Front. Struct. Civ. Eng. 2024, 18(4): 587–614
https://doi.org/10.1007/s11709-024-1052-8

RESEARCH ARTICLE

Fire resistance evaluation through synthetic fire tests and generative adversarial networks

Aybike Özyüksel Çiftçioğlu[a*], M.Z. Naser[b,c]

[a] *Department of Civil Engineering, Manisa Celal Bayar University, 45140, Türkiye*
[b] *School of Civil & Environmental Engineering and Earth Sciences (SCEEES), Clemson University, Clemson, SC 29631, USA*
[c] *Artificial Intelligence Research Institute for Science and Engineering (AIRISE), Clemson University, Clemson, SC 29631, USA*
[*] *Corresponding author. E-mail: aybike.ozyuksel@cbu.edu.tr*

© The Author(s) 2024. This article is published with open access at link.springer.com and journal.hep.com.cn

ABSTRACT This paper introduces a machine learning approach to address the challenge of limited data resulting from costly and time-consuming fire experiments by enlarging small fire test data sets and predicting the fire resistance of reinforced concrete columns. Our approach begins by creating deep learning models, namely generative adversarial networks and variational autoencoders, to learn the spatial distribution of real fire tests. We then use these models to generate synthetic tabular samples that closely resemble realistic fire resistance values for reinforced concrete columns. The generated data are employed to train state-of-the-art machine learning techniques, including Extreme Gradient Boost, Light Gradient Boosting Machine, Categorical Boosting Algorithm, Support Vector Regression, Random Forest, Decision Tree, Multiple Linear Regression, Polynomial Regression, Support Vector Machine, Kernel Support Vector Machine, Naive Bayes, and K-Nearest Neighbors, which can predict the fire resistance of the columns through regression and classification. Machine learning analyses achieved highly accurate predictions of fire resistance values, outperforming traditional models that relied solely on limited experimental data. Our study highlights the potential for using machine learning and deep learning analyses to revolutionize the field of structural engineering by improving the accuracy and efficiency of fire resistance evaluations while reducing the reliance on costly and time-consuming experiments.

KEYWORDS deep learning, fire resistance, generative adversarial networks, machine learning

1 Introduction

Artificial intelligence (AI) represents a groundbreaking innovation in modern engineering by affording machines the ability to learn and solve complex problems analogously to human cognition. AI has revolutionized several fields of computer science and engineering, where it has enabled the development of advanced systems for solving some of the most intractable problems in modern society. Learning from data has become a transformative technology that has revolutionized many fields of engineering and computer science with its ability to reason and make predictions about complex relationships. Among many applications, AI excels at enhancing the precision of predictions, optimizing design models, and updating operational processes. Furthermore, it has the capacity to identify obscure correlations and patterns in data, as well as to augment the efficiency of decision-making [1–8].

AI has garnered considerable attention in recent years due to its potential to revolutionize many fields, including structural and fire engineering [9–11]. The use of AI in these fields has aided numerous tasks, such as load prediction, material characterization, fire spread prediction, identification of fire hazards, and development of evacuation plans. These applications represent a significant shift in the traditional methods of engineering and have enabled the development of more accurate, reliable, and efficient solutions to some of the most pressing challenges in modern society [12–15]. Although the use of AI in fire safety engineering is still in its early stage, the potential benefits are staggering. Engineers

Article history: Received May 8, 2023; Accepted Jul 18, 2023

may use AI to design safer buildings and structures by anticipating how they would react in a fire, allowing them to apply more effective fire prevention measures. As a result, the use of AI in structural and fire engineering has significant promise for improving public safety and expanding the engineering discipline [16].

Deep learning (DL), a subset of AI, has experienced remarkable advancements in recent years [17–20]. However, the continued reliance on big data has become a significant impediment in the development of DL-based systems. The cost and time associated with collecting vast amounts of data are significant challenges [21]. Similarly, conducting large-scale fire tests is also a time-consuming and costly process. To address these issues, a contemporary solution is to produce synthetic data that accurately mimic the distribution of real fire tests. generative adversarial network (GAN), a novel DL model, has emerged as an innovative approach to producing synthetic data that introduce the concept of data manipulation. GAN has already exhibited remarkable efficacy in an array of engineering applications, encompassing structural analysis, optimization of mechanical systems, material science, and control systems engineering [22,23]. A special type of GAN is the tabular vector auto-encoder (TVAE) [24]. TVAE is a generative model that uses data and transforms it into a tabular vector representation to generate synthetic data.

Exhibiting a resolute superiority over conventional methodologies, GAN and TVAE stand as transformative advancements in the domain of data augmentation. These cutting-edge frameworks not only surmount the arduous challenges entailed in amassing copious volumes of fire test data, but also exhibit unparalleled flexibility and efficiency in producing synthetic data that impeccably capture the essence of real-world scenarios [25]. By transcending the limitations of traditional data collection approaches, GAN and TVAE empower researchers and practitioners to circumvent the laborious and cost-intensive process of acquiring extensive fire test data. Instead, these innovative models engender a realm of possibility, wherein synthetic data, imbued with statistical fidelity akin to genuine fire tests, can be proficiently generated. The inherent flexibility of GAN and TVAE stems from their ability to seamlessly adapt to diverse data domains and distributions. Unlike conventional techniques that often struggle to faithfully replicate the intricate nuances of real-world phenomena, GAN and TVAE employ advanced learning mechanisms to faithfully learn the underlying data distribution, thereby enabling the synthesis of data instances that embody the same statistical properties as their authentic counterparts. In addition to their adaptability, GAN and TVAE epitomize efficiency, offering expeditious means of data generation that circumvent the time-consuming and resource-intensive nature of conventional methodologies. By leveraging the power of parallel computation and optimized training procedures, these state-of-the-art models pave the way for accelerated data synthesis, thereby expediting the pace of research and innovation in fire engineering and related disciplines [26–29].

The transformative potential of GAN and TVAE in the realm of data augmentation reverberates across a myriad of applications, ranging from architectural design optimization to structural analysis, from fire safety planning to evacuation simulations. Their inherent capability to generate synthetic data, indistinguishable from real fire tests, facilitates rigorous experimentation and evaluation, thereby enhancing the reliability and efficiency of engineering solutions. In light of these remarkable attributes, the integration of GAN and TVAE into the realm of civil engineering portends a paradigm shift, revolutionizing the data augmentation landscape and underscoring the transformative potential of DL in addressing the pressing challenges faced by the discipline. The developments in generative modeling have the potential to transform the field of civil engineering, allowing for improved engineering design optimization, manufacturing processes, and construction management. Recent developments in AI have allowed it to be used in a wide range of industries, including engineering, design optimization, manufacturing, and even construction management. However, the application of GANs and TVAE in civil engineering, specifically in structural fire engineering, is still limited [30,31].

Fire is one of the most prevalent risks that structural engineers encounter, and its effects can be catastrophic. High temperatures and powerful flames might cause a structure to deteriorate and collapse. Fire resistance is a fundamental design measure used to protect a structure against fire to decrease this risk. The basic aim of fire resistance is to keep fires from spreading and causing additional damage. Additionally, fire resistance tries to limit the duration and damage caused by the fire [32–35]. In modern high-rise buildings, reinforced concrete (RC) columns are a prevalent load-bearing structural system. Therefore, it is crucial for structural engineers to understand how RC columns behave in a fire. In this respect, the employment of AI techniques such as GANs and TVAEs provides a possible avenue for developing more effective and efficient fire resistance strategies for RC columns and other structural systems.

The contributions of this paper can be summarized as follows.

1) The study aims to address the research gap in predicting the fire resistance duration of RC columns based on experimental tests carried out on columns with different dimensions or material properties. The authors discuss how this complex problem can be classified and predicted.

2) This paper presents the first attempt to develop

classification and regression models using TVAE on fire-resistant RC column data. A large number of synthetic data sets are generated by applying the TVAE DL method on a limited number of real-world RC columns.

3) The outcomes of the classification and regression analyses on both real and synthetic data sets demonstrate the promising potential of the proposed method. A rate of accuracy up to 100% is achieved, especially in the classification.

4) The proposed method presents a novel solution for mitigating the data scarcity issue in fire testing. By leveraging DL, engineers can develop generative models that produce training data with varying distributions, enabling them to analyze fire safety engineering problems further than was previously possible. This approach holds promise for improving the accuracy and detail of fire safety engineering assessments.

2 Materials and methods

In contrast to discriminative models, which exclusively forecast target categories, generative models capture a joint distribution of both object-level and label-level attributes [36,37]. Generative models include DL techniques, such as GANs and variational autoencoders (VAE), designed mostly to learn from unstructured data.

2.1 Generative adversarial networks

As mentioned earlier, GANs offer a DL process that generates synthetic data that is statistically identical to samples from actual data [38]. In an adversarial network, there are two AI agents: a generator and a discriminator. The generator attempts to make the network learn by creating artificially-generated samples. The discriminator trains using the samples produced by the generator, to identify the fake samples. The discriminator is tasked with identifying fake samples as quickly as possible. If the discriminator is successful in quickly recognizing fake samples, the network will learn to make it easier to generate realistic samples.

Many efforts have been made to speed up and stabilize the training process since GANs were initially introduced, with application-based studies mostly concentrating on imagery generation. In a tabular data set, there may be both categorical and continuous values. Because of the imbalance in discrete values, the modeling of existing statistical and deep neural network methods may be difficult. The TVAE is an extension of the original GAN [24]. To generate tabular synthetic data from the fire resistance RC columns, we used the DL model TVAE, which uses the VAE to address these challenges.

2.2 Variational autoencoder

The VAE is composed of two neural networks: the inference (encoder) network is trained to learn about distributions over latent variables, and the generator (decoder) network learns how to transform from these probabilities given observed data. Both networks are updated by a loss function that consists of two factors—one ensures that generated synthetic observations match with training data; another ensures that the estimated distribution for hidden variables matches the prior distribution employed in the training process. This is achieved by the Kullback–Leibler divergence [39], which, when estimating a new model after updating it based on a newly calculated loss function (an error), measures the deviation between the current estimate and the previous one provided as input. The TVAE, which adapts VAE to tabular data by modifying the loss function, is intended to address the difficulty of generating tabular data [24]. The loss function of TVAE is altered to lower-bound losses.

TVAE is the generative DL model performed in this study to generate tabular data for the fire-resistant RC column.

2.3 Overview of AI algorithms

This research utilizes eight AI regressors and nine AI classifiers for fire resistance evaluation of RC columns. In this section, we provide a brief overview of the algorithms adopted, namely: Regression algorithms: Extreme gradient boost (XGBoost), support vector regressor (SVR), random forest (RF), decision tree (DT), multiple linear regression (MLR) and polynomial regression (PR) algorithms, and Classification algorithms: DT, RF, support vector machine (SVM), kernel support vector machine (K-SVM), naive bayes classifier (NBC), and K-nearest neighbors (KNN).

2.3.1 Extreme gradient boosting algorithm

Chen and Guestrin [40] introduced XGBoost, which is applicable to both regression and classification tasks. XGBoost is a scalable implementation of gradient boost DTs and has been commonly used to predict material characteristics and structural behavior in engineering [41–43]. Basically, it is a scalable variant of gradient-boosted DTs that utilize regression trees. A regression tree has leaves, each of which represents a numerical weight. Each input variable is allocated to a set of leaves, resulting in an estimated output for that sample. The predicted output of the model for that sample is calculated by summing the total of leaves allocated to that sample in each regression tree. Each new regression tree is included in a way that minimizes the learning objective [44].

The objective function of XGBoost, as described in Eq. (1), comprises a loss function and a regularization

term. $L(\Theta)$ signifies the loss error, and $\Omega(\Theta)$ is the regularization term.

$$J(\Theta) = L(\Theta) + \Omega(\Theta), \tag{1}$$

The loss function ($L(\Theta)$) which is expressed in Eq. (2), attempts to create an accurate prediction by comparing the difference between the predicted value of the tree model, $\hat{y}_i$, and the actual value of the ith sample, y_i, n is the number of predictions.

$$L(\Theta) = \sum_{i=1}^{n} L(y_i, \hat{y}_i), \tag{2}$$

Equation (3) expresses the regularization term employed to regulate the complexity of the method in order to avoid overfitting. f_k is the prediction function of the kth tree for evaluating output in functional space, and m is the number of trees.

$$\Omega(\Theta) = \sum_{k=1}^{m} \Omega(f_k), \tag{3}$$

Specifically, the regularization term is expressed as in Eq. (4):

$$\Omega(f_k) = \gamma T + \frac{1}{2}\lambda \sum_{j=1}^{T} w_j^2, \tag{4}$$

where T is the number of leaf nodes in the tree, γ and λ are the penalty coefficients used to regulate the complexity of the XGBoost, and w_j denotes the weight of the jth leaf.

2.3.2 Support vector machine

SVM [45] is a powerful supervised learning algorithm applicable for both classification and regression tasks. The SVM finds a hyperplane in a high-dimensional space that separates the data into two or more regions. If a sample is above the separating hyperplane, it is referred to as a positive sample; otherwise, it is a negative sample. In other words, an SVM is a classifier that aims at distinguishing various classes based on their distance from the hyperplane. Its extended version, SVR, is used for more precise prediction [46]. The ensuing equation presents the definition of the SVR function:

$$f(x) = w\Phi(x) + b, \tag{5}$$

where b is the intercept, and w is the weight vector. $\Phi(x)$ denotes a nonlinear mapping function with a high number of dimensions. The following function is used to find the optimal values for the w and b parameters.

$$\min R(F) = \frac{1}{2}\|w\|^2 + C\sum_{i=1}^{n} |f(x_i) - y_i|_{\varepsilon}, \tag{6}$$

where C is a penalty factor, n is the number of training samples, and ε is the maximum acceptable error. $|f(x_i) - y_i|_{\varepsilon}$ is described as follows, where $f(x_i)$ and y_i are the estimated and actual values of the ith sample, respectively:

$$|f(x) - y|_{\varepsilon} = \max\{0, |f(x) - y| - \varepsilon\}, \tag{7}$$

After the slack variables ξ_i and ξ_i^* are introduced, Eq. (6) can be rewritten as

$$\min \frac{1}{2}\|w\|^2 + C\sum_{i=1}^{n} (\xi_i + \xi_i^*), \tag{8}$$

The following constraints apply to the optimization problem in Eq. (8).

$$\text{subject to} = \begin{cases} y_i - f(x_i) \leqslant \varepsilon + \xi_i, \\ f(x_i) - y_i \leqslant \varepsilon + \xi_i^*, \\ \xi_i, \xi_i^* \geqslant 0, \end{cases} \tag{9}$$

2.3.3 Kernel-based support vector

Using kernel functions, SVM may be applied to nonlinear data [29]. In nonlinear data, the kernel-based K-SVM is more accurate than the conventional linear SVM shown in Eq. (8) and may be transformed into a solution to the duality problem by including the Lagrange operator (α_i and α_i^*) and kernel function ($K(x_i, x_j)$). The regression function can be rewritten as:

$$f(x) = \sum_{i=1}^{n} (\alpha_i - \alpha_i^*) K(x_i, x_j) + b, \tag{10}$$

where x_i and x_j are the input vectors in training and testing samples.

2.3.4 Decision tree

The tree structure is used in decision analysis as a way to represent decisions and their consequences. A DT[47] uses principles of logic, statistics, and AI to create a visual representation of possible outcomes from a set of potential actions.

The DT is built by selecting an appropriate attribute from the list of features for the root node as well as the decision node. The Gini index is a measurement of impurity/purity of a variable [48]. Binary divisions usually use the Gini index. When all of the data in a node belong to the same class, the node has maximal purity, and the Gini score is zero. However, if the data at a node include all possible classes, the Gini index value will be maximized. Attributes with low Gini indices are preferred if they are low or high. It is calculated using the following formula:

$$\text{Gini}(D) = 1 - \sum_{j=1}^{n} p_j^2, \tag{11}$$

where D is the data set, n is the partition, and p_j is the

probability of a partition being assigned to a specific class.

2.3.5 Random forest

RF [49] uses a large number of DTs to perform classification and regression tasks. Many DTs are generated at random from the data set in the RF method [50]. Each tree examines a fraction of the input data or features and makes a prediction based on that subset in an attempt to answer the problem at hand. The summary output (usually an average) of each tree is combined to produce the final estimate. The final RF estimate is the average of all tree results. The number of trees that are created while employing RF can vary. For simple purposes, a single RF with a small number of trees may suffice. For more complex tasks, multiple RFs may be needed. This is because the number of possible combinations for a given problem increases exponentially as the data set size grows [51].

Consider a training data set $\{x_1, x_2, x_3, ..., x_n\} \in R^{\mathrm{Lxn}}$, n is the number of samples, and L is the number of characteristics. x_i denotes the location of sample i in the space R^{Ln}. The RF ensemble in a tree k can be represented as follows:

$$f_k(x) = f(x, \theta_k), \tag{12}$$

where θ_k is a random vector.

The probability of predicting a specific class z to which a sample x belongs using the RF technique may be described by Eq. (13).

$$P(z|x) = \frac{1}{K}\sum_{k=1}^{K} P_k(z|x), \tag{13}$$

where $P_k(z|x)$ is the predicted density of the class labels of kth tree, and K is the number of trees in the forest. The definition of the decision function for the forest is as follows:

$$C(x) = \underset{j \in Z}{\operatorname{argmax}} P(j|x). \tag{14}$$

The margin function of an RF is

$$\mathrm{ml}(x,z) = P(z|x) - \max_{\substack{j \in Z \\ j \neq z}} P(j|x). \tag{15}$$

The appropriate solution is achieved if $\mathrm{ml}(x,z) > 0$.

2.3.6 Multiple linear algorithm

Linear regression is a statistical method that is used to analyze data and estimate the relationship between variables. MLR extends the basic principle of simple linear regression by exploring multiple predictor variables rather than only one. MLR is one of the most common regression techniques because it is straightforward to compute and interpret. It estimates the relationship between the response variable Y and predictor variables $x_1, x_2, \ldots, x_n$, assuming that there is a linear relationship between them [52]. The MLR is represented as follows:

$$Y = \alpha + \sum_{i=1}^{n} b_i x_i + \varepsilon, \tag{16}$$

where α is the intercept, and ε is the model error.

2.3.7 Polynomial algorithm

PR is a form of regression analysis that is used to analyze the relationship between variables by fitting a polynomial (a continuous function with a degree of three or higher) as an approximate model of the data [53]. In other words, it determines the relationship between a dependent variable and independent variables by fitting a cubic or another degree polynomial instead of linear regression. The coefficient of each independent variable gives information on how sensitive the dependent variable is to changes in that independent variable [54].

$$\hat{y} = \beta_0 + \beta_1 x + \beta_2 x^2 + \beta_3 x^3 + \cdots + \beta_k x^k, \tag{17}$$

where x denotes the input variable, $\hat{y}$ denotes the output variable, $\beta_1, \beta_2, \ldots, \beta_k$ are the coefficients, and β_0 is the intercept.

2.3.8 Naive bayes algorithm

The conditional probability of an occurrence is described by Bayes' theorem, which is based on previous knowledge of conditions that may be relevant to this case. A key assumption of Bayesian networks [55] is that the variables are conditionally independent of each other. NBC is a probabilistic technique based on Bayesian theory that makes decisions based on a simple assumption that all variables are conditionally independent of one another [56,57]. It assumes that all features are independent of each other and that each feature has an equal likelihood of occurring in any given example.

Equation (18) shows the NB method:

$$P(C_j/X) = \frac{P\left(\frac{X}{C_j}\right)P(C_j)}{P(X)}, \tag{18}$$

where $P(C_j/X)$ is the posterior probability, $P(X/C_j)$ is the likelihood, $P(C_j)$ is the class prior probability, and $P(X)$ is the predictor prior probability.

2.3.9 K-Nearest neighbor algorithm

KNN represents an obvious instance-based learning

technique [58]. The algorithm is based on the idea that two or more objects that are closer in proximity should yield a closer similarity match in most cases. The KNN classifier calculates the distance between each input item and each other object in the training set and then categorizes each object based on the objects that are closest to it [59]. The KNN decision rule is as follows:

$$f(x, C_j) = \sum_{t_i \in KNN} \mathrm{sim}(x, t_i) Z(t_i, C_j), \tag{19}$$

where x, t_i, and $\mathrm{sim}(x, t_i)$ represent the test data, training data, and the similarity between them, respectively. In addition, C_j is a potential class.

2.3.10 Light gradient boosting machine

Light gradient boosting machine (LightGBM) is a widely used algorithm in the field of machine learning (ML), known for its efficient training process and computational speed [60]. With its gradient-based optimization strategy, LightGBM rapidly converges and achieves high accuracy by iteratively improving predictions by reducing the gradient of the loss function. Its lightweight design and histogram-based feature discretization contribute to its efficiency, particularly for large-scale data sets. Additionally, LightGBM has built-in support for categorical features, which streamlines the preprocessing procedures. Regression and classification are two tasks where it has shown strong performance. LightGBM is a useful tool for ML researchers and practitioners because of its effective training and capacity for handling large data sets.

2.3.11 Categorical boosting algorithm

Categorical boosting algorithm (CatBoost) [61] has gained significant recognition and praise in the field of ML due to its outstanding ability to handle categorical features effectively. Traditional ML algorithms often require extensive preprocessing, such as one-hot encoding, to handle categorical variables. However, CatBoost offers a clear advantage by eliminating the need for such preprocessing steps. This simplifies the feature engineering process and reduces the risk of dimensionality explosion and information loss that may occur with one-hot encoding. Another feature that distinguishes CatBoost from other ML algorithms is its effective training algorithm. CatBoost achieves a balance between model performance and computational economy by combining gradient-based optimization with a symmetric leaf-wise tree growth technique. The method may create trees level by level, giving better information gain features priority, owing to the symmetric leaf-wise tree growth technique. This method expedites training and helps reduce the risk of overfitting, resulting in more reliable and accurate models. Its ability to handle categorical features effectively and its efficient training methodology contribute to its widespread adoption and its success in various real-world applications across domains such as engineering, finance, and healthcare.

3 Data collection

This section contains information about the data sets used in this study. The data sets are thoroughly detailed, the data synthesis process via TVAE is explained, and the specifics of the synthetic data set are presented.

3.1 Real-world data set with ten-variables

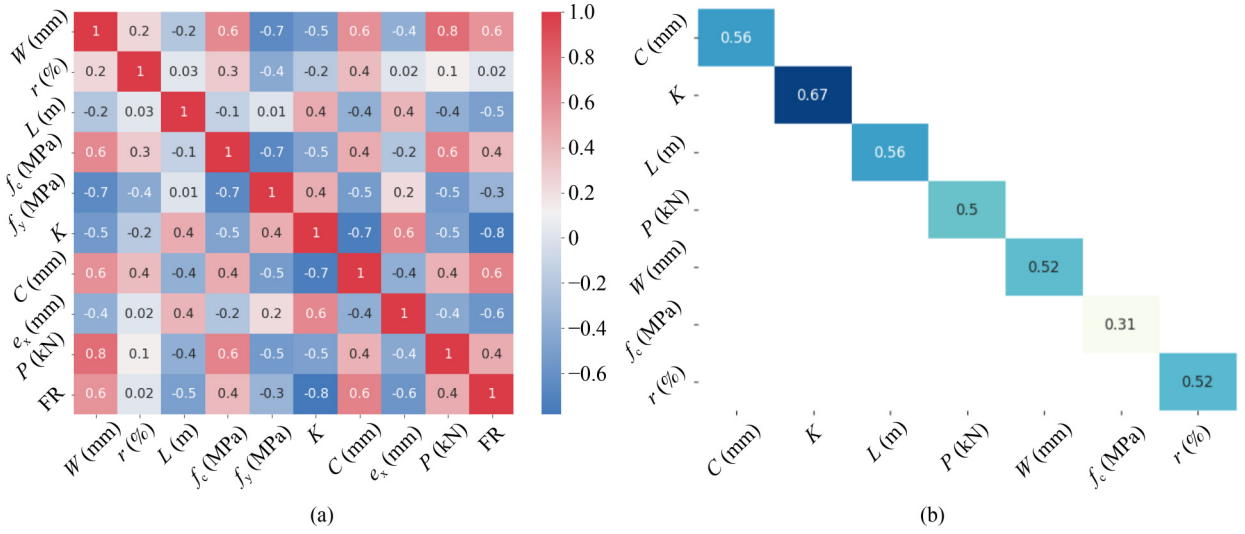
This research uses a compiled database consisting of RC columns subjected to standard fire conditions. Data was taken from open literature sources [62–74]. The following information was collected for each experiment: W is column size; r is steel reinforcement ratio; L is length, f_c is concrete compressive strength; f_y is steel yield strength; K is restraint condition (fixed-fixed = 0.5, fixed-pinned = 0.7, and pinned-pinned = 1); C is concrete cover to reinforcement; e_x is eccentricity in applied load; P is load magnitude; and FR is fire resistance. Table 1 and Fig. 1 (by means of a correlation matrix) present more insights into the compiled data set.

As can be seen, the compressive strength of the concrete, the concrete cover, the width of the concrete, and the applied load have a considerable positive correlation with the fire resistance time. In contrast, there is a negative correlation between eccentricity, restraint conditions, and fire resistance time. In general, there is a significant inverse relationship between fire resistance time and restraint conditions. For more flexible and more vulnerable columns, lower fire resistance times are commonly observed, particularly for eccentrically loaded ones. Moreover, the fire resistance time has a medium negative correlation with column length, as well as a low correlation with eccentricity and steel yield strength.

Moreover, the examination of the maximal information coefficient (MIC) values heatmap, as illustrated in Fig. 1(b), yields valuable insights into the intricate and nonlinear associations among the features within the data set. The MIC values, ranging from 0 to 1, offer a quantitative measure of the strength of association between features. A value of ‘0’ signifies no association, while a value of ‘1’ represents a perfect association. The resulting heatmap provides a visual representation of these associations, with darker shades denoting stronger positive associations and lighter shades indicating weaker or non-existent associations. Upon analyzing the MIC values, several noteworthy associations between features become apparent. The feature “W (mm)” demonstrates a moderate positive association (MIC = 0.52) with “r (%)”

Table 1 Distribution of variables of real data with ten variables

Variable	W (mm)	r (%)	L (m)	f_c (MPa)	f_y (MPa)	K	C (mm)	e_x (mm)	P (kN)	FR (min)
Min.	200.0	0.9	2.1	24.0	354.0	0.5	25.0	0.0	0.0	31.0
Max.	914.0	4.4	5.8	138.0	591.0	1.0	64.0	150.0	5373.0	636.0
Average	313.2	2.2	3.9	50.7	464.6	0.8	39.0	15.6	1316.3	167.0
St. Dv.	71.7	0.6	0.7	29.8	64.8	0.2	6.9	29.1	1176.5	112.7
Skewness	4.1	0.4	0.4	1.5	0.2	−0.1	−0.3	3.2	1.7	1.1



(a) (b)

Fig. 1 Statistical insights into the data set of real data with ten variables: (a) Pearson correlation matrix; (b) MIC Values Heatmap.

(MIC = 0.52). Similarly, "L (m)" (MIC = 0.56) and "C (mm)" (MIC = 0.56) exhibit a moderate positive association. On the other hand, "f_c (MPa)" displays a relatively weaker association (MIC = 0.31) with other features. In contrast, the feature "K" stands out with a notably high MIC value of 0.66, indicating a strong positive association with other features. Additionally, "P (kN)" (MIC = 0.50) demonstrates a moderate positive association.

3.2 Real-world data set with eight variables

As is commonly noted, the yield strength of steel falls under a range of grades, and hence this input parameter does not seem to add much to the above data set. In addition, a few columns were noted to have eccentricity, and thus, the above data set is reduced into eight variables by removing the e_x and f_y variables from the original ten-variable real-world data set. Table 2 provides a more in-depth look at the reduced data set.

Figure 2 shows the statistical insights for the reduced data set. As observed, a moderate correlation exists between most of the features and the fire resistance, generally. There is a minor correlation between the reinforcement ratio and the dependent variable. In addition, a high negative correlation exists between restriction conditions and fire resistance.

The MIC heatmap results of the data set of real data with eight variables, as depicted in Fig. 2(b), exhibit similar patterns to the MIC results of the data set of real data with ten variables presented in Fig. 1(b). These findings suggest consistent associations among the features despite the difference in the number of variables. These consistent findings across the data sets provide substantial evidence of reliable associations among the features. The similarity in the MIC results suggests a consistent underlying structure or relationship within the data set, regardless of the variation in the number of variables.

3.3 Tabular vector auto-encoder based synthetic data generation

In the present study, we devised two unique data generation procedures that exploited 144 data points from both real-world data sets to synthesize 10000 data points separately on a ten-variable real-world data set and an eight-variable real-world data set. Our approach incorporated continuous variables, which allowed us to extend the range of the original variables while maintaining their integrity. Data synthesis refers to the generation of new artificial data points based on existing data or by combining multiple data sets. Data synthesis aims to augment the training set, increase its size, or

Table 2 Distribution of variables for real data with eight variables

Variable	W (mm)	r (%)	L (m)	f_c (MPa)	K	C (mm)	P (kN)	FR
Min.	200.0	0.9	2.1	24.0	0.5	25.0	0.0	31.0
Max.	914.0	4.4	5.8	138.0	1.0	64.0	5373.0	636.0
Average	313.2	2.2	3.9	50.7	0.8	39.0	1316.3	167.0
St. Dv.	71.7	0.6	0.7	29.8	0.2	6.9	1176.5	112.7
Skewness	4.1	0.4	0.4	1.5	−0.1	−0.3	1.7	1.1

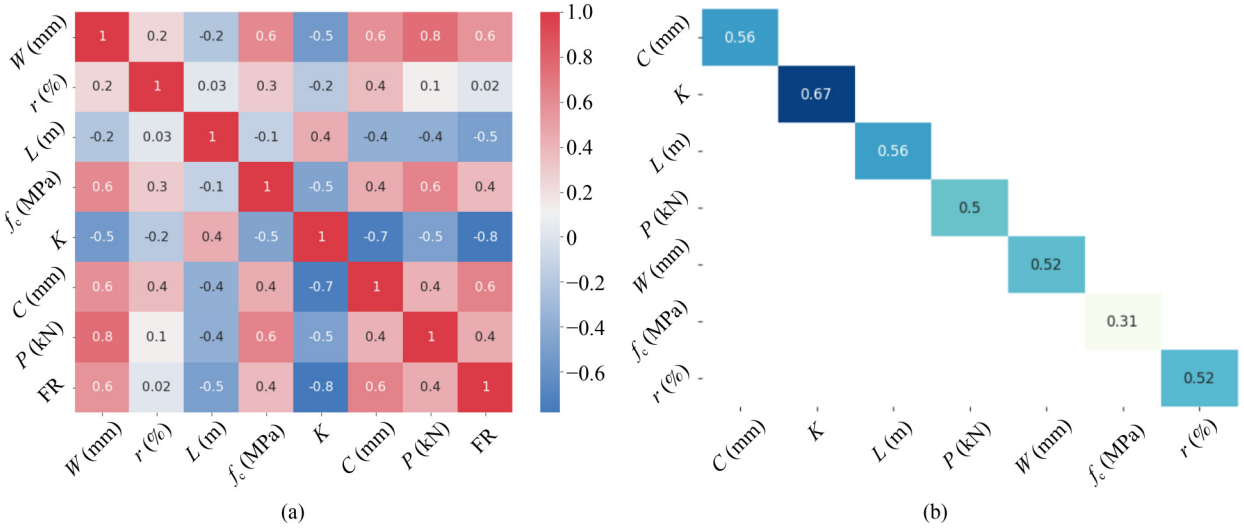


Fig. 2 Statistical insights into the data set of real data with eight variables: (a) Pearson correlation matrix; (b) MIC Values Heatmap.

introduce more diversity into the data. This augmentation endeavor aims to amplify the training set in terms of scale and diversity and imbue ML models with heightened discernment of intricate patterns, thus fostering an enhanced capacity for generalization.

Ensuring data leakage prevention necessitates adherence to the correct order of operations. Specifically, the application of data synthesis precedes the partitioning of the data set into distinct training and testing subsets. This prudent approach ensures that the artificially generated data points are not inadvertently entangled within both the training and testing sets, thereby averting any inadvertent leakage of information. Conversely, if data synthesis were to be conducted subsequent to the data set split, the resulting synthetic data points would be inherently intertwined within both the training and testing subsets. Consequently, evaluating the performance of the ML model on unseen data would become intricate, as the model would have already encountered certain instances from the testing set during the training phase. Applying data synthesis before the partitioning step allows us to expose the model to the augmented data during training, enabling the assimilation of knowledge from these synthetically generated examples. Subsequently, during the evaluation phase on the dedicated test set, the model confronts entirely novel and hitherto unseen instances, thereby furnishing a reliable and unbiased assessment of its generalization capabilities.

The following sections provide a comprehensive analysis of the implementation of synthetic data in the development of ML techniques.

3.3.1 Generating synthetic data for the ten-variable real-world data set

The method employed in this investigation involves the use of TVAE to generate 10000 synthetic data points from an original ten-variable data set comprising 144 data points. Results revealed that the statistical distribution of the generated synthetic data closely resembles that of the real data, as demonstrated in Table 3.

In addition, Fig. 3 illustrates the absolute logarithmic means and standard deviations of the synthetic and real data, while Fig. 4 showcases the comparison of cumulative sums per variable. These figures reveal that the generated synthetic data exhibit a similar range of values as the real data, a comparable log-normal distribution, and the same correlation structure. The blue lines in Fig. 3 represent 45° lines, emphasizing a direct equality relationship between the ten-variable synthetic data and ten-variable real data. It is important to note that these lines represent equality between the two data sets, rather than best-fit lines. These observations suggest that our model parameters are adequately estimated and

Table 3 Distribution of variables for synthetic data with ten variables

Variable	W (mm)	r (%)	L (m)	f_c (MPa)	f_y (MPa)	K	C (mm)	e_x (mm)	P (kN)	FR
Min.	200.0	0.9	2.1	24.0	354.0	0.5	25.0	0.0	0.0	31.0
Max.	914.0	4.4	5.8	138.0	591.0	1.0	64.0	150.0	5373.0	636.0
Average	308.1	2.2	3.9	47.8	462.5	0.8	38.9	15.1	1243.6	155.3
St. Dv.	46.2	0.5	0.7	25.4	58.0	0.2	6.4	24.9	1029.8	100.5
Skewness	1.9	0.2	0.4	1.5	0.3	−0.2	−0.4	2.8	1.7	1.1

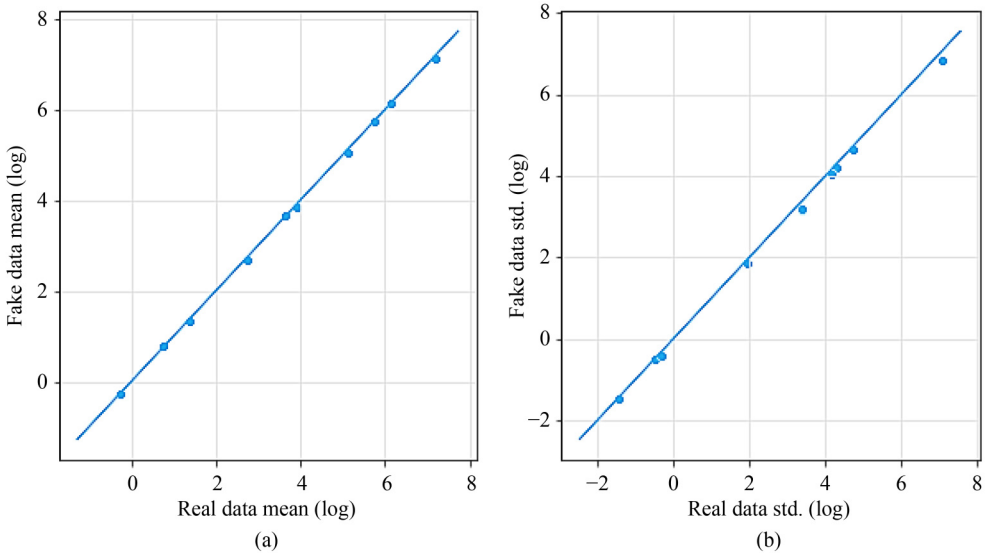


Fig. 3 Absolute logarithmic mean and std. comparison for synthetic vs. real ten-variable data: (a) logarithmic mean comparison; (b) std. comparison.

demonstrate the potential of this method as a valuable tool for generating synthetic data that emulate real-world data. Consequently, the synthetic data generated using the proposed TVAE-based approach possess statistical properties closely resembling those of real-world data sets and thus can serve as a reliable tool for validating and evaluating the performance of ML models.

3.3.2 Generating synthetic data for the eight-variable real-world data set

Using the TVAE, a synthetic data set comprising 10000 data points is generated from the original eight-variable data set consisting of 144 real-world observations. The generated synthetic data set closely approximates the real data in terms of distributions, as demonstrated in Table 4. To further examine the similarity between the synthetic and real data sets, we analyzed their absolute logarithmic mean and standard deviation, as well as their cumulative sums per variable. Figure 5 compares the absolute logarithmic mean and standard deviation of the synthetic and real data sets. The standard deviation and logarithmic means of the synthetic data set are nearly identical to those of the real data, demonstrating a high degree of similarity between the two data sets. In addition, Fig. 6 presents a comparison of the cumulative sums per variable between the synthetic and real data sets. The figure shows that the cumulative sums of the synthetic data set are quite close to those of the real data, proving the similarity between the two data sets. Notably, the differences between the real and synthetic data sets are almost negligible, implying that synthetic data may be efficiently employed for model validation and performance evaluation.

Taken together, the findings of this study provide strong evidence for the efficacy of TVAE in generating synthetic data with distributions that closely approximate those of real data. As such, this method has the potential to be a valuable tool for data augmentation and model validation in various research domains.

4 Results and discussion

This section presents the outcomes and in-depth discourse of the conducted analysis. We performed a comprehensive analysis of diverse data sets concerning the fire resistance times of RC columns to assess the effectiveness of various regression and classification models.

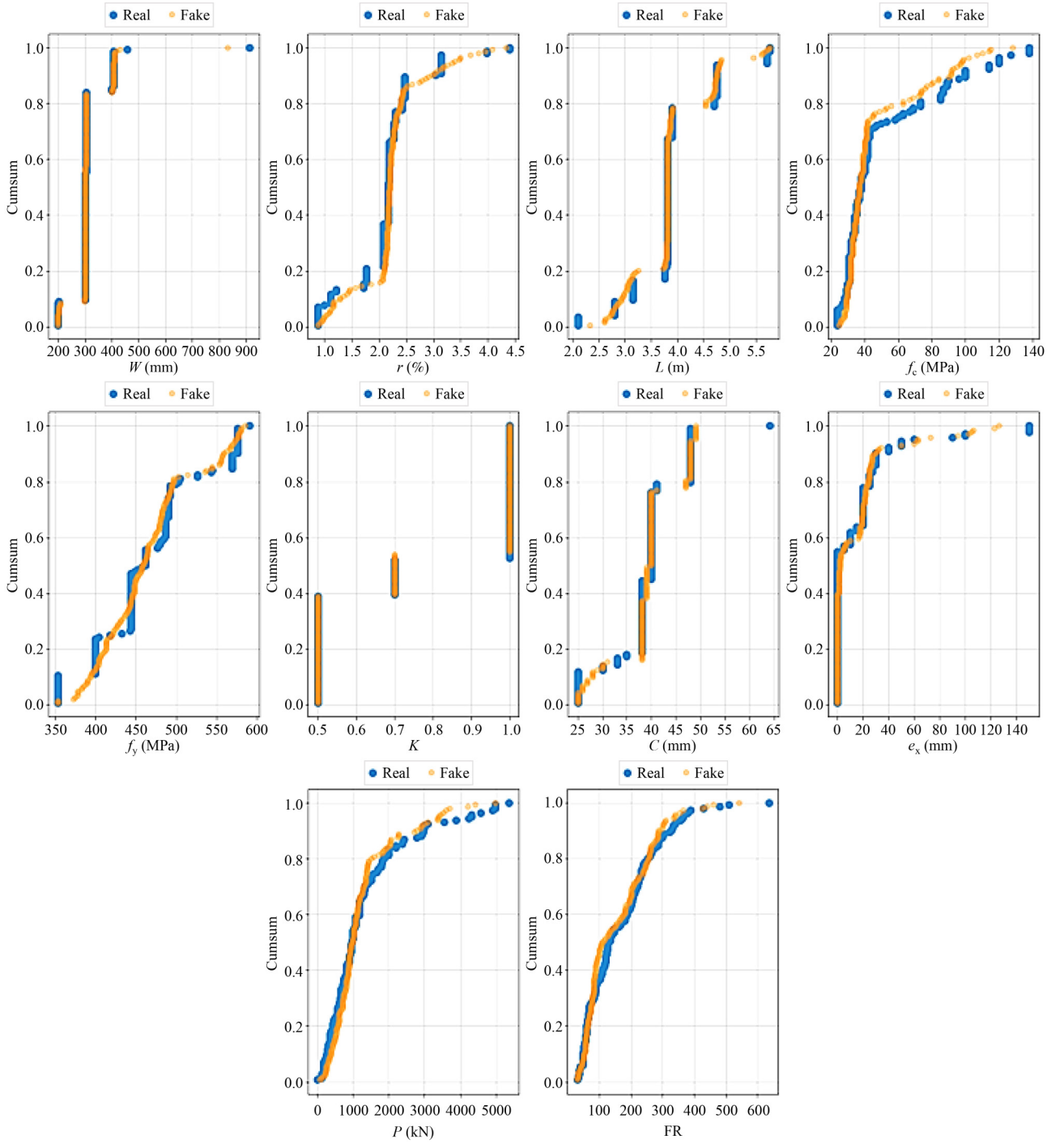


Fig. 4 Comparisons of cumulative sums per variable of ten-variable synthetic data with ten-variable real data.

Table 4 Distribution of variables for synthetic data with eight variables

Variable	W (mm)	r (%)	L (m)	f_c (MPa)	K	C (mm)	P (kN)	FR
Min	200.0	0.9	2.1	24.0	0.5	25.0	0.0	31.0
Max	914.0	4.4	5.8	138.0	1.0	56.0	5373.0	636.0
Average	314.9	2.2	3.9	48.9	0.8	38.9	1387.4	163.1
St Dv.	51.2	0.6	0.6	26.8	0.2	6.8	1129.0	103.5
Skewness	1.2	0.5	−0.2	1.2	−0.1	−0.4	1.5	0.5

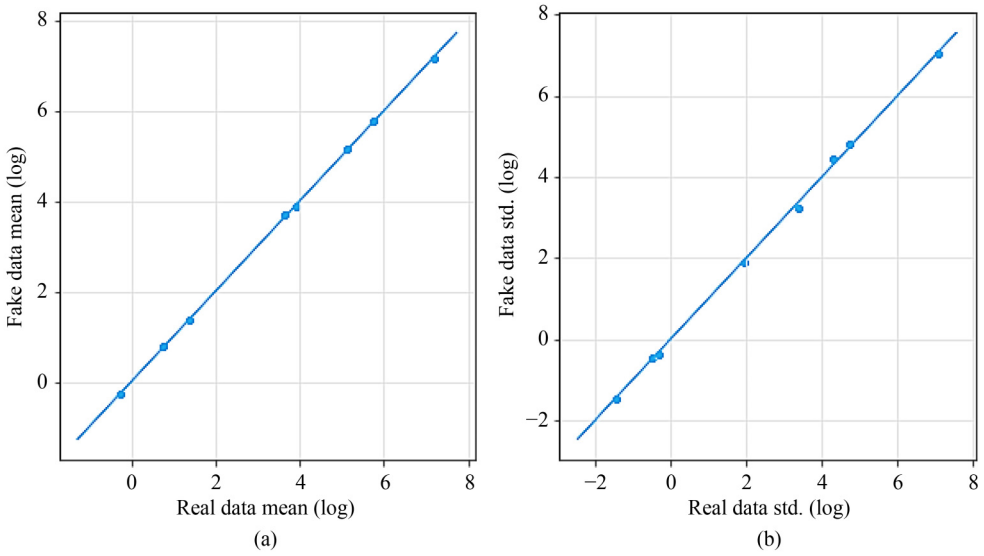


Fig. 5 Absolute logarithmic mean and std. comparison for synthetic vs. real eight-variable data: (a) logarithmic mean comparison; (b) std. comparison.

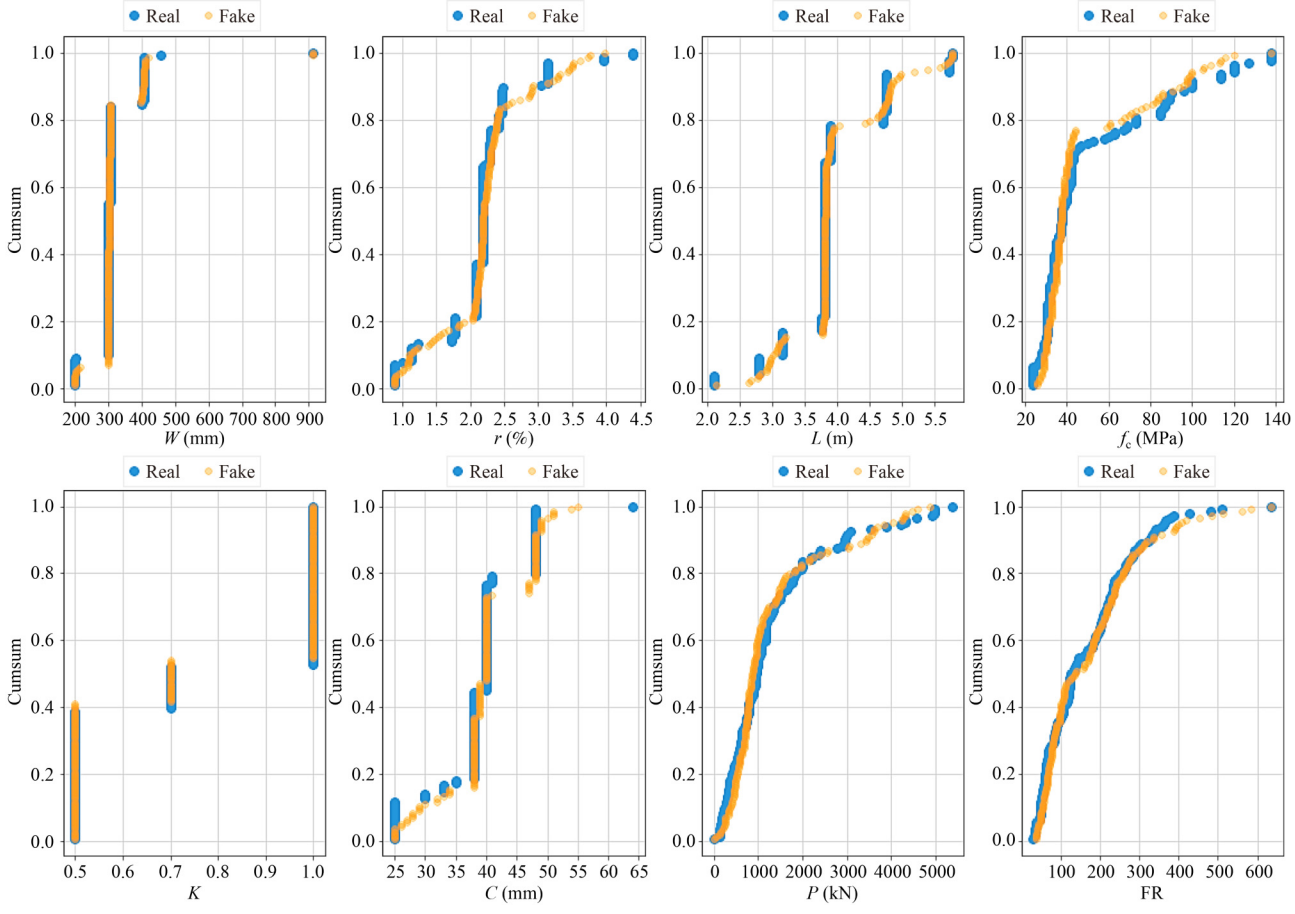


Fig. 6 Cumulative sums per variable of eight-variable synthetic data with eight-variable real data.

Hyperparameter tuning plays a crucial role in achieving optimal performance and reproducibility in ML models. In this study, we present the hyperparameter settings for regression and classification algorithms, highlighting the key choices that govern the behavior of these models. Table 5 presents the hyperparameters for regression algorithms, including the learning rate, number of estimators, and random state. These hyperparameters

significantly impact the model's ability to capture complex relationships within the data. Table 6, on the other hand, showcases the hyperparameters specific to classification algorithms, such as the choice of the penalty parameter, the number of DTs in an ensemble, and the maximum depth allowed for each tree. These hyperparameters influence the model's ability to distinguish between different classes accurately. Our hyperparameter tuning and selection procedure adheres to established best practices, ensuring a fair evaluation and comparison across the algorithms employed. The following tables and their respective descriptions delve into the specific hyperparameter values adopted for each algorithm, shedding light on the rationale behind our choices and facilitating further exploration and fine-tuning for future research.

4.1 Comparative analysis of AI-based regression prediction methods

To evaluate the efficacy of eight different regression models—namely, XGBoost, SVR, RF, DT, MLR, PR, CatBoost, and LightGBM—in predicting fire resistance times of RC columns, we conducted four separate case studies on four distinct data sets. Specifically, we employed the ten-variable real data set, the eight-variable real data set, the ten-variable synthetic data set, and the eight-variable synthetic data set. We allocated 80% of the data for training and 20% for testing purposes. Our regression results are presented in Figs. 7–10 for the ten-variable real data set, eight-variable real data set, ten-variable synthetic data set, and eight-variable synthetic data set. To conduct a comprehensive assessment, we compared the performance metrics of the regression models on the four distinct data sets listed in Table 7.

As shown in Fig. 7, the scatter plots illustrate the relative positions of the data points for each ML method in relation to the 45° line, using a real-world data set consisting of ten variables. Upon analysis, it becomes evident that the RF, XGBoost, and CatBoost models demonstrate superior performance in terms of scatter plot results. These models exhibit a tightly distributed set of points closely aligning with the 45° line. On the other hand, the DT and MLR models display a wider range of data points, positioning them slightly further away from the 45° line compared to the aforementioned methods. While these models still yield reasonable results, they do not perform as well as the RF, XGBoost, and CatBoost models.

Furthermore, the scatter plot of the SVR model demonstrates a moderately dispersed arrangement of data points, indicating a relatively closer proximity to the 45° line. Similarly, the scatter plots generated by the LightGBM model exhibit a satisfactory level of performance,

Table 5 Regression techniques and hyperparameters

Algorithms	Parameters
XgBoost	n_estimators = 34, learning_rate = 1, max_depth = 10, subsample = 0.8, random_state = 29
RF	n_estimators = 14, max_depth = None, min_samples_split = 5, random_state = 40
CatBoost	n_estimators = 25, learning_rate = 1, max_depth = 3, subsample = 0.8, random_state = 8
LGBM	n_estimators = 40, learning_rate = 0.8, max_depth = 5, subsample = 0.8, random_state = 0
PR	degree = 2
MLR	–
DT	random_state = 0, criterion = 'mse', splitter = 'best', max_depth = None, min_samples_split = 2, min_samples_leaf = 1
SVR	kernel = 'rbf', degree = 3, gamma = 'scale', C = 1.0, epsilon = 0.1

Table 6 Classification techniques and hyperparameters

Algorithms	Parameters
CatBoost	iterations = 1000, learning_rate = 0.03, depth = 6, loss_function = 'Logloss', random_strength = 1
XgBoost	booster = 'gbtree', objective = 'binary:logistic', learning_rate = 0.1, max_depth = 3, subsample = 1
LightGBM	boosting_type = 'gbdt', objective = 'binary', learning_rate = 0.1, max_depth = −1, subsample = 1, colsample_bytree = 1
DT	criterion = 'gini', splitter = 'best', max_depth = None, min_samples_split = 2
SVM	kernel = 'linear', random_state = 0
K-SVM	kernel = 'rbf', random_state = 0
RF	n_estimators = 100, criterion = 'gini', max_depth = None, min_samples_split = 2
KNN	n_neighbors = 5, weights = 'uniform'
NBC	–

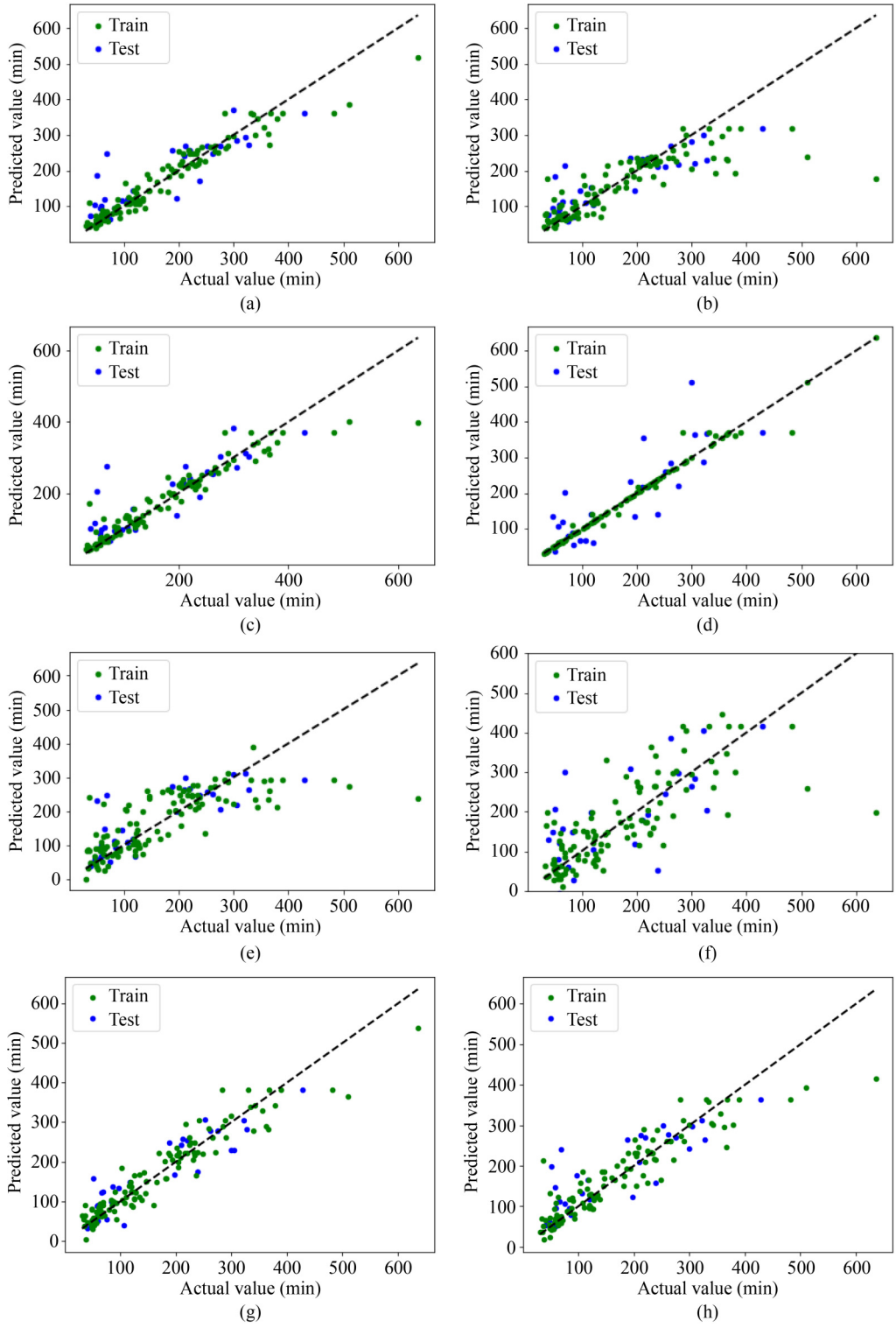


Fig. 7 Evaluation of the predictive performance of ML models on fire resistance time distribution of the ten-variable real-world data set: (a) XGBoost; (b) SVR; (c) RF; (d) DT; (e) MLR; (f) PR; (g) CatBoost; (h) LightGBM.

positioning it in a favorable position following the RF, XGBoost, and CatBoost models. In contrast, the PR model exhibits the poorest performance among the considered algorithms. This is primarily due to the presence of outliers, which significantly impact its scatter plot by deviating from the desired alignment with the 45° line.

As depicted in Fig. 8, the scatter plot analysis reveals the superior performance of specific algorithms in terms of their predictive accuracy. Among them, RF exhibited

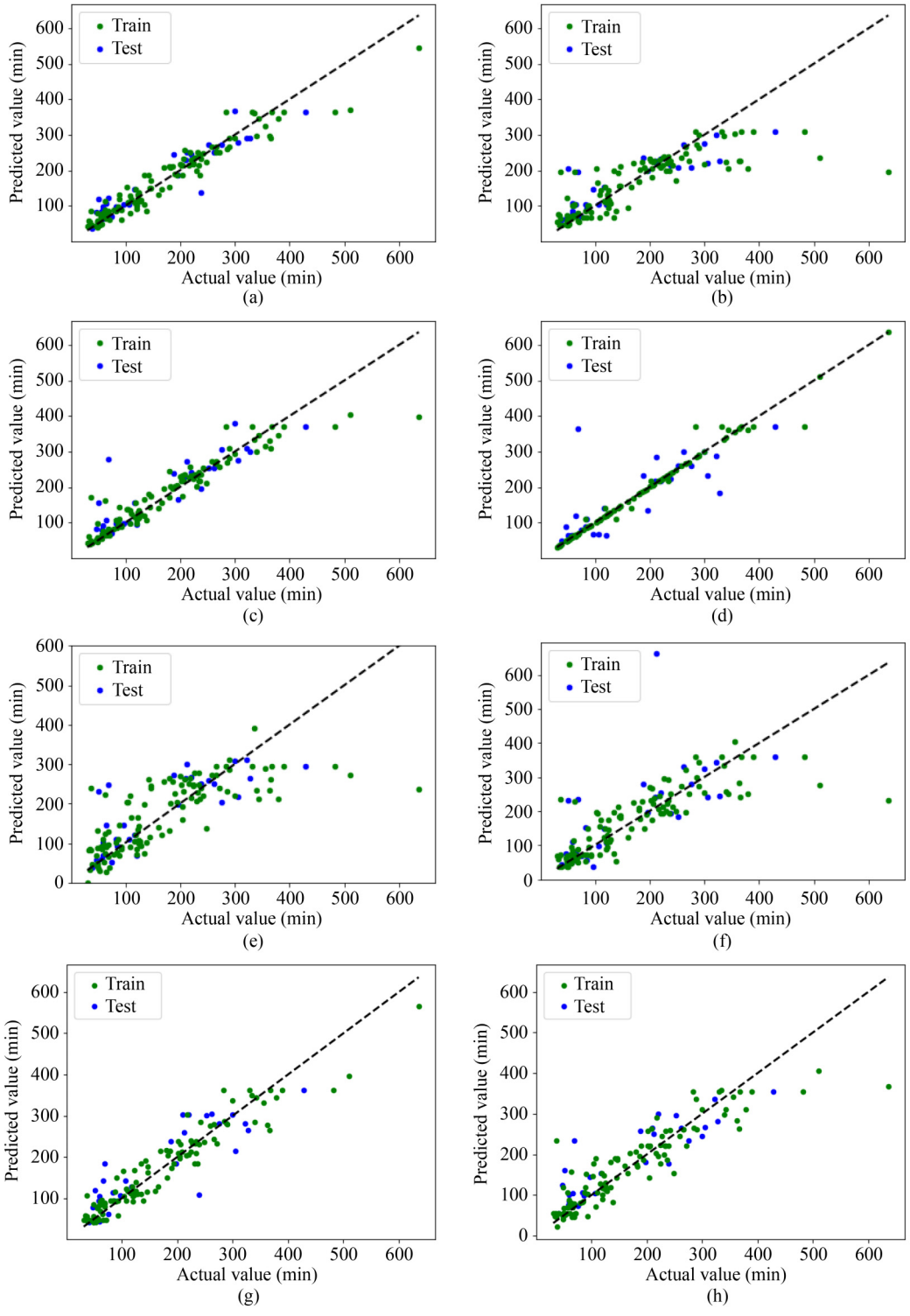


Fig. 8 Evaluation of the predictive performance of ML models on fire resistance time distribution of the eight-variable real-world data set: (a) XGBoost; (b) SVR; (c) RF; (d) DT; (e) MLR; (f) PR; (g) CatBoost; (h) LightGBM.

the most favorable results, followed closely by XGBoost, Catboost, and LightGBM. These algorithms displayed tightly distributed data points, aligning closely with the 45° line, indicating their high accuracy as predictors. Conversely, MLR, DT, and SVR exhibited a broader distribution of data points, deviating further from the 45° line, suggesting lower predictive accuracy. Notably, PR showcased the poorest performance among the algorithms studied, displaying the widest range of data points and being the farthest from the 45° line.

In the case of the ten-variable synthetic data set, the scatter plot analysis revealed the performance of the

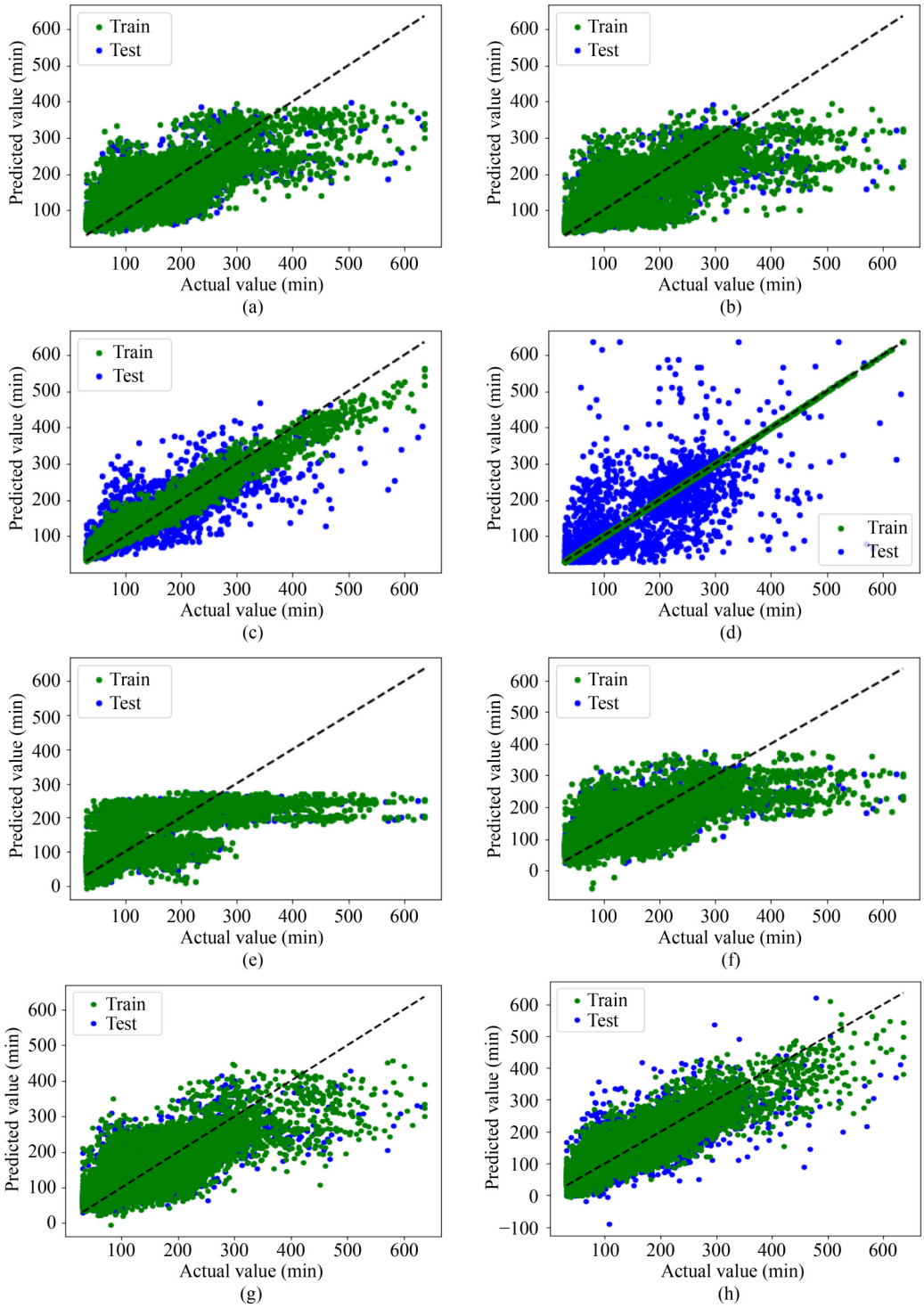


Fig. 9 Evaluation of the predictive performance of ML models on fire resistance time distribution of the ten-variable synthetic data set: (a) XGBoost; (b) SVR; (c) RF; (d) DT; (e) MLR; (f) PR; (g) CatBoost; (h) LightGBM.

various algorithms in terms of their predictive accuracy. RF emerged as the most reliable predictor, exhibiting tightly distributed data points and closely aligning with the 45° line, as depicted in Fig. 9. Catboost and LightGBM also demonstrated strong performance, following RF in proximity to the 45° line. PR, MLR, and XGBoost showed moderate performance, displaying a wider distribution of data points and deviating further from the 45° line. DT and SVR algorithms followed, indicating a lower level of predictive accuracy. Notably, the DT model was found to suffer from overfitting, compromising its performance.

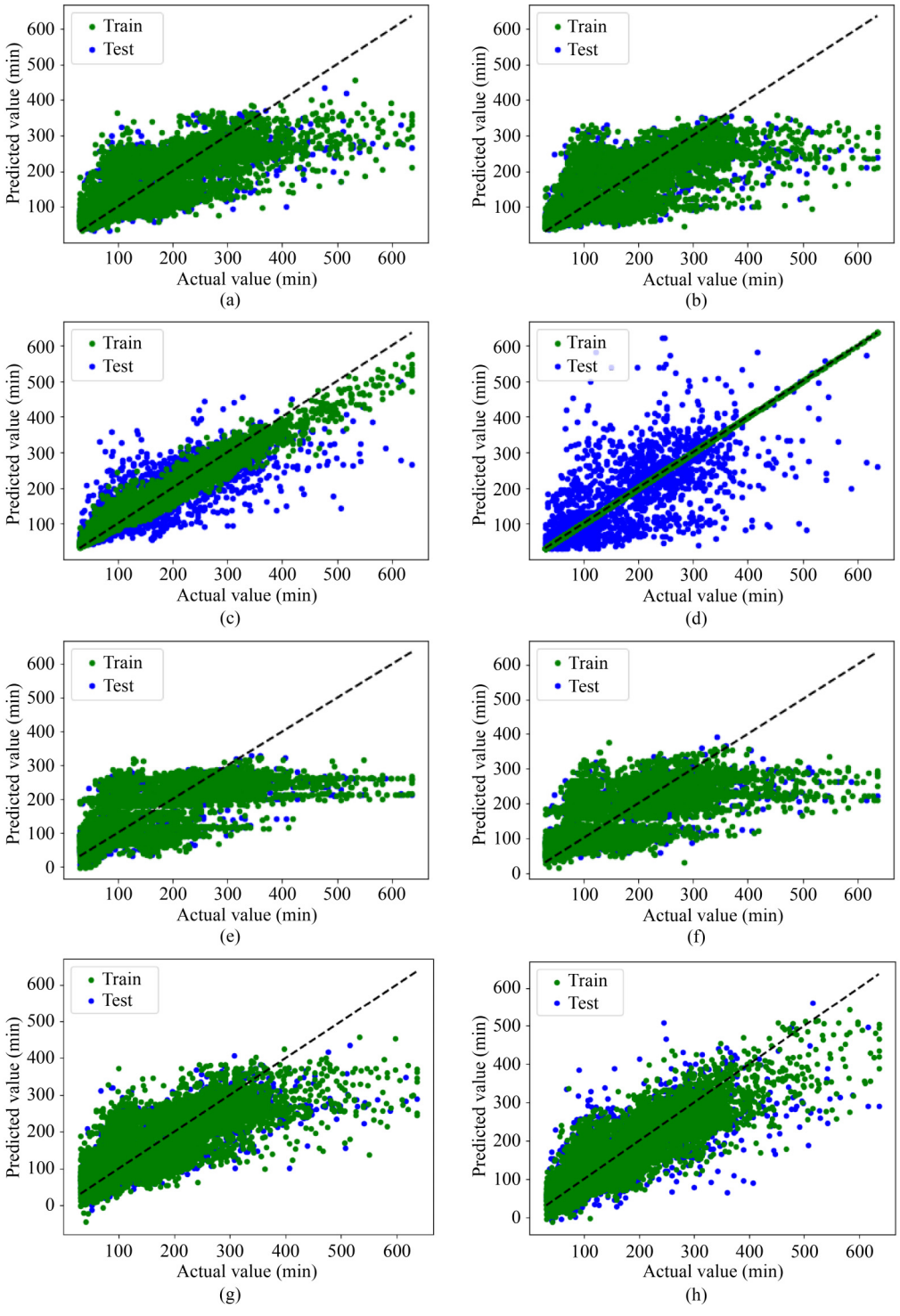


Fig. 10 Evaluation of the predictive performance of ML models on fire resistance time distribution of the eight-variable synthetic data set: (a) XGBoost; (b) SVR; (c) RF; (d) DT; (e) MLR; (f) PR; (g) CatBoost; (h) LightGBM.

Based on the analysis presented in Fig. 10, the scatter plot performance highlights the accuracy of different algorithms in predicting the eight-variable synthetic data set. RF emerged as the most accurate predictor, displaying evenly distributed data points and aligning closest to the 45° line. Catboost and LightGBM also demonstrated strong performance, following RF in proximity to the 45° line. PR, MLR, XGBoost, and SVR exhibited moderate performance, showcasing a broader distribution of data points and slightly deviating from the 45° line. DT algorithm followed, indicating a lower level of predictive accuracy and suffering from overfitting, as observed.

To evaluate the performances of the regression models, four widely used metrics, root mean square error (*RMSE*),

Table 7 Regression Model Performance Evaluation

Model	Statistical estimators	Data			
		Ten-variable real-world data set	Eight-variable real-world data set	Ten-variable synthetic data set	Eight-variable synthetic data set
XGBoost	R^2	0.71	0.87	0.62	0.61
	RMSE	57.57	38.04	62.03	64.53
	MAE	43.39	30.00	43.96	45.02
	MAPE	0.45	0.26	0.36	0.33
SVR	R^2	0.72	0.72	0.58	0.55
	RMSE	56.68	56.06	64.86	69.85
	MAE	43.97	39.88	43.39	45.96
	MAPE	0.44	0.38	0.33	0.31
RF	R^2	0.68	0.74	0.64	0.65
	RMSE	60.68	54.04	60.19	61.84
	MAE	41.39	36.20	41.65	41.83
	MAPE	0.48	0.37	0.34	0.30
DT	R^2	0.60	0.57	0.25	0.35
	RMSE	67.22	68.92	86.29	83.81
	MAE	47.25	40.59	55.08	54.46
	MAPE	0.37	0.36	0.43	0.38
MLR	R^2	0.59	0.63	0.51	0.51
	RMSE	68.05	64.82	69.87	72.99
	MAE	46.41	47.95	49.91	52.53
	MAPE	0.44	0.47	0.42	0.40
PR	R^2	−0.87	0.04	0.55	0.52
	RMSE	145.78	104.33	67.06	71.67
	MAE	99.19	60.05	47.67	50.76
	MAPE	0.93	0.51	0.40	0.38
CatBoost	R^2	0.81	0.72	0.61	0.61
	RMSE	45.97	56.06	62.79	64.83
	MAE	38.87	44.31	45.13	46.00
	MAPE	0.35	0.37	0.38	0.35
LightGBM	R^2	0.66	0.73	0.55	0.60
	RMSE	62.03	55.64	66.76	66.03
	MAE	46.48	43.11	48.09	46.13
	MAPE	0.47	0.42	0.40	0.34

mean absolute percentage error *(MAPE)*, mean absolute error (*MAE*), and coefficient of determination (R^2) were employed in this study. By employing these metrics, this study aimed to provide a comprehensive and reliable assessment of the performances of the regression models considered in this research. The parameters mentioned are reliant on the dissimilarity between predicted and actual output values, as demonstrated by Eqs. (20)–(23) ($y_{r,i}$ and $y_{p,i}$ are respectively the real and predicted values and $y_{m,i}$ is the mean of the data).

$$R^2 = 1 - \frac{\sum_i (y_{p,i} - y_{r,i})^2}{\sum_i (y_{r,i} - y_{m,i})^2}, \tag{20}$$

$$RMSE = \sqrt{\frac{1}{n}\sum_1^n (y_{p,i} - y_{r,i})^2}, \tag{21}$$

$$MAPE = \frac{1}{n}\sum_1^n \left| \frac{y_{p,i} - y_{r,i}}{y_{r,i}} \right|, \tag{22}$$

$$MAE = \frac{1}{n}\sum_{1}^{n}\left|(y_{p,i} - y_{r,i})\right|. \quad (23)$$

The models with the highest R^2 value and lowest *RMSE*, *MAE*, and *MAPE* values are generally considered to be more reliable. In Table 7, RF, Catboost, LightGBM, and XGBoost demonstrate good performance across different data sets.

RF consistently performs well across various data sets, showing high R^2 values and relatively low *RMSE*, *MAE*, and *MAPE* values. Catboost and LightGBM also exhibit favorable results, with relatively high R^2 values and competitive performance in terms of *RMSE*, *MAE*, and *MAPE*. XGBoost shows strong performance as well, particularly in the eight-variable real-world data set where it achieves an R^2 value of 0.87.

On the other hand, PR shows lower performance in real-world data sets but performs relatively better in the synthetic data set with a higher number of data points. This suggests that PR may be more suitable for data sets with a large number of observations, while its accuracy diminishes in smaller real-world data sets. The remaining regression algorithms, including MLR, DT, and SVR, generally exhibit lower accuracies compared to those of RF, Catboost, LightGBM, and XGBoost.

The performance of various ML algorithms for regression tasks was evaluated through a comparative analysis of regression models with k-fold cross-validation on multiple data sets. The results in Table 8, show that the RF algorithm consistently outperformed the other algorithms across all data sets. The RF algorithm achieved the lowest average *RMSE* value of 50.28 on the ten-variable real-world data set, followed closely by CatBoost (58.40) and XGBoost (59.83). The PR and SVR algorithms exhibited significantly higher average *RMSE* values of 144.26 and 97.54, respectively. On the eight-variable real-world data set, the RF algorithm again achieved the lowest average *RMSE* value of 41.72, followed by DT (64.15) and PR (66.46). The PR algorithm yielded the highest average *RMSE* value of 98.64 for this data set. The ten-variable synthetic data set produced a different pattern, with the RF algorithm achieving the lowest average *RMSE* value of 60.38, closely trailed by CatBoost (61.40) and LightGBM

Table 8 Comparative analysis of regression models with K-fold cross-validation: Assessing *RMSE* performance on data set

Dataset	Item	XGBoost	RF	CatBoost	LightGBM	DT	MLR	PR	SVR
Ten-variable real-world dataset	1	59.45	44.58	36.23	50.01	60.24	67.11	145.58	109.73
	2	41.67	38.02	42.68	41.54	74.11	57.61	158.11	56.68
	3	67.02	62.12	51.25	65.86	71.27	73.86	119.61	105.84
	4	53.58	66.17	83.34	92.66	46.25	85.66	178.42	104.41
	5	77.44	40.50	78.50	71.03	77.60	62.98	119.61	111.03
	Average	59.83	50.28	58.40	64.22	65.89	69.44	144.26	97.54
Eight-variable real-world dataset	1	35.28	35.00	56.06	58.41	74.27	68.74	93.87	102.05
	2	40.26	41.84	42.84	66.84	57.28	86.69	70.61	56.06
	3	62.65	38.45	55.52	55.60	65.02	73.49	123.13	104.05
	4	58.47	47.74	62.00	43.79	75.04	61.56	92.33	107.65
	5	43.00	45.56	70.78	96.09	83.22	41.81	113.24	104.66
	Average	47.93	41.72	57.44	64.15	70.96	66.46	98.64	94.90
Ten-variable synthetic dataset	1	62.03	62.83	61.79	65.24	75.28	69.71	70.05	124.82
	2	73.95	60.40	60.35	61.00	81.33	71.54	68.69	125.12
	3	79.46	58.82	62.85	62.31	82.41	69.63	65.88	123.87
	4	79.93	61.27	58.48	59.55	86.05	73.74	63.59	64.86
	5	79.95	58.60	63.51	64.93	81.79	66.94	67.25	125.21
	Average	75.06	60.38	61.40	62.61	81.37	70.31	67.09	112.78
Eight-variable synthetic dataset	1	78.86	62.45	63.72	62.42	82.42	72.24	71.18	129.58
	2	79.01	59.86	62.69	61.12	79.03	68.79	69.34	128.93
	3	64.53	63.24	64.17	64.19	83.70	71.51	68.75	69.85
	4	79.12	59.33	62.03	63.60	84.93	72.85	71.88	128.35
	5	78.32	62.67	58.00	64.17	84.43	72.60	71.83	129.24
	Average	75.97	61.51	62.12	63.10	82.90	71.60	70.60	117.19

(62.61). The SVR algorithm exhibited the highest average *RMSE* value of 112.78. Finally, on the eight-variable synthetic data set, the RF algorithm achieved the lowest average *RMSE* value of 61.51, followed by CatBoost (62.12) and LightGBM (63.10). The SVR algorithm exhibited the highest average *RMSE* value of 117.19. These results suggest that the RF algorithm is a robust and versatile ML algorithm that can be effectively used for regression tasks on a variety of data sets. The RF algorithm is typically more accurate than other ML algorithms, and it is also relatively unaffected by overfitting.

4.2 Comparative analysis of AI-based classification models

To facilitate the analysis of RC columns, we have classified them into five distinct classes based on their fire resistance duration. Specifically, Class 1 indicates a fire resistance duration of less than 60 min, Class 2 between 60 and 120 min, Class 3 between 120 and 180 min, Class 4 between 180 and 240 min, and Class 5 more than 240 min. The performance of nine different ML methods for classification, namely DT, RF, SVM, K-SVM, NBC, KNN, XgBoost, CatBoost, and LightGBM is assessed on four different data sets consisting of ten-variable and eight-variable real-world data sets, as well as ten-variable and eight-variable synthetic data sets.

In evaluating the classification algorithms, this study uses confusion matrices and their corresponding metrics. Each cell of the confusion matrix represents the number of instances in which the model correctly or incorrectly predicts a class as belonging to another class. Confusion matrices enable objective quantification of discrepancies between algorithms in assigning samples to varying groups based on external criteria. Additionally, they provide insight into the likely reasons for discrepancies between algorithms, even if different algorithms give similar results with respect to positive samples.

Confusion reports are employed to comprehensively evaluate AI models, including metrics such as accuracy, precision, recall, and F1 score. Accuracy is the most significant metric in the confusion report, indicating how well the classifier performed. It measures the ability of the model to correctly identify the predicted class by dividing the number of true positives (i.e., the correct predictions) by the total number of predictions. A high accuracy score implies that the prediction system is more likely to be correct than incorrect. Precision is the proportion of true positives out of true positives plus false positives (i.e., the incorrect predictions). It represents the number of positive class forecasts that are genuinely in the positive class. Recall measures how well the prediction system distinguishes between desired and undesired items by dividing the number of true positives by the number of true positives plus false negatives (i.e., when an item predicted as positive is actually negative). The F1 score is obtained by taking the harmonic mean of precision and recall, reflecting the model's overall performance. The higher the F1 score, the better the model's performance. The mathematical expressions defining these metrics are illustrated in Table 9.

Table 9 Evaluation of classification performance metrics

Metric	Equation
Accuracy (*ACC*)	$ACC = \frac{TP+TN}{TP+TN+FP+FN}$
F_1 score	$F_1 = \frac{2TP}{2TP+FP+FN}$
True positive rate (*TPR*) or sensitivity or recall	$TPR = \frac{TP}{TP+FN}$
True negative rate (*TNR*) or specificity	$TNR = \frac{TN}{TN+FP}$

Figure 11 and Table 10 present the confusion matrix and report for the classification algorithms applied to the ten-variable real-world data set. The confusion matrix reveals that DT and CatBoost achieve flawless results with no misclassifications in any RC column, while XgBoost, LightGBM, and RF demonstrate near-perfect results with only a few misclassifications. Table 10 displays that DT achieves 100% classification accuracy, while RF achieves 94% accuracy. The NBC algorithm obtained an accuracy of 83%, whereas SVM, K-SVM, and KNN achieved accuracies of 72%, 64%, and 58%, respectively. Moreover, DT and CatBoost demonstrated the highest precision, recall, and F1-scores (100%) for each class. XgBoost, LightGBM, and RF closely follow them with consistently high accuracy. NBC and SVM exhibit intermediate performance, while K-SVM performs relatively less well. KNN shows the poorest performance among the evaluated algorithms.

Figure 12 and Table 11 present the confusion matrix and report for the eight-variable real-world data set, respectively. Figure 12 displays the predicted and actual outcomes for each class, illustrating that the DT, CatBoost, Xgboost, and RF algorithms achieved the best performance across all classes. Notably, the DT and CatBoost algorithms achieved perfect classification accuracy of 100%, while the XgBoost and RF algorithms achieved close accuracy scores of 97% and 94%, respectively. LightGBM, although slightly lower in accuracy, still demonstrates a respectable performance with an accuracy of 89%. In contrast, the SVM, K-SVM, and KNN algorithms struggled to perform, achieving accuracy scores of 72%, 64%, and 56%, respectively. Further analysis of Table 11 highlights the superior performance of the DT and CatBoost algorithms, with the highest precision, recall, and F1-score (100%), whereas the KNN algorithm demonstrated the lowest precision, recall, and F1-score. The results obtained emphasize the importance of selecting appropriate classification

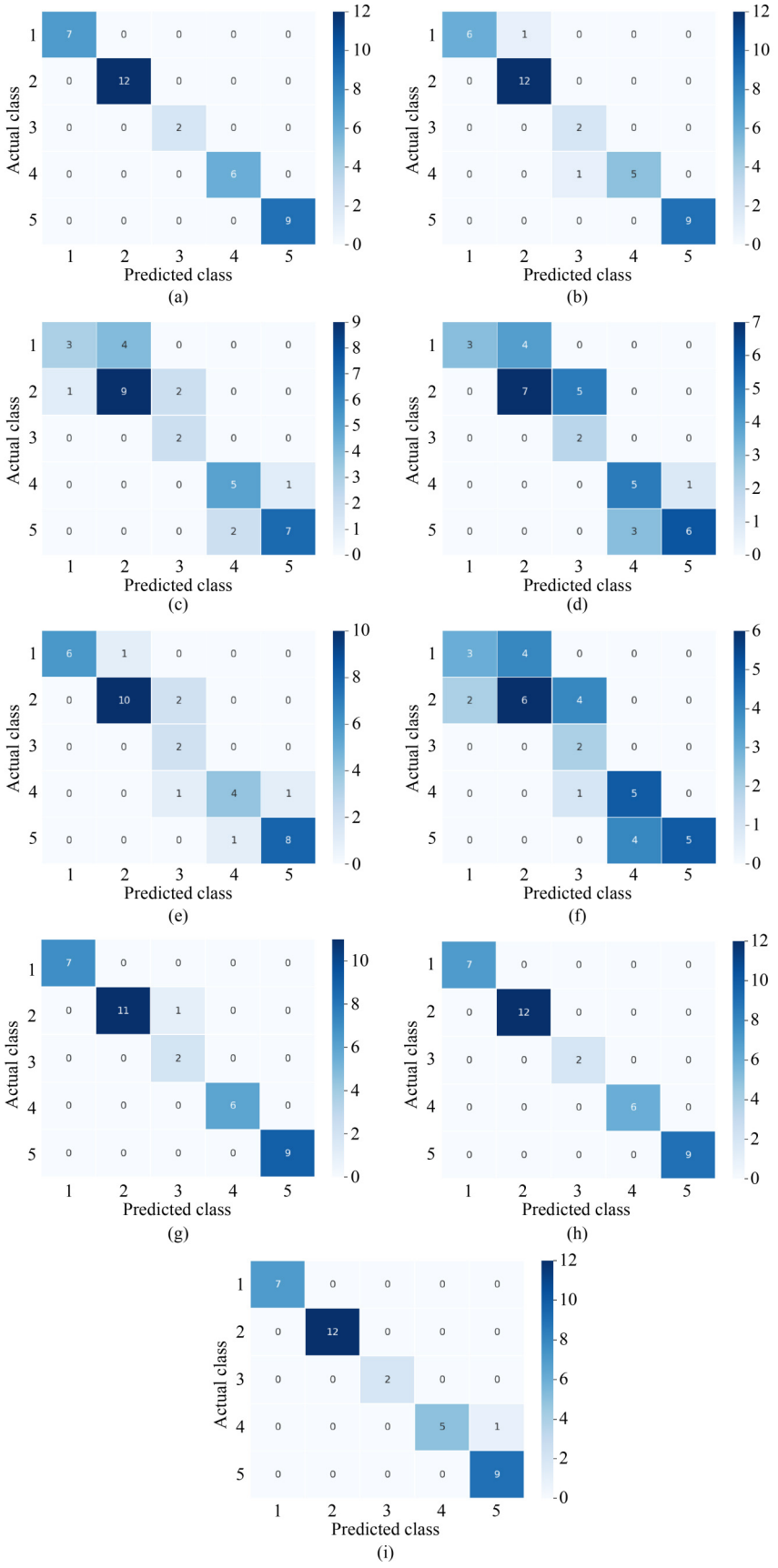


Fig. 11 Confusion matrices for classification models trained on the ten-variable real-world data set: (a) DT; (b) RF; (c) SVM; (d) K-SVM; (e) NBC; (f) KNN; (g) XgBoost; (h) CatBoost; (i) LightGBM.

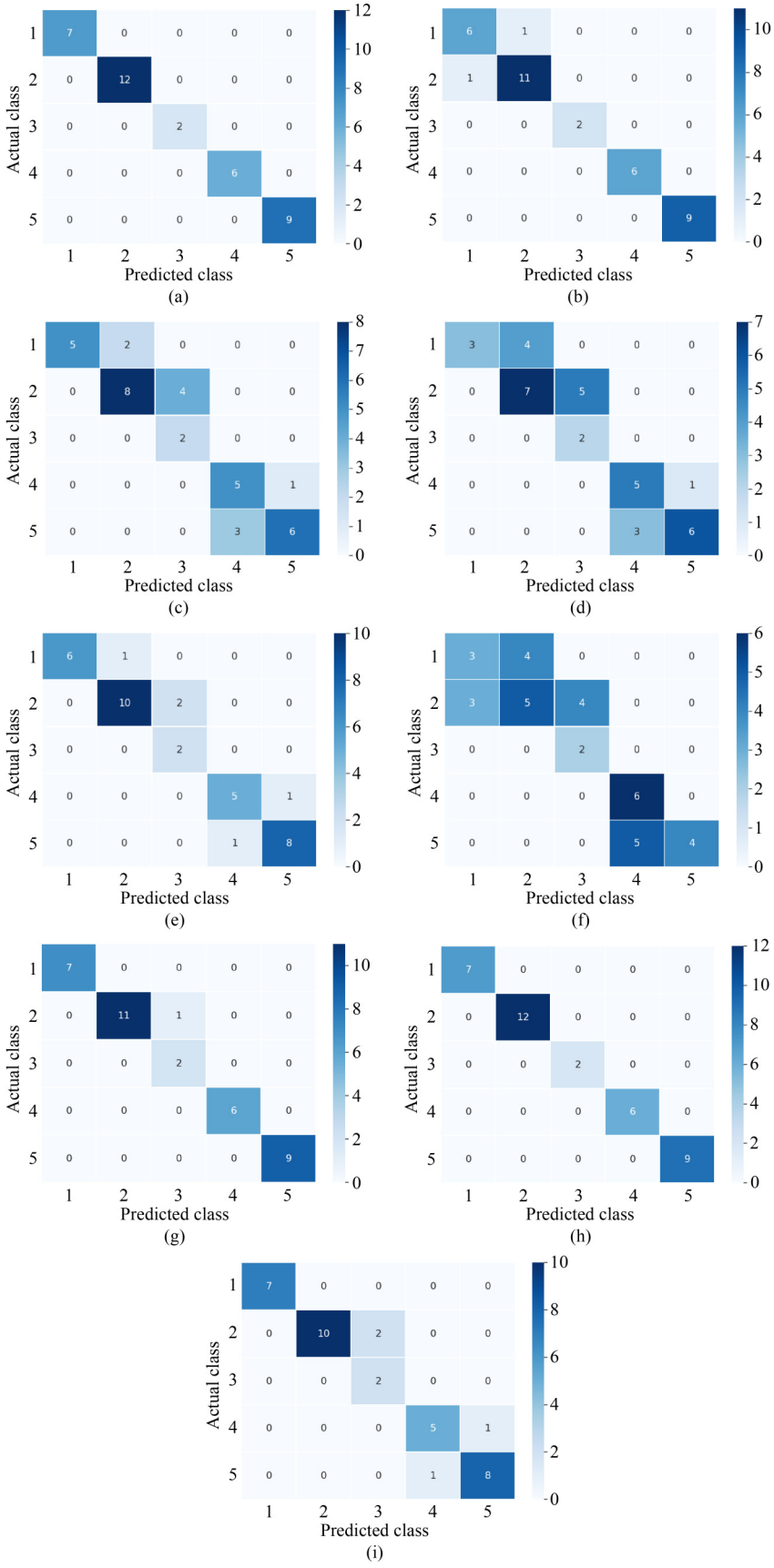


Fig. 12 Confusion matrices for classification models trained on the eight-variable real-world data set: (a) DT; (b) RF; (c) SVM; (d) K-SVM; (e) NBC; (f) KNN; (g) XgBoost; (h) CatBoost; (i) LightGBM.

Table 10 Classification performance metrics of ML models on the ten-variable real-world data set

Models	Class No.	Precision	Recall	F1-score	Accuracy
DT	1	1.00	1.00	1.00	1.00
	2	1.00	1.00	1.00	
	3	1.00	1.00	1.00	
	4	1.00	1.00	1.00	
	5	1.00	1.00	1.00	
RF	1	1.00	0.86	0.92	0.94
	2	0.92	1.00	0.96	
	3	0.67	1.00	0.8	
	4	1.00	0.83	0.91	
	5	1.00	1.00	1.00	
SVM	1	0.75	0.43	0.55	0.72
	2	0.69	0.75	0.72	
	3	0.5	1.00	0.67	
	4	0.71	0.83	0.77	
	5	0.88	0.78	0.82	
K-SVM	1	1.00	0.43	0.6	0.64
	2	0.64	0.58	0.61	
	3	0.29	1.00	0.44	
	4	0.62	0.83	0.71	
	5	0.86	0.67	0.75	
NBC	1	1.00	0.86	0.92	0.83
	2	0.91	0.83	0.87	
	3	0.4	1.00	0.57	
	4	0.8	0.67	0.73	
	5	0.89	0.89	0.89	
KNN	1	0.6	0.43	0.5	0.58
	2	0.6	0.5	0.55	
	3	0.29	1.00	0.44	
	4	0.56	0.83	0.67	
	5	1.00	0.56	0.71	
XgBoost	1	1.00	1.00	1.00	0.97
	2	1.00	0.92	0.96	
	3	0.67	1.00	0.80	
	4	1.00	1.00	1.00	
	5	1.00	1.00	1.00	
CatBoost	1	1.00	1.00	1.00	1.00
	2	1.00	1.00	1.00	
	3	1.00	1.00	1.00	
	4	1.00	1.00	1.00	
	5	1.00	1.00	1.00	
LightGBM	1	1.00	1.00	1.00	0.97
	2	1.00	1.00	1.00	
	3	1.00	1.00	1.00	
	4	1.00	0.83	0.91	
	5	0.90	1.00	0.95	

Table 11 Classification performance metrics of ML models on the eight-variable real-world data set

Models	Class No.	Precision	Recall	F1-score	Accuracy
DT	1	1.00	1.00	1.00	1.00
	2	1.00	1.00	1.00	
	3	1.00	1.00	1.00	
	4	1.00	1.00	1.00	
	5	1.00	1.00	1.00	
RF	1	0.86	0.86	0.86	0.94
	2	0.92	0.92	0.92	
	3	1.00	1.00	1.00	
	4	1.00	1.00	1.00	
	5	1.00	1.00	1.00	
SVM	1	1.00	0.71	0.83	0.72
	2	0.8	0.67	0.73	
	3	0.33	1.00	0.5	
	4	0.62	0.83	0.71	
	5	0.86	0.67	0.75	
K-SVM	1	1.00	0.43	0.6	0.64
	2	0.64	0.58	0.61	
	3	0.29	1.00	0.44	
	4	0.62	0.83	0.71	
	5	0.86	0.67	0.75	
NBC	1	1.00	0.86	0.92	0.86
	2	0.91	0.83	0.87	
	3	0.5	1.00	0.67	
	4	0.83	0.83	0.83	
	5	0.89	0.89	0.89	
KNN	1	0.5	0.43	0.46	0.56
	2	0.56	0.42	0.48	
	3	0.33	1.00	0.5	
	4	0.55	1.00	0.71	
	5	1.00	0.44	0.62	
XgBoost	1	1.00	1.00	1.00	0.97
	2	1.00	0.92	0.96	
	3	0.67	1.00	0.80	
	4	1.00	1.00	1.00	
	5	1.00	1.00	1.00	
CatBoost	1	1.00	1.00	1.00	1.00
	2	1.00	1.00	1.00	
	3	1.00	1.00	1.00	
	4	1.00	1.00	1.00	
	5	1.00	1.00	1.00	
LightGBM	1	1.00	1.00	1.00	0.89
	2	1.00	0.83	0.91	
	3	0.50	1.00	0.67	
	4	0.83	0.83	0.83	
	5	0.89	0.89	0.89	

algorithms for accurate and efficient classification tasks.

The results, presented in Fig. 13 and Table 12, demonstrate the superior performance of the DT and RF algorithms, which outperformed other ML algorithms

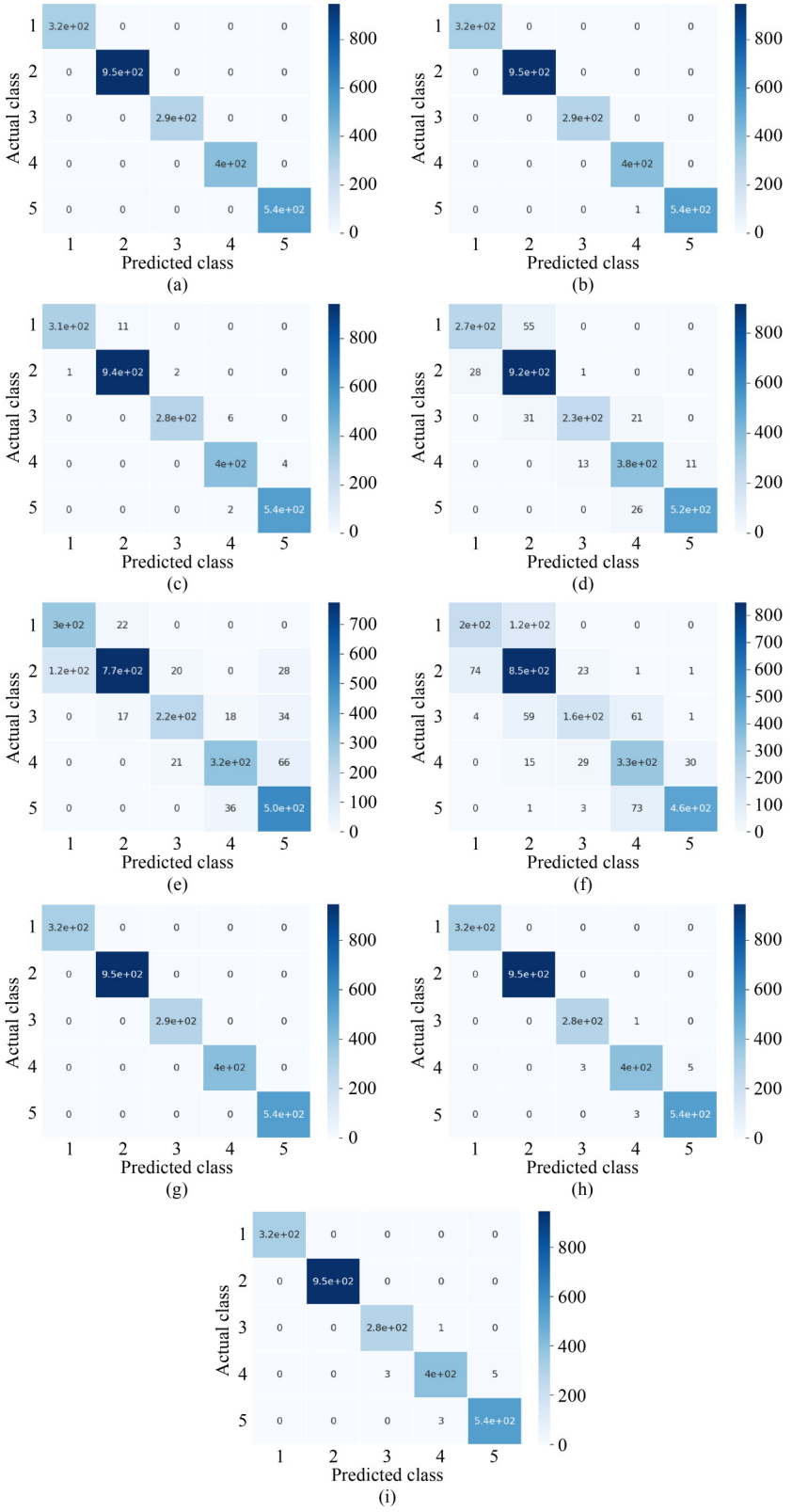


Fig. 13 Confusion matrices for classification models trained on the ten-variable synthetic data set: (a) DT; (b) RF; (c) SVM; (d) K-SVM; (e) NBC; (f) KNN; (g) XgBoost; (h) CatBoost; (i) LightGBM.

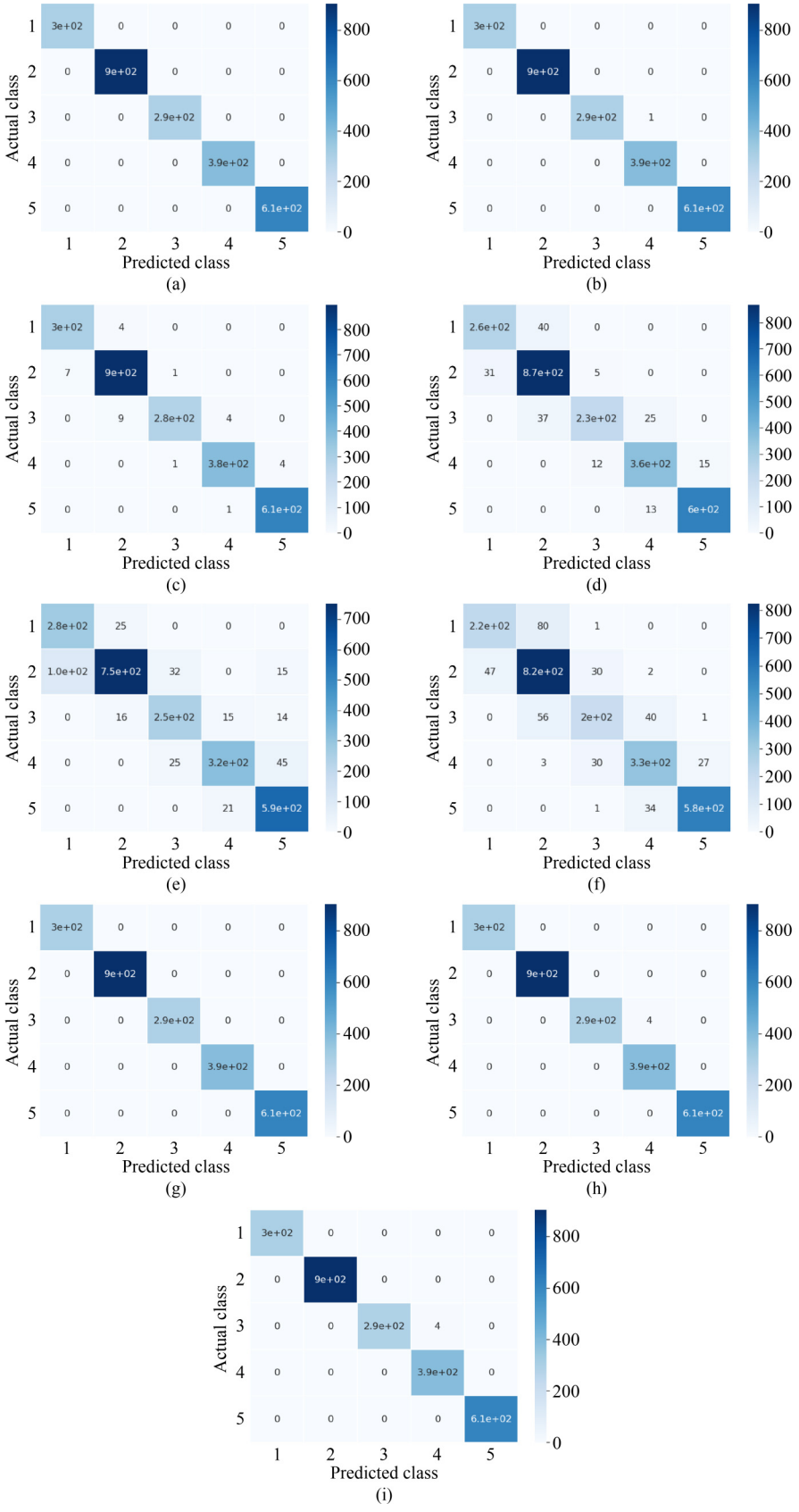


Fig. 14 Confusion matrices for classification models trained on the eight-variable synthetic data set: (a) DT; (b) RF; (c) SVM; (d) K-SVM; (e) NBC; (f) KNN; (g) XgBoost; (h) CatBoost; (i) LightGBM.

Table 12 Classification performance metrics of ML models on the ten-variable synthetic data set

Models	Class No.	Precision	Recall	F1-score	Accuracy
DT	1	1.00	1.00	1.00	1.00
	2	1.00	1.00	1.00	
	3	1.00	1.00	1.00	
	4	1.00	1.00	1.00	
	5	1.00	1.00	1.00	
RF	1	1.00	1.00	1.00	1.00
	2	1.00	1.00	1.00	
	3	1.00	1.00	1.00	
	4	1.00	1.00	1.00	
	5	1.00	1.00	1.00	
SVM	1	1.00	0.97	0.98	0.99
	2	0.99	1.00	0.99	
	3	0.99	0.98	0.99	
	4	0.98	0.99	0.99	
	5	0.99	1.00	0.99	
K-SVM	1	0.91	0.83	0.87	0.93
	2	0.91	0.97	0.94	
	3	0.94	0.82	0.88	
	4	0.89	0.94	0.91	
	5	0.98	0.95	0.97	
NBC	1	0.71	0.93	0.8	0.85
	2	0.95	0.82	0.88	
	3	0.84	0.76	0.8	
	4	0.85	0.78	0.82	
	5	0.8	0.93	0.86	
KNN	1	0.72	0.63	0.67	0.80
	2	0.81	0.9	0.85	
	3	0.75	0.56	0.64	
	4	0.71	0.82	0.76	
	5	0.94	0.86	0.89	
XgBoost	1	1.00	1.00	1.00	1.00
	2	1.00	1.00	1.00	
	3	1.00	1.00	1.00	
	4	1.00	1.00	1.00	
	5	1.00	1.00	1.00	
CatBoost	1	1.00	1.00	1.00	1.00
	2	1.00	1.00	1.00	
	3	1.00	1.00	1.00	
	4	1.00	1.00	1.00	
	5	1.00	1.00	1.00	
LightGBM	1	1.00	1.00	1.00	1.00
	2	1.00	1.00	1.00	
	3	0.99	1.00	0.99	
	4	0.99	0.98	0.99	
	5	0.99	0.99	0.99	

Table 13 Classification performance metrics of ML models on the eight-variable synthetic data set

Models	Class No.	Precision	Recall	F1-score	Accuracy
DT	1	1.00	1.00	1.00	1.00
	2	1.00	1.00	1.00	
	3	1.00	1.00	1.00	
	4	1.00	1.00	1.00	
	5	1.00	1.00	1.00	
RF	1	1.00	1.00	1.00	1.00
	2	1.00	1.00	1.00	
	3	1.00	1.00	1.00	
	4	1.00	1.00	1.00	
	5	1.00	1.00	1.00	
SVM	1	0.98	0.99	0.98	0.99
	2	0.99	0.99	0.99	
	3	0.99	0.96	0.97	
	4	0.99	0.99	0.99	
	5	0.99	1.00	1.00	
K-SVM	1	0.89	0.87	0.88	0.93
	2	0.92	0.96	0.94	
	3	0.93	0.79	0.85	
	4	0.91	0.93	0.92	
	5	0.98	0.98	0.98	
NBC	1	1.00	0.86	0.92	0.83
	2	0.91	0.83	0.87	
	3	0.4	1.00	0.57	
	4	0.8	0.67	0.73	
	5	0.89	0.89	0.89	
KNN	1	0.83	0.73	0.78	0.86
	2	0.86	0.91	0.88	
	3	0.76	0.67	0.71	
	4	0.81	0.85	0.83	
	5	0.95	0.94	0.95	
XgBoost	1	1.00	1.00	1.00	1.00
	2	1.00	1.00	1.00	
	3	1.00	1.00	1.00	
	4	1.00	1.00	1.00	
	5	1.00	1.00	1.00	
CatBoost	1	1.00	1.00	1.00	1.00
	2	1.00	1.00	1.00	
	3	1.00	0.99	0.99	
	4	0.99	1.00	0.99	
	5	1.00	1.00	1.00	
LightGBM	1	1.00	1.00	1.00	1.00
	2	1.00	1.00	1.00	
	3	1.00	0.99	0.99	
	4	0.99	1.00	0.99	
	5	1.00	1.00	1.00	

across all classes. The accuracy of all the models generally exceeded 80% in the ten-variable synthetic data set. Among the classifiers evaluated, DT, RF, XgBoost, CatBoost, and LightGBM exhibited the highest accuracy of 100%. SVM demonstrated another high performance with an accuracy of approximately 99%. Furthermore, K-SVM and NBC showed promising results, with 93% and 85% accuracy, respectively. The precision, recall, and F1-score metrics also demonstrate the superior performance of DT and RF algorithms, while KNN exhibited the lowest values. The least accurate model was observed to be KNN, with an accuracy of 80%.

The confusion matrix and report for the eight-variable synthetic data set are presented in Fig. 14 and Table 13. The results demonstrate that DT, RF, XgBoost, CatBoost, and LightGBM outperform other ML algorithms across all classes, as seen in Fig. 14. Additionally, Table 13 indicates that all ML algorithms achieve an accuracy higher than 80% on the eight-variable synthetic data set. DT, RF, XgBoost, CatBoost, and LightGBM are the most successful ML classifiers, achieving a 100% accuracy rate, followed by SVM with 99%. The K-SVM and KNN algorithms exhibit an accuracy of 93% and 86%, respectively. On the other hand, NBC has the lowest accuracy of 83%. Moreover, DT, RF, XgBoost, CatBoost, and LightGBM demonstrate the highest precision, recall, and F1-scores, while NBC and KNN display the lowest. Overall, the findings highlight that for the eight-variable synthetic data set, the DT, RF, XgBoost, CatBoost, and LightGBM algorithms are the most effective techniques, while NBC is the least effective.

5 Conclusions

This paper presents the development of a series of regression and classification ML models based on real and synthetic data. Synthetic data are obtained by using real fire tests as benchmarks for taking sample points from the real data distribution. More specifically, the synthetic data are generated utilizing DL (i.e., GAN and VAE). The outcome of this analysis can be further articulated in the following points.

1) ML cannot only predict the fire resistance phenomenon in RC columns with high accuracy but it can also be used to draw data points from the space of real data – thus effectively remedying the limited data dilemma often faced in structural fire engineering.

2) With the exponential growth of synthetic data, regression models can struggle a bit in capturing fire resistance. On the other hand, the increase in data seems to enhance the prediction capabilities of classification models.

3) The RF model consistently exhibited the best performance metrics across all regression and classification analyses. However, the SVR model performed the worst on synthetic data sets, the PR model performed the worst on real data sets, and the KNN model performed the worst on classification analyses.

Overall, this study provides a comprehensive approach to predicting fire resistance in RC columns, demonstrating the potential of ML models in utilizing both real and synthetic data. The study also highlights the limitations and challenges that need to be addressed for further research in this area. The comparative analysis of regression and classification models with k-fold cross-validation on multiple data sets provides valuable insights into the performance of various ML algorithms. These results suggest that the RF model is a robust and versatile ML algorithm that can be effectively used for both regression and classification tasks on a variety of data sets.

Open Access This article is licensed under a Creative Commons Attribution 4.0 International License (https://creativecommons.org/licenses/by/4.0/), which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit http://creativecommons.org/licenses/by/4.0/.

Competing interests The authors declare that they have no competing interests.

References

1. Abedi M, Naser M Z. RAI: Rapid, Autonomous and Intelligent machine learning approach to identify fire-vulnerable bridges. Applied Soft Computing, 2021, 113: 107896
2. Khalilpourazari S, Khalilpourazary S, Özyüksel Çiftçioğlu A, Weber G W. Designing energy-efficient high-precision multi-pass turning processes via robust optimization and artificial intelligence. Journal of Intelligent Manufacturing, 2021, 32(6): 1621−1647
3. Khalilpourazari S, Hashemi Doulabi H. A flexible robust model for blood supply chain network design problem. Annals of Operations Research, 2023, 328(1): 701−726
4. Özyüksel Çiftçioğlu A, Naser M Z. Hiding in plain sight: What can interpretable unsupervised machine learning and clustering analysis tell us about the fire behavior of reinforced concrete columns? Structures, 2022, 40: 920–935
5. Chakraborty S, Adhikari S. Machine learning based digital twin for dynamical systems with multiple time-scales. Computers & Structures, 2021, 243: 106410
6. Liang Y, Izzuddin B A. Locking-free 6-noded triangular shell

elements based on hierarchic optimisation. Finite Elements in Analysis and Design, 2022, 204: 103741
7. Bahaghighat M, Abedini F, Xin Q, Zanjireh M M, Mirjalili S. Using machine learning and computer vision to estimate the angular velocity of wind turbines in smart grids remotely. Energy Reports, 2021, 7: 8561−8576
8. Moslemi S, Mirzazadeh A, Weber G-W, Sobhanallahi M A. Integration of neural network and AP-NDEA model for performance evaluation of sustainable pharmaceutical supply chain. Opsearch. 2021: 1–42
9. Kaveh A, Biabani Hamedani K, Kamalinejad M. Improved slime mould algorithm with elitist strategy and its application to structural optimization with natural frequency constraints. Computers & Structures, 2022, 264: 106760
10. Kaveh A, Zaerreza A. Reliability-based design optimization of the frame structures using the force method and SORA-DM framework. Structures, 2022, 45: 814−827
11. Lin S, Zheng H, Han B, Li Y, Han C, Li W. Comparative performance of eight ensemble learning approaches for the development of models of slope stability prediction. Acta Geotechnica, 2022, 17(4): 1477−1502
12. Naser M Z, Alavi A H. Error metrics and performance fitness indicators for artificial intelligence and machine learning in engineering and sciences. Architecture, Structures and Construction, 2021: 1–19
13. Abueidda D W, Koric S, Sobh N A. Topology optimization of 2D structures with nonlinearities using deep learning. Computers & Structures, 2020, 237: 106283
14. Leite J P B, Topping B H V. Improved genetic operators for structural engineering optimization. Advances in Engineering Software, 1998, 29(7-9): 529−562
15. Samaniego E, Anitescu C, Goswami S, Nguyen-Thanh V M, Guo H, Hamdia K, Zhuang X, Rabczuk T. An energy approach to the solution of partial differential equations in computational mechanics via machine learning: Concepts, implementation and applications. Computer Methods in Applied Mechanics and Engineering, 2020, 362: 112790
16. Tapeh A, Naser M Z. Artificial intelligence, machine learning, and deep learning in structural engineering: A scientometrics review of trends and best practices. Archives of Computational Methods in Engineering, 2023, 30(1): 115−159
17. Guo H, Zhuang X, Fu X, Zhu Y, Rabczuk T. Physics-informed deep learning for three-dimensional transient heat transfer analysis of functionally graded materials. Computational Mechanics, 2023, 72(3): 513−524
18. Guo H, Zhuang X, Chen P, Alajlan N, Rabczuk T. Stochastic deep collocation method based on neural architecture search and transfer learning for heterogeneous porous media. Engineering with Computers, 2022, 38(6): 5173−5198
19. Zhuang X, Guo H, Alajlan N, Zhu H, Rabczuk T. Deep autoencoder based energy method for the bending, vibration, and buckling analysis of Kirchhoff plates with transfer learning. European Journal of Mechanics. A, Solids, 2021, 87: 104225
20. Guo H, Zhuang X, Rabczuk T. A deep collocation method for the bending analysis of Kirchhoff plate. 2021, arXiv: 2102.02617
21. Varone G, Ieracitano C, Çiftçioğlu A Ö, Hussain T, Gogate M, Dashtipour K, Al-Tamimi B N, Almoamari H, Akkurt I, Hussain A. A Novel Hierarchical Extreme Machine-Learning-Based Approach for Linear Attenuation Coefficient Forecasting. Entropy, 2023, 25(2): 1−19
22. de Rosa G H, Papa J P. A survey on text generation using generative adversarial networks. Pattern Recognition, 2021, 119: 108098
23. Elakkiya R, Vijayakumar P, Kumar N. An optimized generative adversarial network based continuous sign language classification. Expert Systems with Applications, 2021, 182: 115276
24. Xu L, Skoularidou M, Cuesta-Infante A, Veeramachaneni K. Modeling Tabular data using Conditional GAN. Advances in Neural Information Processing Systems, 2019, 32
25. Wang H, Wei W. Machine learning for synthetic data generation: A review. 2023, arXiv: 2302.04062
26. Shahriar S. GAN computers generate arts? A survey on visual arts, music, and literary text generation using generative adversarial network Displays, 2022, 73: 102237
27. Zhang R, Chen Z, Chen S, Zheng J, Büyüköztürk O, Sun H. Deep long short-term memory networks for nonlinear structural seismic response prediction. Computers & Structures, 2019, 220: 55−68
28. Khalilpourazari S, Hashemi Doulabi H. Designing a hybrid reinforcement learning based algorithm with application in prediction of the COVID-19 pandemic in Quebec. Annals of Operations Research, 2022, 312(2): 1261−1305
29. Ma Q, Sun C, Cui B, Jin X. A novel model for anomaly detection in network traffic based on kernel support vector machine. Computers & Security, 2021, 104: 102215
30. Naser M Z, Kodur V, Thai H T, Hawileh R, Abdalla J, Degtyarev V V. StructuresNet and FireNet: Benchmarking databases and machine learning algorithms in structural and fire engineering domains. Journal of Building Engineering, 2021, 44: 102977
31. Thai H T. Machine learning for structural engineering: A state-of-the-art review. Structures, 2022, 38: 448−491
32. Banerji S, Kodur V. Numerical model for tracing the response of Ultra-High performance concrete beams exposed to fire. Fire and Materials, 2022, 47(3): 322−340
33. McNamee R, Sjöström J, Boström L. Reduction of fire spalling of concrete with small doses of polypropylene fibres. Fire and Materials, 2021, 45(7): 943−951
34. Mohaine S, Boström L, Lion M, McNamee R, Robert F. Cross-comparison of screening tests for fire spalling of concrete. Fire and Materials, 2021, 45(7): 929−942
35. Van Coile R, Hopkin D, Elhami-Khorasani N, Gernay T. Demonstrating adequate safety for a concrete column exposed to fire, using probabilistic methods. Fire and Materials, 2021, 45(7): 918−928
36. Goodfellow I. NIPS 2016 Tutorial: Generative Adversarial Networks. 2017, arXiv: 1701.00160
37. Hukkelås H, Mester R, Lindseth F. DeepPrivacy: A generative adversarial network for face anonymization. In: 14th International Symposium on Visual Computing. Cham: Springer International Publishing, 2019: 565–578
38. Goodfellow I J, Pouget-Abadie J, Mirza M, Xu B, Warde-Farley D, Ozair S, Courville A, Bengio Y. Generative adversarial networks. Communications of the ACM, 2020, 63(11): 139−144

39. Kullback S, Leibler R A. On Information and sufficiency. Annals of Mathematical Statistics, 1951, 22(1): 79−86
40. Chen T, Guestrin C. XGBoost: A scalable tree boosting system. In: Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. New York, NY: Association for Computing Machinery, 2016: 785–794
41. Feng D C, Wang W J, Mangalathu S, Hu G, Wu T. Implementing ensemble learning methods to predict the shear strength of RC deep beams with/without web reinforcements. Engineering Structures, 2021, 235: 111979
42. Nguyen H, Vu T, Vo T P, Thai H T. Efficient machine learning models for prediction of concrete strengths. Construction & Building Materials, 2021, 266: 120950
43. Nguyen-Sy T, Wakim J, To Q D, Vu M N, Nguyen T D, Nguyen T T. Predicting the compressive strength of concrete from its compositions and age using the extreme gradient boosting method. Construction & Building Materials, 2020, 260: 119757
44. Wang Y, Sun S, Chen X, Zeng X, Kong Y, Chen J, Guo Y, Wang T. Short-term load forecasting of industrial customers based on SVMD and XGBoost. International Journal of Electrical Power & Energy Systems, 2021, 129: 106830
45. Cortes C, Vapnik V. Support-vector networks. Machine Learning, 1995, 20(3): 273−297
46. Smola A J, Scholkopf B. A tutorial on support vector regression. Statistics and Computing, 2004, 14(3): 199−222
47. Quinlan J R. Induction of decision trees. Machine Learning, 1986, 1(1): 81−106
48. Shorabeh S N, Samany N N, Minaei F, Firozjaei H K, Homaee M, Boloorani A D. A decision model based on decision tree and particle swarm optimization algorithms to identify optimal locations for solar power plants construction in Iran. Renewable Energy, 2022, 187: 56−67
49. Breiman L. Random Forests. Machine Learning, 2001, 45(1): 5−32
50. Lin W, Wu Z, Lin L, Wen A, Li J. An Ensemble Random Forest Algorithm for insurance Big Data analysis. IEEE Access: Practical Innovations, Open Solutions, 2017, 5: 16568−16575
51. Harrison J W, Lucius M A, Farrell J L, Eichler L W, Relyea R A. Prediction of stream nitrogen and phosphorus concentrations from high-frequency sensors using Random Forests Regression. Science of the Total Environment, 2021, 763: 143005
52. Júnior A M G, Silva V V R, Baccarini L M R, Mendes L F S. The design of multiple linear regression models using a genetic algorithm to diagnose initial short-circuit faults in 3-phase induction motors. Applied Soft Computing, 2018, 63: 50−58
53. Bradley R A, Srivastava S S. Correlation in polynomial regression. American Statistician, 1979, 33(1): 11−14
54. Ostertagová E. Modelling using polynomial regression. Procedia Engineering, 2012, 48: 500−506
55. Bayes T. An essay towards solving a problem in the doctrine of chances. Philosophical Transactions of the Royal Society of London, 1763, 53: 370−418
56. Khajenezhad A, Bashiri M A, Beigy H. A distributed density estimation algorithm and its application to naive Bayes classification. Applied Soft Computing, 2021, 98: 106837
57. Farid D M, Zhang L, Rahman C M, Hossain M A, Strachan R. Hybrid decision tree and naïve Bayes classifiers for multi-class classification tasks. Expert Systems with Applications, 2014, 41(4): 1937−1946
58. Fix E, Hodges J L. Discriminatory analysis. Nonparametric discrimination: Consistency properties. International Statistical Review/Revue Internationale de Statistique, 1989, 57(3): 238−247
59. Pandya D H, Upadhyay S H, Harsha S P. Fault diagnosis of rolling element bearing with intrinsic mode function of acoustic emission data using APF-KNN. Expert Systems with Applications, 2013, 40(10): 4137−4145
60. Ke G, Meng Q, Finley T, Wang T, Chen W, Ma W, Q Ye, TY Liu. LightGBM: A highly efficient gradient boosting decision tree. In: Proceedings of the 31st International Conference on Neural Information Processing Systems. Red Hook, NY: Curran Associates Inc., 2017: 3149–3157
61. Dorogush A V, Ershov V, Gulin A. CatBoost: Gradient boosting with categorical features support. 2018, arXiv:1810.11363
62. Naser M Z. Heuristic machine cognition to predict fire-induced spalling and fire resistance of concrete structures. Automation in Construction, 2019, 106: 102916
63. Hertz K D D. Limits of spalling of fire-exposed concrete. Fire Safety Journal, 2003, 38(2): 103−116
64. Shah A H, Sharma U K. Fire resistance and spalling performance of confined concrete columns. Construction & Building Materials, 2017, 156: 161−174
65. Kodur V, Cheng F, Wang T, Latour J, Leroux P. Fire Resistance of High-Performance Concrete Columns. Ottawa: National Research Council Canada, 2001
66. Klingsch E W H. Explosive spalling of concrete in fire. Dissertation for the Doctoral Degree. Gifhorn: ETH Zurich, 2014
67. Kodur V, McGrath R, Leroux P, Latour J. Experimental studies for evaluating the fire endurance of high-strength concrete columns. National Research Council Canada, Internal Report, 2005, 197
68. Liu J C C, Tan K H, Yao Y. A new perspective on nature of fire-induced spalling in concrete. Construction & Building Materials, 2018, 184: 581−590
69. Phan L T, Carino N J. Fire Performance of High Strength Concrete: Research Needs. Advanced Technology in Structural Engineering. Reston, VA: American Society of Civil Engineers, 2000, 1–8
70. Raut N, Kodur V. Response of reinforced concrete columns under fire-induced biaxial bending. ACI Structural Journal, 2011, 108(5).
71. Harmathy T Z. Effect of mositure on the fire endurance of building elements. ASTM Special Technical Publication, 1965, 385: 74−95
72. Bažant Z P, Kaplan M F, Bazant Z P. Concrete at High Temperatures: Material Properties and Mathematical Models. London: Addison-Wesley, 1996
73. Ulm F J, Coussy O, Bažant Z P. The “Chunnel” fire. I: Chemoplastic softening in rapidly heated concrete. Journal of Engineering Mechanics, 1999, 125(3): 272−282
74. Song T Y, Han L H, Tao Z. Structural behavior of SRC beam-to-column joints subjected to simulated fire including cooling phase. Journal of Structural Engineering, 2015, 141(9): 04014234