construct. Build. Mater. (2020), https://doi.org/10.1016/j.conbuildmat.2020.119889.

[45] M.Z. Naser, Autonomous fire resistance evaluation, ASCE Jounral Struct. Eng. 146 (2020), https://doi.org/10.1061/(ASCE)ST.1943-541X.0002641.
[46] M.Z. Naser, A. Seitllari, Concrete under fire: an assessment through intelligent pattern recognition, Eng. Comput. 36 (2020), https://doi.org/10.1007/s00366-019-00805-1.
[47] R. Solhmirzaei, H. Salehi, V. Kodur, M.Z. Naser, Machine learning framework for predicting failure mode and shear capacity of ultra high performance concrete beams, Eng. Struct. (2020), https://doi.org/10.1016/j.engstruct.2020.111221.
[48] J.A. Abdalla, R.A. Hawileh, Artificial neural network predictions of fatigue life of steel bars based on hysteretic energy, J. Comput. Civ. Eng. (2013), https://doi.org/10.1061/(asce)cp.1943-5487.0000185.
[49] J.A. Abdalla, R. Hawileh, Modeling and simulation of low-cycle fatigue life of steel reinforcing bars using artificial neural network, J. Franklin Inst., 2011, https://doi.org/10.1016/j.jfranklin.2010.04.005.
[50] H. Nguyen, T. Vu, T.P. Vo, H.T. Thai, Efficient machine learning models for prediction of concrete strengths, Construct. Build. Mater. (2021), https://doi.org/10.1016/j.conbuildmat.2020.120950.
[51] M. Zarringol, H.T. Thai, S. Thai, V. Patel, Application of ANN to the Design of CFST Columns, Structures, 2020, https://doi.org/10.1016/j.istruc.2020.10.048.
[52] M.Z. Naser, S. Thai, H.-T. Thai, Evaluating structural response of concrete-filled steel tubular columns through machine learning, J. Build. Eng. (2020), https://doi.org/10.1016/j.jobe.2020.101888.
[53] M.Z. Naser, Observational analysis of fire-induced spalling of concrete through ensemble machine learning and surrogate modeling, J. Mater. Civ. Eng. 33 (2021), 04020428, https://doi.org/10.1061/(ASCE)MT.1943-5533.0003525.
[54] M. Naser, Database for Fire-Induced Spalling in Reinforced Concrete (RC) Columns, 2020. https://doi.org/10.17632/zdb2wd2h77.4.
[55] J.A. Abdalla, A. Elsanosi, A. Abdelwahab, Modeling and simulation of shear resistance of R/C beams using artificial neural network, J. Franklin Inst. (2007), https://doi.org/10.1016/j.jfranklin.2005.12.005.
[56] O.R. Abuodeh, J.A. Abdalla, R.A. Hawileh, Prediction of shear strength and behavior of RC beams strengthened with externally bonded FRP sheets using machine learning techniques, Compos. Struct. (2020), https://doi.org/10.1016/j.compstruct.2019.111698.
[57] I.-C. Yeh, Modeling of strength of high-performance concrete using artificial neural networks, Cement Concr. Res. 28 (1998) 1797–1808, https://doi.org/10.1016/S0008-8846(98)00165-3.
[58] S. Thai, H.T. Thai, B. Uy, T. Ngo, Concrete-filled steel tubular columns: test database, design and calibration, J. Constr. Steel Res. (2019), https://doi.org/10.1016/j.jcsr.2019.02.024.
[59] S. Thai, H. Thai, B. Uy, T. Ngo, M. Naser, Test Database on Concrete-Filled Steel Tubular Columns, 2019, https://doi.org/10.17632/3XKNB3SDB5.1.
[60] V.V. Degtyarev, N.V. Degtyareva, Numerical simulations on cold-formed steel channels with flat slotted webs in shear. Part I: elastic shear buckling characteristics, Thin-Walled Struct. (2017), https://doi.org/10.1016/j.tws.2017.05.026.
[61] V.V. Degtyarev, N.V. Degtyareva, Numerical simulations on cold-formed steel channels with flat slotted webs in shear. Part II: ultimate shear strength, Thin-Walled Struct. (2017), https://doi.org/10.1016/j.tws.2017.05.028.
[62] V.V. Degtyarev, N.V. Degtyareva, Numerical simulations on cold-formed steel channels with longitudinally stiffened slotted webs in shear, Thin-Walled Struct. (2018), https://doi.org/10.1016/j.tws.2018.05.001.
[63] J.A. Abdalla, R.A. Hawileh, F. Oudah, K. Abdelrahman, Energy-based prediction of low-cycle fatigue life of BS 460B and BS B500B steel bars, Mater. Des. (2009), https://doi.org/10.1016/j.matdes.2009.04.003.
[64] R.A. Hawileh, J.A. Abdalla, F. Oudah, K. Abdelrahman, Low-cycle fatigue life behaviour of BS 460B and BS B500B steel reinforcing bars, Fatig. Fract. Eng. Mater. Struct. (2010), https://doi.org/10.1111/j.1460-2695.2010.01452.x.
[65] M.M. Kashani, A.K. Barmi, V.S. Malinova, Influence of inelastic buckling on low-cycle fatigue degradation of reinforcing bars, Construct. Build. Mater. (2015), https://doi.org/10.1016/j.conbuildmat.2015.07.102.
[66] M. Tripathi, R.P. Dhakal, F. Dashti, L.M. Massone, Low-cycle fatigue behaviour of reinforcing bars including the effect of inelastic buckling, Construct. Build. Mater. (2018), https://doi.org/10.1016/j.conbuildmat.2018.09.192.
[67] O.R. Abuodeh, J.A. Abdalla, R.A. Hawileh, Prediction of shear strength and behavior of RC beams strengthened with externally bonded FRP sheets using machine learning techniques, Compos. Struct. (2019) 111698, https://doi.org/10.1016/J.COMPSTRUCT.2019.111698.
[68] K.D.D. Hertz, Limits of spalling of fire-exposed concrete, Fire Saf. J. 38 (2003) 103–116, https://doi.org/10.1016/S0379-7112(02)00051-6.
[69] T.Z. Harmathy, Effect of Mositure on the Fire Endurance of Building Elements, ASTM Spec. Tech. Publ., 1965, https://doi.org/10.1520/STP48429S.
[70] V.R. Kodur, R.C. Mcgrath, J.C. Latour, J.W. Maclaurin, Experimental Studies for Evaluating the Fire Endurance of High- Strength Concrete Columns, 2000. http://nparc.cisti-icist.nrc-cnrc.gc.ca/npsi/ctrl?lang=en.
[71] J.-C.C. Liu, K.H. Tan, Y. Yao, A new perspective on nature of fire-induced spalling in concrete, Construct. Build. Mater. 184 (2018) 581–590, https://doi.org/10.1016/j.conbuildmat.2018.06.204.
[72] L.T. Phan, N.J. Carino, Fire performance of high strength concrete: research needs. Adv. Technol. Struct. Eng., American Society of Civil Engineers, Reston, VA, 2000, pp. 1–8. https://doi.org/10.1061/40492(2000)181.
[73] N. Raut, V. Kodur, Response of reinforced concrete columns under fire-induced biaxial bending, ACI Struct. J. (2011), https://doi.org/10.14359/51683218.
[74] Z.P. Bažant, M.F. Kaplan, Z.P. Bazant, Concrete at High Temperatures: Material Properties and Mathematical Models, 1996 accessed, https://www.scholars.northwestern.edu/en/publications/concrete-at-high-temperatures-material-properties-and-mathematica. (Accessed 28 November 2018).
[75] F.-J. Ulm, O. Coussy, Z.P. Bažant, The "chunnel" fire. I: chemoplastic softening in rapidly heated concrete, J. Eng. Mech. 125 (1999) 272–282. https://doi.org/10.1061/(ASCE)0733-9399(1999)125:3(272).
[76] T.-Y. Song, L.-H. Han, Z. Tao, Structural behavior of SRC beam-to-column joints subjected to simulated fire including cooling phase, J. Struct. Eng. (2015). https://doi.org/10.1061/(ASCE)ST.1943-541X.0001211.
[77] A.H. Shah, U.K. Sharma, Fire resistance and spalling performance of confined concrete columns, Construct. Build. Mater. 156 (2017) 161–174, https://doi.org/10.1016/j.conbuildmat.2017.08.167.
[78] V. Kodur, F. Cheng, T. Wang, J. Latour, P. Leroux, Fire Resistance of High-Performance Concrete Columns, 2001. http://doi.org/10.4224/20378506.
[79] E.W.H. Klingsch, Explosive Spalling of Concrete in Fire, Tese Doutorado, 2014, https://doi.org/10.3929/ethz-a-010076314.
[80] M.Z. Naser, Heuristic machine cognition to predict fire-induced spalling and fire resistance of concrete structures, Autom. ConStruct. 106 (2019) 102916, https://doi.org/10.1016/J.AUTCON.2019.102916.
[81] V. Kodur, R. McGrath, P. Leroux, J. Latour, Experimental Studies for Evaluating the Fire Endurance of High-Strength Concrete Columns, 2005. http://doi.org/10.4224/20378032.
[82] H. Sun, H.V. Burton, H. Huang, Machine learning applications for building structural design and performance assessment: state-of-the-art review, J. Build. Eng. 33 (2021) 101816, https://doi.org/10.1016/j.jobe.2020.101816.
[83] P. Lu, S. Chen, Y. Zheng, Artificial intelligence in civil engineering, Math. Probl Eng. (2012) 1–23, https://doi.org/10.1155/2012/145974, 2012.
[84] M.Z. Naser, Mechanistically informed machine learning and artificial intelligence in fire engineering and sciences, Fire Technol. (2021) 1–44, https://doi.org/10.1007/s10694-020-01069-8.
[85] S.R. Safavian, D. Landgrebe, A survey of decision tree classifier methodology, IEEE Trans. Syst. Man. Cybern. 21 (1991) 660–674, https://doi.org/10.1109/21.97458.
[86] Scikit, sklearntree, DecisionTreeClassifier — Scikit-Learn 0.24.1 Documentation, Scikit, 2020 accessed, https://scikit-learn.org/stable/modules/generated/sklearn.tree.DecisionTreeClassifier.html. (Accessed 9 February 2021).
[87] Decision Tree SciKit, accessed, https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.ExtraTreesClassifier.html?highlight=decision tree#sklearn.ensemble.ExtraTreesClassifier.decision_path, 2020. (Accessed 22 January 2021).
[88] D. Che, Q. Liu, K. Rasheed, X. Tao, Decision Tree and Ensemble Learning Algorithms with Their Applications in Bioinformatics, Springer, New York, NY, 2011, pp. 191–199, https://doi.org/10.1007/978-1-4419-7046-6_19.
[89] J.-S.S. Chou, C.-F.F. Tsai, A.-D.D. Pham, Y.-H.H. Lu, Machine learning in concrete strength simulations: multi-nation data analytics, Construct. Build. Mater. 73 (2014) 771–780, https://doi.org/10.1016/j.conbuildmat.2014.09.054.
[90] A. Liaw, M. Wiener, Classification and Regression by RandomForest, 2002 accessed April 8, 2019, https://www.researchgate.net/publication/228451484.
[91] Scikit, sklearn.ensemble.RandomForestClassifier — Scikit-Learn 0.24.1 Documentation, 2020 accessed February 9, 2021, https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.RandomForestClassifier.html.
[92] Y. Freund, R.E. Schapire, A decision-theoretic generalization of on-line learning and an application to boosting, J. Comput. Syst. Sci. (1997), https://doi.org/10.1006/jcss.1997.1504.
[93] Gradient Boosted Tree (GBT), 2019 accessed April 9, 2019, https://software.intel.com/en-us/daal-programming-guide-details-24.
[94] Scikit, sklearn.ensemble.GradientBoosting, Regressor — Scikit-Learn 0.24.1 Documentation, 2020 accessed, https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.GradientBoostingRegressor.html. (Accessed 9 February 2021).
[95] XGBoost Python Package, Python Package Introduction — Xgboost 1.4.0-SNAP-SHOT Documentation, 2020 accessed, https://xgboost.readthedocs.io/en/latest/python/python_intro.html#early-stopping. (Accessed 10 February 2021).
[96] Y. Freund, R.E. Schapire, Experiments with a new boosting algorithm, Proc. 13th Int. Conf. Mach. Learn (1996). https://cseweb.ucsd.edu/~yfreund/papers/boostingexperiments.pdf.
[97] LightGBM, Welcome to LightGBM's Documentation! — LightGBM 3.1.1.99 Documentation, 2020 accessed, https://lightgbm.readthedocs.io/en/latest/. (Accessed 9 February 2021).
[98] TensorFlow, GitHub - Tensorflow/tensorflow: an Open Source Machine Learning Framework for Everyone, 2020 accessed, https://github.com/tensorflow/tensorflow. (Accessed 9 February 2021).
[99] H. Li, Z. Xu, G. Taylor, C. Studer, T. Goldstein, Visualizing the loss landscape of neural nets, Adv. Neural Inf. Process. Syst. (2018).
[100] Keras, GitHub - Keras-Team/keras: Deep Learning for Humans, 2020 accessed, https://github.com/keras-team/keras. (Accessed 9 February 2021).
[101] M.D. Schmidt, H. Lipson, Age-fitness pareto optimization. https://doi.org/10.1145/1830483.1830584, 2010.
[102] P. Cremonesi, Y. Koren, R. Turrin, Performance of Recommender Algorithms on Top-N Recommendation Tasks Categories and Subject Descriptors, RecSys, 2010.
[103] M. Laszczyk, P.B. Myszkowski, Survey of quality measures for multi-objective optimization: construction of complementary set of multi-objective quality measures, Swarm Evol. Comput. 48 (2019) 109–133, https://doi.org/10.1016/J.SWEVO.2019.04.001.
[104] M.Z. Naser, A. Alavi, Insights into Performance Fitness and Error Metrics for Machine Learning, 2020 accessed, http://arxiv.org/abs/2006.00887. (Accessed 4 August 2020).

[105] A. Botchkarev, A new typology design of performance metrics to measure errors in machine learning regression algorithms, Interdiscipl. J. Inf. Knowl. Manag. (2019), https://doi.org/10.28945/4184.
[106] A.H. Alavi, A.H. Gandomi, M.G. Sahab, M. Gandomi, Multi expression programming: a new approach to formulation of soil classification, Eng. Comput. 26 (2010) 111–118, https://doi.org/10.1007/s00366-009-0140-7.
[107] M.Z. Naser, A. Seitllari, Concrete under fire: an assessment through intelligent pattern recognition, Eng. Comput. (2019) 1–14, https://doi.org/10.1007/s00366-019-00805-1.
[108] W.Z. Taffese, E. Sistonen, Machine learning for durability and service-life assessment of reinforced concrete structures: recent advances and future directions, Autom. ConStruct. (2017), https://doi.org/10.1016/j.autcon.2017.01.016.
[109] C.J. Willmott, K. Matsuura, Advantages of the mean absolute error (MAE) over the root mean square error (RMSE) in assessing average model performance, Clim. Res. (2005), https://doi.org/10.3354/cr030079.
[110] S. Makridakis, Accuracy measures: theoretical and practical concerns, Int. J. Forecast. (1993), https://doi.org/10.1016/0169-2070(93)90079-3.
[111] M.V. Shcherbakov, A. Brebels, N.L. Shcherbakova, A.P. Tyukov, T.A. Janovsky, V. A. evich Kamaev, A survey of forecast error measures, World Appl. Sci. J. (2013), https://doi.org/10.5829/idosi.wasj.2013.24.itmies.80032.
[112] A. Sauca, C. Zhang, A. Chernovsky, M. Seif, Communication framework for hybrid fire testing: developments and applications in virtual and real environments, Fire Saf. J. (2020), https://doi.org/10.1016/j.firesaf.2019.102937.
[113] M. Aumüller, E. Bernhardsson, A. Faithfull, ANN-Benchmarks: a benchmarking tool for approximate nearest neighbor algorithms, Inf. Syst. (2020), https://doi.org/10.1016/j.is.2019.02.006.
[114] H. Xiao, K. Rasul, R. Vollgraf, Fashion-mniST: A Novel Image Dataset for Benchmarking Machine Learning Algorithms, ArXiv, 2017.
[115] J. Stallkamp, M. Schlipsing, J. Salmen, C. Igel, Man vs. Computer: Benchmarking Machine Learning Algorithms for Traffic Sign Recognition, Neural Networks, 2012, https://doi.org/10.1016/j.neunet.2012.02.016.
[116] Z. Wu, B. Ramsundar, E.N. Feinberg, J. Gomes, C. Geniesse, A.S. Pappu, K. Leswing, V. Pande, MoleculeNet: a benchmark for molecular machine learning, Chem. Sci. (2018), https://doi.org/10.1039/c7sc02664a.
[117] J. Redmon, A. Farhadi, YOLO9000: better, faster, stronger. Proc. IEEE Conf. Comput. Vis. Pattern Recognit., 2017 accessed January 10, 2019, http://pjreddie.com/yolo9000/.
[118] Y. Gao, K.M. Mosalam, PEER Hub ImageNet, A large-scale multiattribute benchmark data set of structural images, J. Struct. Eng. (2020), https://doi.org/10.1061/(asce)st.1943-541x.0002745.
[119] X. Wu, Y. Park, A. Li, X. Huang, F. Xiao, A. Usmani, Smart Detection of Fire Source in Tunnel Based on the Numerical Database and Artificial Intelligence, Fire Technol, 2020, https://doi.org/10.1007/s10694-020-00985-z.
[120] C. Parker, On measuring the performance of binary classifiers, Knowl. Inf. Syst. (2013), https://doi.org/10.1007/s10115-012-0558-x.
[121] D. Berrar, An empirical evaluation of ranking measures with respect to robustness to noise, J. Artif. Intell. Res. (2014), https://doi.org/10.1613/jair.4136.
[122] V. Degtyarev, Elastic Buckling and Post-buckling Strength of CFS Decks in Bending: FE Simulation Results, 2020, p. 2, https://doi.org/10.17632/3KM4B3SFCW.2.
[123] I. Yeh, UCI Machine Learning Repository: Concrete Compressive Strength Data Set, 2007 accessed July 7, 2021, http://archive.ics.uci.edu/ml/datasets/concrete+compressive+strength.
[124] M. van Smeden, K.G. Moons, J.A. de Groot, G.S. Collins, D.G. Altman, M.J. Eijkemans, J.B. Reitsma, Sample Size for Binary Logistic Prediction Models: beyond Events Per Variable Criteria:, Https://Doi.Org/10.1177/0962280218784726. 28 (2018) 2455–2474. https://doi.org/10.1177/0962280218784726.
[125] R.D. Riley, K.I.E. Snell, J. Ensor, D.L. Burke, F.E. Harrell, K.G.M. Moons, G. S. Collins, Minimum sample size for developing a multivariable prediction model: PART II - binary and time-to-event outcomes, Stat. Med. (2019), https://doi.org/10.1002/sim.7992.