Polymer Degradation and Stability 245 (2026) 111857

Contents lists available at ScienceDirect

Polymer Degradation and Stability

journal homepage: www.journals.elsevier.com/polymer-degradation-and-stability

Methods for predicting the fire behaviour of fibre reinforced thermoset composites

Ákos Pomázi [a,b,c], Gergely Magyar [d], Andrea Toldy [a,b,*]

[a] Budapest University of Technology and Economics, Faculty of Mechanical Engineering, Department of Polymer Engineering, Műegyetem rkp. 3., H-1111 Budapest, Hungary
[b] MTA-BME Lendület Sustainable Polymers Research Group, Budapest University of Technology and Economics, Műegyetem rkp. 3., H-1111 Budapest, Hungary
[c] HUN-REN–BME Research Group for Composite Science and Technology, Budapest University of Technology and Economics, Műegyetem rkp. 3., H-1111 Budapest, Hungary
[d] Budapest University of Technology and Economics, Faculty of Mechanical Engineering, Department of Manufacturing Science and Engineering, Műegyetem rkp. 3., Budapest H-1111, Hungary

ARTICLE INFO

Keywords:
Prediction of flammability
Epoxy resin composites
Chemical structure
Machine learning algorithms

ABSTRACT

Destructive tests are typically used to evaluate the fire performance of polymers and their composites, implying high material costs and long testing times. Developing numerical models to predict flammability requires advanced mathematical expertise, IT resources, and realistic input parameters. In this study, we aimed to predict the key flammability parameters based on the chemical structure of the resin matrices and fibre content of composites, providing a potential alternative to costly experimental methods. We employed Random Forest Classifier (RFC), XGBoost algorithms, and an artificial neural network (ANN) model to predict key combustion parameters: peak heat release rate (pHRR), time to ignition (TTI), total heat release (THR) and the char residue (CR) solely based on chemical structure of the epoxy matrix and fibre content of the composite. After making the predictions, we assessed the performance of the models using consistent statistical indicators (mean absolute error (MAE), mean square error (MSE), and the determination parameter (R^2)).

1. Introduction

1.1. The application of artificial intelligence in the prediction of flammability

Large-scale measurements to investigate the fire behaviour of polymers require sophisticated equipment and, apart from their considerable material costs, are certainly not non-destructive. Likewise, developing numerical models requires a high level of mathematical knowledge and outstanding IT and software resources. In addition, developing a realistic numerical model to determine fire performance requires realistic input parameters. However, such data are often difficult to obtain due to the complexity of combustion conditions. For this reason, holistic methods and approximations are frequently used, which are only valid under certain constraints [1]. Given these obstacles, it is worth exploring other approaches. Artificial intelligence can overcome this issue and is particularly suitable for solving problems with a large number of variables (parameters) or where the relationships between the input parameters and the expected outputs are unknown. This property is also beneficial for determining combustion parameters [1,2].

Artificial intelligence (AI) is a branch of computer science dedicated to building systems capable of performing tasks that typically require human intelligence, for example, learning, reasoning, pattern recognition, sequence planning, perception, and problem-solving. Machine learning (ML) is only a subset of artificial intelligence, using mathematical algorithms and a statistical approach to perform the learning process without explicit instructional programming [3]. The main machine learning algorithms can be grouped according to Fig. 1. To apply supervised learning techniques, we must know the input parameters (x_1, x_2, …) and the output (y). The algorithm's task, in this case, is to determine the mapping function describing the relationship between them. This is based on the recognition and application of the relationship patterns. Algorithms for the determination of fire characteristics are usually selected from this group. Supervised learning can be further

* Corresponding author at: Budapest University of Technology and Economics, Faculty of Mechanical Engineering, Department of Polymer Engineering, Műegyetem rkp. 3., H-1111 Budapest, Hungary.
E-mail address: atoldy@edu.bme.hu (A. Toldy).

https://doi.org/10.1016/j.polymdegradstab.2025.111857
Received 24 July 2025; Received in revised form 17 November 2025; Accepted 10 December 2025
Available online 15 December 2025

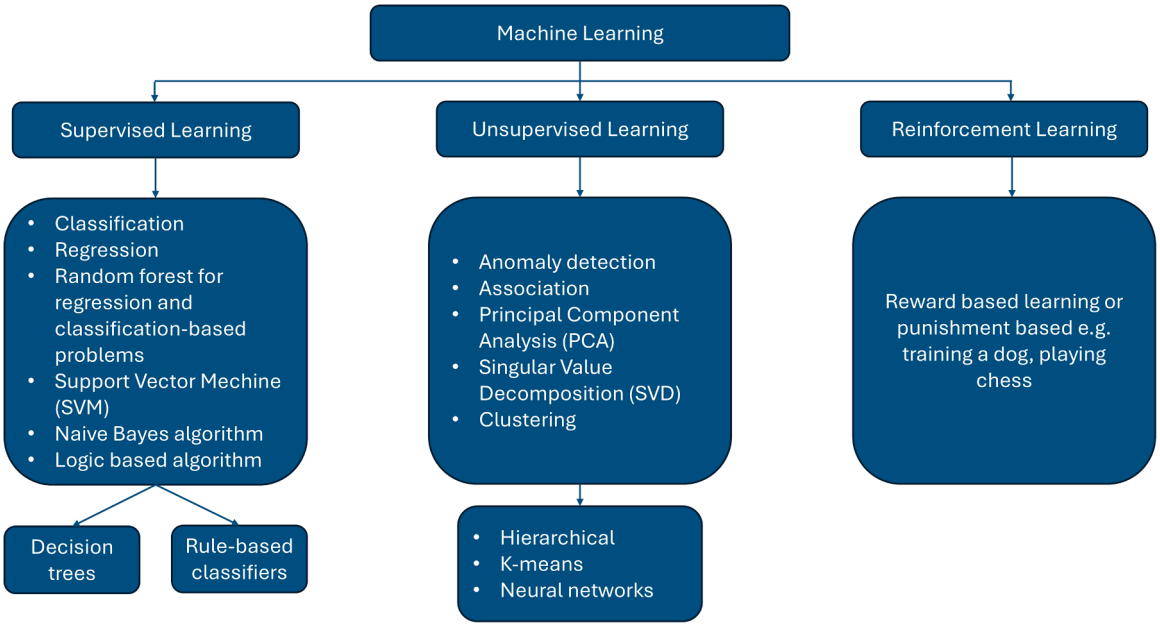

Fig. 1. Classification of machine learning algorithms (based on [4]).

divided into two main branches: regression and classification. In regression, the output is an actual number; in the latter, it is a categorical value. Related algorithms often work for both cases [4]. One of the most important first steps in most learning algorithms is partitioning the available input sets (x_1, x_2, …) and outputs (y) into training and test groups. The ideal quantitative ratio is around 80 %−20 %.

In this chapter, we will discuss algorithms for supervised learning regression problems since they are typically used for the prior estimation of combustion parameters.

1.2. The base of machine learning: the linear regression

In order to understand machine learning predictions in the case of continuous factors like flammability parameters, it is necessary to first understand the mathematical foundation of the methods. Linear regression provides the basis for supervised machine learning algorithms. More complex, computationally intensive learning algorithms, such as neural networks, are also built from variations of linear regression.

Using a line, simple linear regression describes the relationship between an input parameter (x) and an output (y). The model can be extended to the multiple linear regression (MLR) form (1), where the output is defined as a function of several independent variables [5].

$$y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \ldots + \beta_n x_n \quad (1)$$

where β_0 is the value where the function intersects the y-axis, x_i is the input parameter of index i, y is the output, and β_i is the weight of the parameter of index i.

The error of the mapping function is assumed to be independent of x, and the mean of the sign deviations is zero. The least squares method can estimate the error, and in writing the function, we aim to minimise this error.

1.2.1. Multivariate linear regression (MLR) in combustion parameter prediction

In calculating the limiting oxygen index (LOI), Van Krevelen's empirical formula typically fails to capture the complex chemical processes in the gas phase of halogen or phosphorus-containing polymers. This is addressed by the research of Wang et al. [6], in which the effect of the flame retardant groups is investigated in both the gas and condensed phases, emphasising that both cases need to be considered at all times. Besides the structural design, the results of small-scale tests and LOI values are known. In their artificial intelligence model, the first MLR algorithm calculated the contribution of each group to the charring in the condensed phase. The obtained value was then used as input by the second MLR algorithm together with the other original parameters, and finally, the LOI values were given as output (2).

$$LOI = 20.95 + \sum_i C_i N_i + \sum_i G_i N_i \quad (2)$$

where 20.95 % is the oxygen concentration in air, N_i is the number of occurrences of index group i, C_i is the regression coefficient of index group i in the condensed phase, and G_i is the regression coefficient of index group i in the gas phase.

The training results showed an accuracy of 89.8 %, and the test results showed an accuracy of 83.8 % with R^2, which was tested on 20 different polymer materials.

In addition, in 2021, Nguyen et al. [7] estimated the HRR parameter of composites with a phenolic resin matrix, glass fibre reinforcement, and phosphorus-based flame retardants using the MLR algorithm. The weight distribution of the components was used as input. The R^2 test was used to produce a value of 0.309 for test values, and the standard error of the fit was found to be 15.97 kW/m^2.

1.3. Support vector machines for regression

Support Vector Regression (SVR) is a machine learning method applied to regression tasks as an extension of Support Vector Machines (SVM) classification technology [8]. Its goal is to find the function or hyperplane that best fits the available data to minimise errors while staying within a certain tolerance. SVR can also handle nonlinear relationships if a suitable kernel function is used. Basically, SVR is a method for predicting continuous values, such as prices, temperatures, or other numerical data points. SVR has also been used to predict the properties of polymeric systems [8], for example, the fire performance: Zhang et al. [9] estimated the flame retardancy index (FRI) from THR, TTI, and pHRR using SVR. They produced R^2 values of 55 % for training data and 58 % for test data.

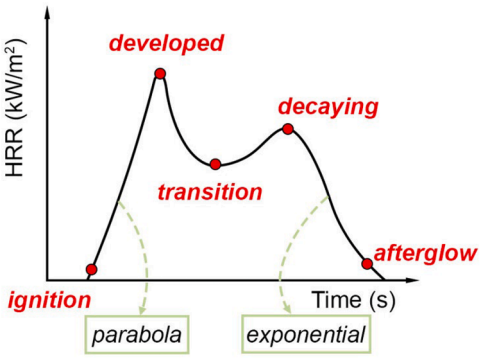

Fig. 2. Characteristics of the heat release rate curve [10].

1.4. Decision trees

While regression models fit a function to a dataset, other algorithms utilize a branching logic that more closely mimics human decision-making. Decision trees are based on simple binary trees. The algorithm sorts the training data into leaves according to the values of the input parameters. The data is assigned to one or another branch based on the conditions of each node. The algorithm tries all possible combinations of conditions and chooses the one that produces the cleanest clusters, i.e. similar outputs are placed on the same leaf. After training, the unknown data is also assigned to one of the leaves based on its inputs under the same node conditions, and its output is the average of the values there or the most frequent value [4]. More precisely, the output for regression tasks is the mean of the sample values in the leaf node, whereas for classification tasks, it is the mode (the most frequent class).

The Random Forest (RF) algorithm combines several decision trees on the same database, and the unknown output is the average of the estimations of each tree. Combining several algorithms gives a better result as it is not sensitive to individual errors. However, avoiding a strong correlation between trees is essential and can be done in several ways. To achieve this, the model employs Bootstrap Aggregating (bagging), an ensemble method where each tree is trained on a different dataset created (bootstrapping), then introduces additional randomness by also considering only a random subset of the input variables at each split (random feature selection) [3].

Multiple decision trees can be combined using a method called gradient boosting, which builds the trees one after the other, taking into account the errors of the previous instance. Usually, the most efficient type is XGBoost (XGB).

1.4.1. Random forest classifier (RFC) in the prediction of fire performance

Xiao, Hobson et al. published an article in 2023 [10] where they used a Random Forest Classifier (RFC) to predict the heat release rate (HRR) curve over time of metal hydroxide-containing flame retardant polymers. In their model, 9 RFC algorithms were coupled such that the following algorithm received the output of the previous one as the input parameter. They used classification instead of regression because the HRR curve values are usually error-laden, and they only tried to determine the characteristics of the curve. The 9 parameters estimated were the characteristic features of the curve: the type of curve, the ignition time, the time and HRR of fully developed combustion (developed), the time and HRR of the transition point (transition), the initial time and HRR of decaying and the afterglow (afterglow), see Fig. 2.

The validity of each algorithm was checked by an R^2 test, which produced values between 0.83 and 0.89 for the test case and around 1 for the training set.

Phan et al. [11] also used a Random Forest Classifier-based method to predict the flammability index (FI) and cone calorimetry results (time to ignition, maximum heat release rate, total smoke release and fire growth rate) of mostly thermoplastic polymers, although their model seemed to be promising for other polymer systems too. Besides the limited experimental data, they used generated synthetic polymers using Synthetic Data Vault (SDV), an open-source Python library and polymer descriptors generated by RDKit, an open-source toolkit for cheminformatics. They also developed the POLYCOMPRED (POLYmer COMposite Properties PREDiction) module, integrated into a cloud-based MatVerse platform, to create a user-friendly site for flammability prediction. Their results showed the robustness and applicability of their method with high R^2 values for both the training and testing datasets. They stated that the SDV method significantly improves the prediction of FI and cone calorimetry results, although their POLYCOMPRED module also needs further development by extending the database and using multiple model options for prediction (e.g. Gradient Boosting or neural networks).

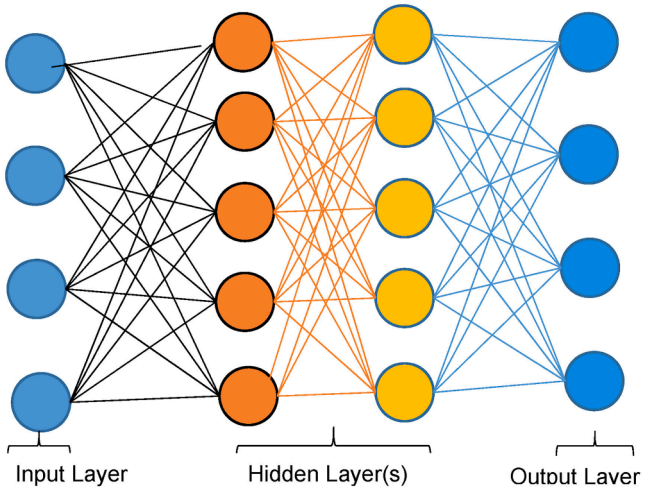

Fig. 3. The structure of an artificial neural network [12].

1.4.2. XGBoost in the prediction of fire performance

Zhang et al. [9] also applied the XGB algorithm to predict the fire retardancy index. They used the structural data of the polymer and the flame retardant as input. They obtained R^2 values of 97 % for training data, 81 % for test data and 93 % overall.

1.5. Artificial neural networks

In contrast to the more rigid structure of regression and decision tree models, deep learning models (e.g. artificial neural networks) offer a more flexible architecture that is capable of modeling highly non-linear relationships. Artificial neural networks (ANNs) are based on artificial neurons, which are similar in structure to the human brain. The algorithm is trained on a training set of data, and the effectiveness of the training can be checked on the test set. The neurons of artificial neural networks are composed of layers, which can be divided into three categories: input layer, hidden layer(s), and output layer, see Fig. 3. In the supervised case, the database contains real inputs and outputs, and the algorithm maps the patterns between them using the hidden layers. Each neuron-neuron connection has a weight and a bias [12]. During training, the algorithm changes these values. To ensure that adjusting weights and biases does not cause a large change in the algorithm's operation, nonlinear transfer functions are used as activation functions (typically the sigmoid function) that determine the output.

1.5.1. Feedforward neural network (FNN) in the prediction of fire performance

In their study, Toldy and Pomázi [13] used an artificial neural network to predict the time to ignition (TTI), peak heat release rate (pHRR), total heat release (THR), and char residue (CR) values of an

epoxy resin containing phosphorus-based additive flame retardants. Their artificial intelligence model was a feedforward neural network (FNN), which, unlike a recurrent neural network (RNN), involves a one-way flow of information between neurons. Their inputs were the structural parameters of the matrix resin, the crosslinking agent and the flame retardants, small-scale thermal and flammability measurement results, and molecular structure characteristics such as the mass ratios of individual atoms and the ratios of aliphatic, cycloaliphatic, and aromatic units. On 65 % of the estimated data, the average absolute error was around 7 %. Therefore, the model was suitable for prediction, although the accuracy could be improved by using a bigger training dataset.

Bifulco et al. [14] also developed a feedforward neural network-based model to predict the heat release capacity of hybrid magnesium hydroxide ($Mg(OH)_2$)-epoxy nanocomposites. Their input layer consisted of the molecular weight, the van der Waals volume, the molar volume, the density, the glass transition temperature, total heat release and char yield values from pyrolysis combustion flow calorimetry (PCFC) measurements, etc. After the hyperparameter optimisation, they confirmed that their algorithm had good predictive capability according to their mean absolute error (MAE) and root mean square error (RMSE).

1.5.2. Self-regulating deep neural network (SDNN) in the prediction of fire performance

Yan et al. [15] used an artificial neural network to predict LOI, pHRR and THR values of epoxy resins. The first step was identifying the repeating units in the chemical structure, which was helped by Substructure or Morgan fingerprinting. The prediction was generated by a self-controlled, self-enforced deep neural network (SDNN). The model consisted of two ANN algorithms, with the first's output receiving the second's input parameters. In addition, the substructures mapped at the beginning were included in the input of both algorithms, weighted by the molar mass of the structures. This arrangement together gave the self-adjusting character. The algorithm had a fit error of 3 % and a prediction error of 17 %. The R^2 test showed a fitting accuracy of 87 % for pHRR test data, 86 % for LOI data and 85 % for THR data.

In our previous study [13], we developed an artificial neural network (ANN)-based model to predict the flammability of epoxy resins using their chemical structure and small-scale thermal and flammability test results. This study builds upon that work by further developing the prediction model, relying solely on the chemical structure (ratio of different atoms, aliphatic, cycloaliphatic, aromatic structures) and, for composites, the fibre content, which significantly impacts the fire performance [16,17]. While numerous existing studies focus on pure polymers or thermoplastic systems, this work aims to explore the effectiveness of machine learning algorithms in predicting the fire performance of the more complex system of fibre reinforced thermoset composites, with specific emphasis on the feasibility of predictions based solely on chemical structure and compositional parameters. We employed Random Forest Regressor (RFR), XGBoost algorithms, and an ANN to predict key combustion parameters: peak heat release rate (pHRR), time to ignition (TTI), total heat release (THR) and the char residue (CR). After making the predictions, we assessed the performance of the models using consistent statistical indicators. For both XGBoost and RFR algorithms, we also identified the most critical parameters for each output. The predicted parameters were validated against real-time mass loss type cone calorimetry (MLC) test results of flame retarded, carbon fibre-reinforced epoxy composites to provide a potential platform for easier, faster and reliable material development in the future [18,19].

2. Materials and methods

2.1. Materials

A pentaerythritol-based tetrafunctional epoxy resin was used as the base resin to produce the composites for validation. The main component is the tetraglycidyl ether of pentaerythritol (PER, IPOX MR 3016; Ipox Chemicals Ltd., Budapest, Hungary; viscosity: 0.9–1.2 Pa·s at 25 °C; density: 1.24 g/cm^3 at 25 °C; epoxy equivalent weight: 156–170 g/eq). The crosslinking agent was a cycloaliphatic amine (IPOX MH 3122; Ipox Chemicals Ltd.; Budapest, Hungary; main component: 3,3′-dimethyl-4,4′-diaminocyclohexylmethane; viscosity: 80–120 mPa·s at 25 °C; density: 0.944 g/cm^3 at 25 °C; amine hydrogen equivalent: 60 g/eq).

Zoltek PX35FBUD300 type unidirectional carbon fibre plies were used as fibre reinforcement in the composites using $[0]_5$ layup. This type of carbon fabric consists of Panex 35 50k rovings (fibre diameter: 7.2 μm) with an areal weight of 300 g/m^2 (supplier: Zoltek Zrt, Nyergesújfalu, Hungary).

We applied resorcinol bis(diphenyl phosphate) (RDP) (supplier: ICL Industrial Products (Beer Sheva, Israel), trade name: Fyrolflex RDP, P content: 10.7 %) as a flame retardant additive in different phosphorus (P)-contents in the composites used for validation. This flame retardant has a well-known chemical structure, favouring the prediction method.

2.2. Methods

2.2.1. Sample preparation

Composite samples were made by wet compression moulding (hand lamination followed by hot pressing) containing 0, 1, 2 and 3 mass % phosphorus content from RDP, respectively. Hand lamination was carried out in a press mould, and each fibre reinforcement layer was impregnated separately using a brush. After we reached the required $[0]_5$ layup, the mould was hot pressed in a T30 type hydraulic press (Metal Fluid Engineering s.r.l., Verdello Zingonia, Italy) using 200 bar hydraulic pressure (which is approx. 28 bar on the composite plate). The curing cycle contained two isothermal steps: 1 h at 80 °C followed by 1 h at 100 °C (according to previous thermal characterisation of the epoxy resin[20]). The weight of the dry fibre sheets and the prepared composites was measured to calculate the fibre content of the composites.

2.2.2. Characterisation of fire behaviour

Mass loss type cone calorimetry (MLC) tests were performed with an instrument made by FTT Inc. (East Grinstead, UK) according to the ISO 13927 standard method. 100 mm x 100 mm x 2 mm specimens were ignited and exposed to a constant heat flux of 50 kW/m^2. Heat release values and mass reduction were recorded during burning.

2.2.3. Algorithms

In this study, we applied three selected machine learning algorithms to predict four different combustion parameters: XGBoost, Random Forest Regressor (RFR), and an Artificial Neural Network (ANN). The predicted parameters were the peak heat release rate (pHRR), total heat release (THR), time to ignition (TTI), and char residue (CR). The algorithms were trained using a dataset compiled from data obtained through mass loss type cone calorimetry (MLC) experiments. We aimed to assess how accurately these models could predict the parameters when only the chemical structure and the fibre content were used as input parameters.

The machine learning models were implemented in Python using Jupyter Notebooks. After data processing, we developed the models and determined the key hyperparameters influencing their performance and also optimizing the performance of the models using GridSearch optimization technique. We identified the most critical parameters for decision-making for each output by calculating the feature importance values for all models. Finally, we calculated the performance metrics for the entire dataset and the predicted set, including the mean absolute error (MAE), the mean square error (MSE), and the determination coefficient (R^2).

3. Development of the algorithms

We used different learning algorithms with diverse approaches and mathematical relationships to estimate the flammability characteristics. However, their implementation follows similar steps in each case. This section describes the process, covering the environment used, the data required for learning, the model building and its evaluation. During the development of the learning algorithms, we used the Python programming language and Jupyter Notebooks, which provide an interactive development environment, allowing for quick code testing and visualisation of the results. We used several useful libraries for data processing and modelling: Pandas for spreadsheet data management, Scikit-learn for basic machine learning tools, and TensorFlow for creating deep learning models.

3.1. Creating the database

Previous results and publications of the research group were used to create the database [21], which is essential for a successful model. The database concluded 75 samples. The 15 input parameters were: the epoxy ratio [%], the hardener ratio [%], the ratio of the first flame retardant (FR1) [%], the ratio of the second flame retardant (FR2) [%], the ratio of carbon (C) atoms [%], the ratio of hydrogen (H) atoms [%], the ratio of oxygen (O) atoms [%], the ratio of nitrogen (N) atoms[%], the ratio of phosphorus (P) atoms [%], the ratio of aliphatic structures [%], the ratio of cycloaliphatic structures [%], the ratio of aromatic structures [%], the heat flux during MLC [kW/m^2], and the fibre content [%]. The study aimed to predict the flammability from the chemical structure and the fibre content. The chemical structure was based on the calculation of the atomic percentages (C, H, O, N, P %) and the structural moiety percentages (aliphatic, cycloaliphatic, aromatic %).

An important first step in data preparation is to separate the test and training sets (20 %−80 %), which was performed using a standard random split method. As the validation dataset contained only four samples, additional test data was needed to evaluate the models accurately. To prepare the inputs properly, we scaled them proportionally so that each feature has the same effect on the model. This way, different scales do not distort the learning process. For this purpose, we used standardisation, transforming the data so that its distribution is nonzero (mean = 0) and has a standard deviation of unity (standard deviation = 1). The features are numerical, and we indexed each input by sample name and source. As outputs, we determined the peak heat release rate (pHRR), total heat release (THR), time to ignition (TTI) and char residue (CR) of the sample.

3.2. Modelling and optimisation

After data preparation, the next step was modelling, which involved general parameter setting, training, and fine-tuning of the chosen algorithms. The training is done on the training set; the algorithm also sees these outputs and uses them to refine the weights to achieve a more accurate estimation. The model predicts the test and validation (prepared samples) outputs. Finally, we visualised the data and evaluated the model by summarising the training, test and validation sets. Separating the test values was necessary to get a more accurate picture of the model's performance. We also examined the fit of the test and validation sets together.

The optimisation of the hyperparameters was also done at this stage. These are values that are set manually before teaching, and they determine how the algorithm learns. Since the hyperparameters are not optimised directly during the learning process, it is crucial to choose them correctly, as they significantly impact the model's performance. The optimisation was done using the GridSearch technique (for RFR and XGBoost algorithms). GridSearch is a searching method in machine learning that finds the best combination of hyperparameters of a model by trying all possible combinations in a predefined matrix.

Fine-tuning of the models should be performed carefully to avoid overfitting and underfitting. In the former case, the algorithm tries too hard to adapt to the inputs, so it can learn the noise that comes with it. Conversely, underfitting occurs when the model fails to represent the outputs adequately.

3.3. Purpose and evaluation of the model

The primary goal of the modelling was to evaluate and improve the performance of the algorithms. During the evaluation, we compared the estimated values with the results of the MLC measurements and plotted them on graphs. In addition, we focused on analysing the feature importance of the inputs, which helps to understand which inputs are essential for estimating the individual outputs.

Standardised metrics were used for benchmarking. One of the most important of these is the determination coefficient, or R^2 (3). This is a statistical measure of how much the change in variable y can be explained by variable x in a regression model. Its value ranges from 0 to 1.

$$R^2\ [\%] = 1 - \frac{\sum (y_i - \hat{y}_i)^2}{\sum (y_i - \bar{y})^2} \tag{3}$$

where y_i are the original outputs, $\hat{y}_i$ are the estimated outputs, and $\bar{y}$ is the mean of the y values.

Various error values were also used to help determine the effectiveness of the estimation. These are measured in the same units as the estimated values. One of these is the mean squared error (4) (MSE), which represents the average of the squares of the differences between predictions and actual outputs. The smaller the value, the better the model fits the data.

$$MSE = \frac{1}{n}\sum_{i=1}^{n} (y_i - \hat{y}_i)^2 \tag{4}$$

where n is the number of the data points, y_i are the original outputs, and $\hat{y}_i$ are the estimated outputs.

The mean absolute error (5) (MAE) is also an important indicator, which measures the error associated with the model by taking the absolute value of each error and averaging them. The lower the value, the better the fit.

$$MAE = \frac{1}{n}\sum_{i=1}^{n} |y_i - \hat{y}_i| \tag{5}$$

where n is the number of the data points, y_i are the original outputs, $\hat{y}_i$ are the estimated outputs, and $|y_i - \hat{y}_i|$ is the absolute value of each data point's deviation.

3.4. Application of the random forest regressor

RFR models combine multiple decision trees to form a model with high accuracy and low overfitting for classification problems. The decision trees used in these models are all trained on a randomly sampled subset of the training dataset, and a random selection of features at each node is considered for splitting, increasing the model's diversity and robustness. The final classification decision of the model is determined by aggregating the predictions of all trees, typically by majority voting. RFR models are particularly effective for dealing with multidimensional data sets, as well as for mitigating the effects of noisy or irrelevant features.

3.5. Application of the XGBoost algorithm

The XGBoost (Extreme Gradient Boosting) algorithm relies on the "boosting" technique, which combines weak learners, in this case,

Table 1
XGBoost optimal hyperparameters for estimating each output.

Hyperparameters	pHRR [kW/m^2] estimation	TTI [s] estimation	THR [MJ/m^2] estimation	CR [%] estimation
n_estimators [-]	500	500	500	200
learning_rate [-]	0.2	0.2	0.01	0.2
max_depth [-]	3	3	3	3
subsample [-]	0.8	0.8	0.9	0.7
colsample_bytree [-]	0.7	0.9	0.9	0.9

decision trees, to create stronger models. The idea of boosting is that models correct each other's errors so that new algorithms give more accurate predictions with each iteration. Using gradient-based optimisation, XGBoost can efficiently handle nonlinearities and complex problems while providing high flexibility to fine-tune parameters. Its advantages include speed, accuracy and efficiency when working with large datasets. In addition, the model is easy to control and employs robust regularisation techniques that can reduce the risk of overfitting by penalising models that are too complex or have too large weights. We chose this set of decision trees because of its properties and the promising results of previous research.

The first step in the modelling process was to prepare the data. Then, the hyperparameters were optimised using GridsearchCV. Each hyperparameter has its purpose in increasing the efficiency of the model. One of the most important among them is the number of decision trees generated (n_estimators). This directly affects the complexity of the model: too few trees can lead to an underfitting model, while too many trees can lead to overfitting. When creating a new decision tree, the learning rate (eta) determines how much the model modifies its previous error. A low learning rate, such as 0.01 or 0.1, can lead to slower but often more accurate learning, as more subtle changes can be made to the model, while higher values can lead to faster convergence but may be more prone to over-learning. An important parameter, in addition to these, is the maximum depth of the decision tree (max_depth), which is the number of generations the tree can contain. When building each decision tree, the sampling rate (subsample) is used to select a subset of the training data. This is used to set how much of the training set the algorithm can work from during training. Finally, the 'colsample_bytree' parameter controls the percentage of the input features used to generate each tree. Like the sampling rate, lower values introduce more randomness, while higher values can give more accurate models but are prone to overfitting. Again, we searched various possible parameters to find the best-fitting settings.

After finding the best hyperparameters (see Table 1), we ran the model on the training data and then estimated and evaluated the data. In

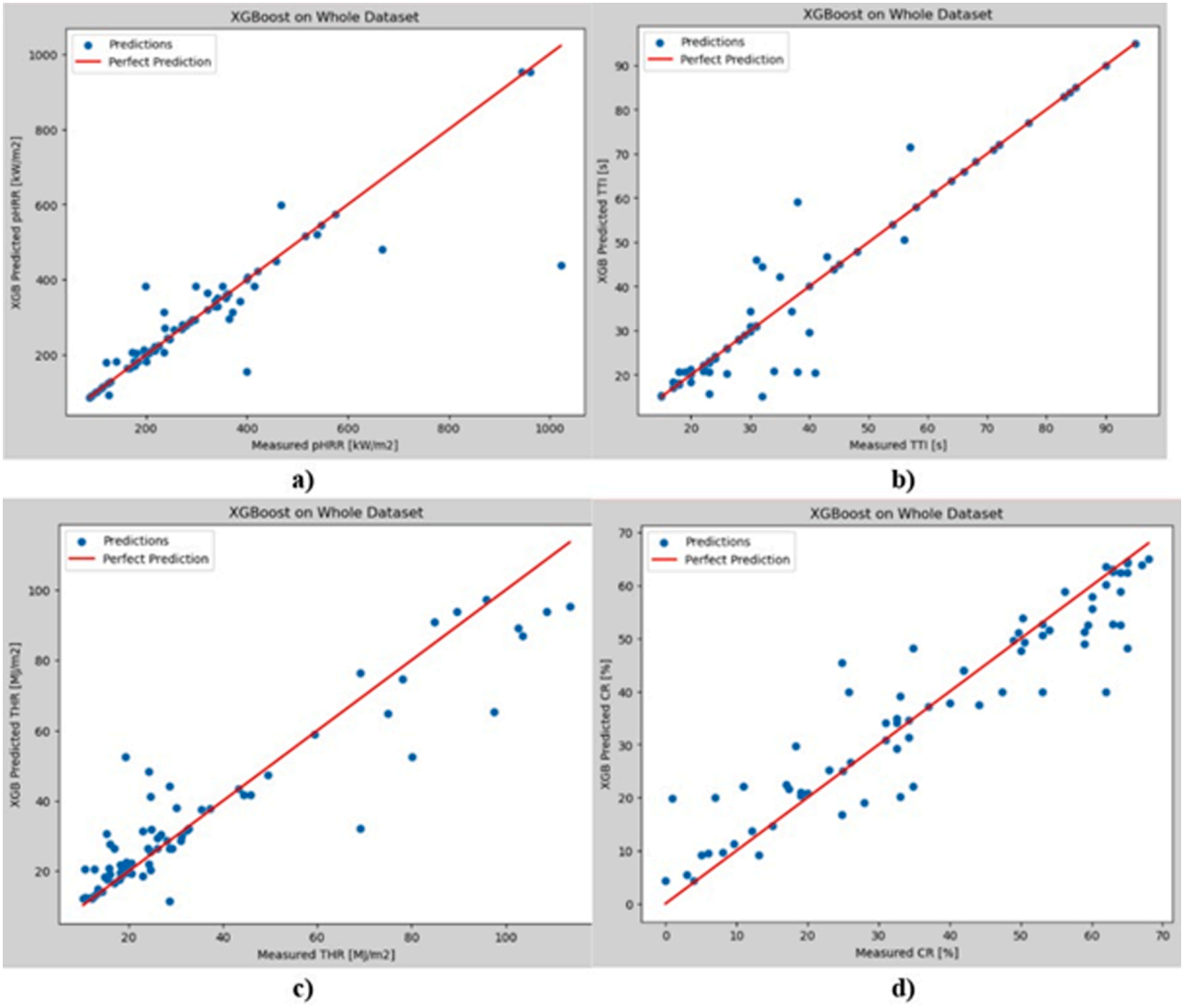

Fig. 4. Visualisation of XGBoost algorithm estimation: a) pHRR [kW/m^2], b) TTI [s], c) THR [MJ/m^2], d) CR [%].

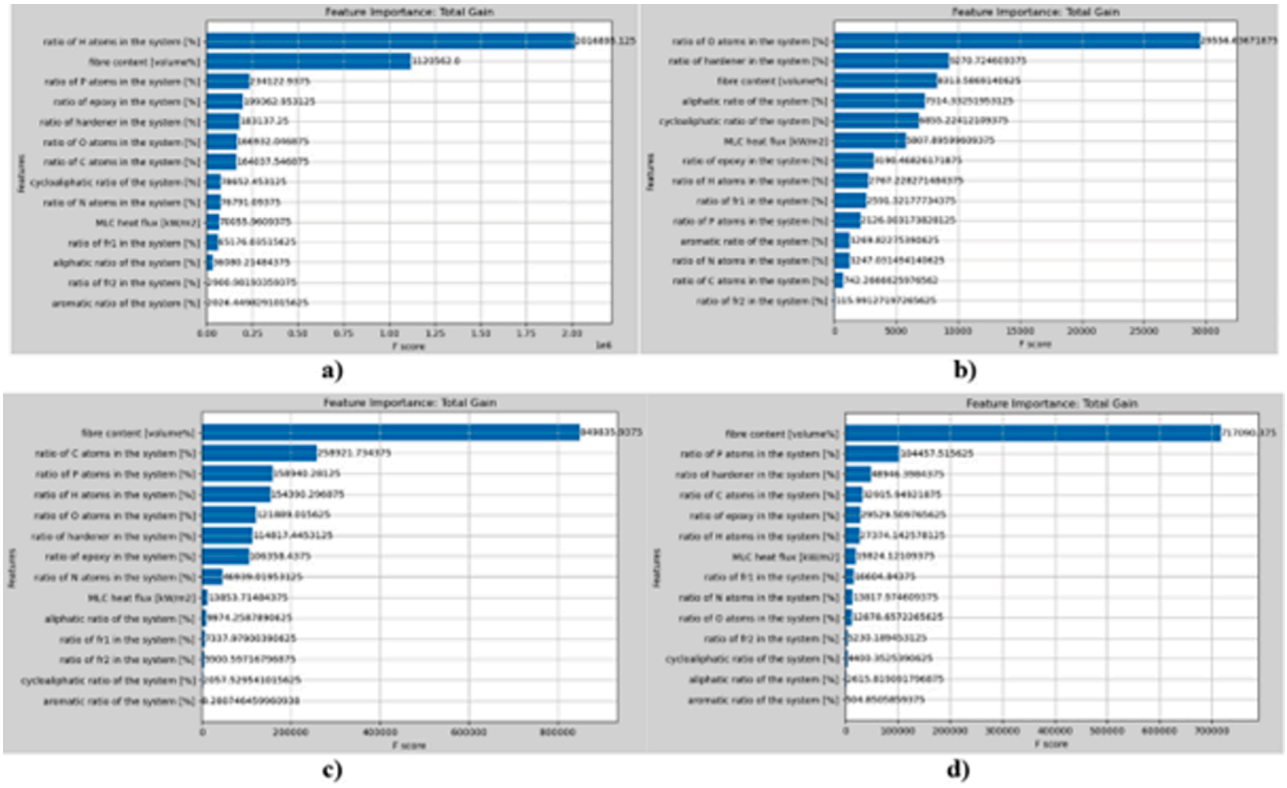

Fig. 5. Visualisation of estimation of XGBoost algorithm: a) pHRR [kW/m^2], b) TTI [s], c) THR [MJ/m^2], d) CR [%].

Table 2
The importance of XGBoost inputs for the estimation of each output.

Importance	pHRR [kW/m^2] estimation	TTI [s] estimation	THR [MJ/m^2] estimation	CR [%] estimation
1	ratio of H atoms [%]	ratio of O atoms [%]	fibre content [%]	fibre content [%]
2	fibre content [%]	ratio of the hardener [%]	ratio of C atoms [%]	ratio of P atoms [%]
3	ratio of P atoms [%]	fibre content [%]	ratio of P atoms [%]	ratio of the hardener [%]
4	ratio of epoxy [%]	aliphatic ratio [%]	ratio of H atoms [%]	ratio of C atoms [%]

the next step, the predictions for the entire data set were visualised as scatter plots for each output. In the graphs, a red line indicates the ideal prediction, showing the perfect correlation between the measured and estimated values. This allowed us to check the performance of the model visually. The horizontal axis shows the measured values, the vertical axis the estimated values, and the blue dots the estimated data points (Fig. 4).

Finally, the importance of the input variables used by the model was also examined. XGBoost provides an opportunity to determine which characteristics contribute most to the predictions. Using the plot_importance() function, we plotted the importance of the input variables, which helped us to understand which parameters have the most influence on the prediction for each output (Fig. 5). Table 2 concludes the four most important input parameters that were found to be important in estimating each output.

As shown in Table 2, fibre content ranks among the top four parameters for all predicted parameters, which is not surprising considering that it is related to the non-combustible material content. In most cases, atomic ratios (H, O, P, C) primarily influence fire performance. In the case of pHRR, H content is the most significant parameter, but this may be primarily because it is the most abundant in the system. Except for TTI, the P content is also present in all parameters, which is related to the amount of flame retardant. In the estimation of THR and CR, the fibre content became the most significant influencing parameter, which is also related to the amount of combustible material.

3.6. *Application of artificial neural network (ANN)*

The artificial neural network was chosen as a third learning model due to its frequent use in the literature for estimating fire performance. We need network layers, an activation function, an optimisation algorithm, a loss function and various operational parameters for model building. The input layer contains the input parameters for estimation, and the hidden layers perform the actual calculations. The output layer contains the estimated parameter. The number of hidden layers determines the depth of the mesh. The more layers you use, the more complex patterns the mesh can learn, but too many layers can make learning complex and slow and lead to overfitting. The number of neurons in a hidden layer determines its capacity, i.e. its ability to process information. The activation function plays a key role in determining the response of neurons. Its purpose is to transform the input data into a nonlinear space, thus allowing data processing with complex relations. One of the most popular activation functions is the Rectified Linear Unit (ReLU), which sets negative values to zero. This works well for most problems as it provides faster convergence.

Neural network learning is driven by an optimisation algorithm that updates the model parameters (weights and biases) to minimise the loss function. The loss function is usually the MAE or MSE metrics already described. One of the most commonly used optimisation algorithms is the Adaptive Momentum Estimation (Adam) method. Adam is an algorithm that updates the model parameters with the moving average of the derivative (gradient) of the loss function, thus providing fast and stable learning.

Once the model is built, we need to identify the parameters that should be fitted to the input data. One of these parameters is the 'epoch', which defines the number of times the data passes through the layers of

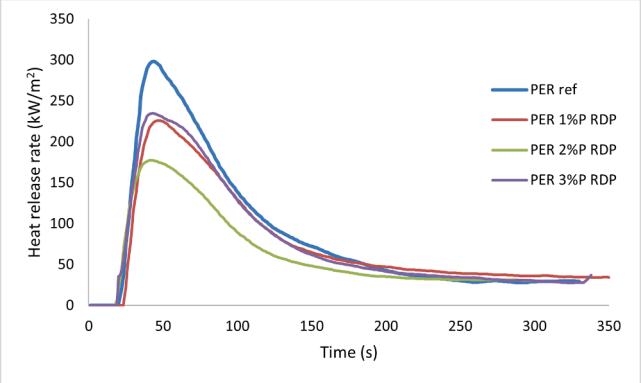

Fig. 6. Heat release rate curves of the flame retarded composites for validation.

Table 3
MLC results of flame retarded composites for validation.

Sample	TTI [s]	pHRR [kW/m^2]	Time to pHRR [s]	THR [MJ/m^2]	Char residue [%]
PER ref	19	298	44	28.7	53
PER 1 %P RDP	22	226	47	31.0	53
PER 2 %P RDP	18	177	42	19.5	60
PER 3 %P RDP	20	235	43	26.1	64

TTI: time to ignition; pHRR: peak of heat release rate; THR: total heat release. Average standard deviation of the measured mass loss calorimeter values: TTI: ±3; pHRR: ±30; time to pHRR: ±5; char residue: ±2.

the mesh, i.e. the number of times the optimisation algorithm updates the learning characteristics. The 'batch' size describes how many data points are used at a time to update the mesh within an iteration. If a smaller batch size is chosen, the learning will update the parameters more frequently, although this may slow down convergence. The mesh will update less with a larger value, but learning may be more stable. On the other hand, the learning rate is determined by the number of steps in the direction of the gradient that each parameter is updated. If too high, learning can become unstable; if too low, convergence can be slow.

The first step in the modelling process was to prepare the data as described above. Then, we created the model skeleton using the built-in libraries of Python Keras and TensorFlow. The model type was chosen to be sequential, allowing for a linear layer structure. The connectivity of the layers was set such that all neurons in the neighbouring layers were connected. The first layer has 128 neurons and uses the ReLU activation function to learn the nonlinear features. We then created a so-called 'dropout' layer, which uses 20 % random neuron removal to reduce the risk of over-learning. We then added two additional hidden layers with 64 and 32 neurons, both with ReLU activation, to allow the mesh to handle even more complex patterns. The model was terminated by the output layer with a single neuron. The Keras compile() function was utilised to configure the skeleton. This was used to select the optimisation algorithm (Adam with a learning rate of 0.001) and the loss function (MSE). Finally, the model was trained over 200 epochs with a batch size of 16. All the parameters listed were chosen according to the task to be performed. Unlike the RFR and XGBoost models, an automated hyperparameter search was not performed for the ANN due to the significantly higher computational cost associated with optimizing its large parameter space (e.g., layer architecture, learning rate, and batch size simultaneously). The parameters listed were chosen manually based on common practices for this type of regression task. This mesh can estimate continuous numerical values from inputs with complex nonlinear relationships. Once the model was constructed, the estimation and analysis of the results were carried out.

Table 4
Predicted pHRR values using different models.

pHRR [kW/m^2]				
Sample	Validated	Predicted		
		RFR	XGBoost	ANN
PER ref	**298**	324.33	324.70	330.45
PER 1 %P RDP	**226**	235.99	181.66	233.66
PER 2 %P RDP	**177**	199.94	151.80	231.98
PER 3 %P RDP	**235**	194.22	195.66	230.29

Table 5
Predicted THR values using different models.

THR [MJ/m^2]				
Sample	Validated	Predicted		
		RFR	XGBoost	ANN
PER ref	**28.7**	25.75	26.37	37.38
PER 1 %P RDP	**31.0**	17.41	15.56	37.38
PER 2 %P RDP	**19.5**	15.35	13.02	37.38
PER 3 %P RDP	**26.1**	15.48	15.76	37.38

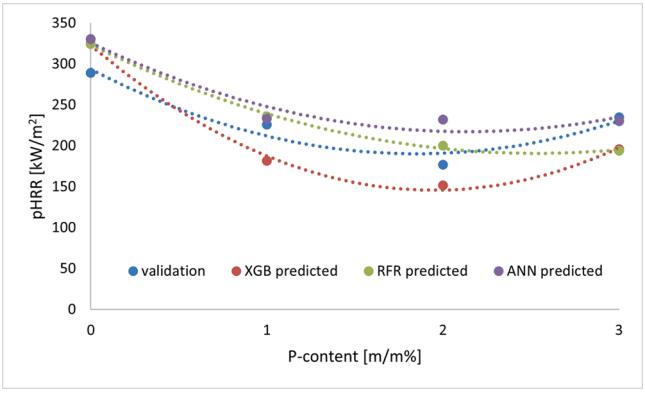

Fig. 7. Correlation between validated and predicted pHRR values.

4. Evaluation of the results

4.1. Mass loss calorimetry results for validation

To validate the models, we prepared carbon fibre reinforced PER-based composites containing 0, 1, 2 and 3 mass % P from RDP and investigated their fire performance using mass loss type cone calorimetry (MLC). We registered the heat release rate (Fig. 6), the time to ignition, the total heat release and the char residue (Table 3). The time to ignition remained almost the same, whether the composite contained flame retardant (FR) or not. The addition of RDP reduced the peak and total heat release, while the char residue increased as expected. The composite containing 2 % P RDP had the best overall performance, as the heat releases did not reduce further with higher FR content.

4.2. Evaluation of the prediction

In general, the algorithms learned to estimate the different combustion characteristics with varying degrees of success, even though the optimisation of the hyperparameters (for XGBoost and RFR) was specific to each characteristic. In our latest research [11], we predicted the fire performance using the ANN algorithm. It should be noted that small-scale measurements were also used as input in that research. Since a similar 'feature importance' analysis was also done, we know that these small-scale test results greatly influenced the output, especially for

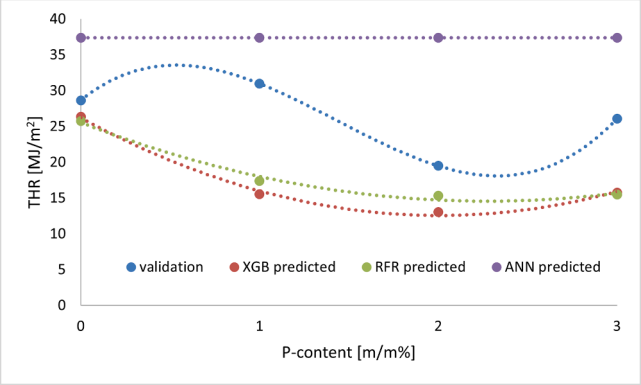

Fig. 8. Correlation between validated and predicted THR values.

Table 6
Predicted TTI values using different models.

TTI [s]				
Sample	Validated	Predicted		
		RFR	XGBoost	ANN
PER ref	19	22.19	21.27	25.29
PER 1 %P RDP	22	23.55	22.32	25.29
PER 2 %P RDP	18	25.64	24.80	25.29
PER 3 %P RDP	20	25.32	24.81	25.29

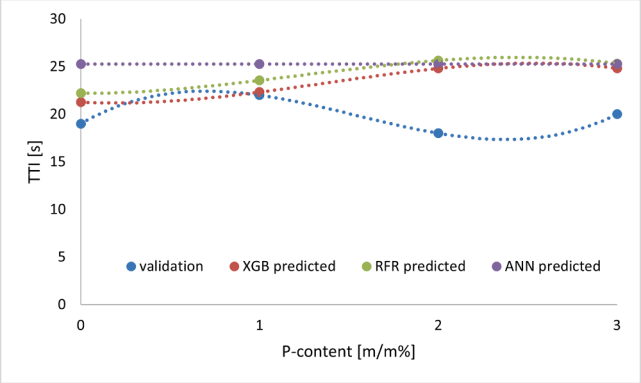

Fig. 9. Correlation between validated and predicted TTI values.

pHRR and THR estimation. Without this, the XGBoost and RFR algorithms we created determined different parameter accuracy for each output. One of the shortcomings of ANN is that it is impossible to analyse the 'feature importance' without supporting software. We summarised the pHRR values in Table 4 and the THR estimation results in Table 5, while Figs. 7 and 8 show the correlation between the validation and the predicted heat releases.

In the case of pHRR, the XGBoost algorithm and the ANN slightly overestimated the real values, but all algorithms followed a similar trend to the validation dataset (Fig. 7). The overestimation can be explained by the fact that the algorithms typically considered the fibre content the most essential factor. In our dataset used for training (for 75 data points), this attribute takes four different values (around 0 (neat resin), 40 (hand laminated composite), 60 (wet compressed composite) and 67 (vacuum infused composite)). This results in the difficulty of tracking small changes with such input. Our database typically has much higher pHRR results, so the algorithms are also overestimating.

The total heat release was underestimated by the XGB and RFR algorithms, and the ANN predicted the same THR value for each composite (Fig. 8). This can be attributed to the limited variation and small

Table 7
Predicted CR values using different models.

CR [%]				
Sample	Validated	Predicted		
		RFR	XGBoost	ANN
PER ref	53	50.82	50.18	53.06
PER 1 %P RDP	53	55.90	55.43	55.29
PER 2 %P RDP	60	58.41	56.39	57.52
PER 3 %P RDP	64	57.60	56.20	59.75

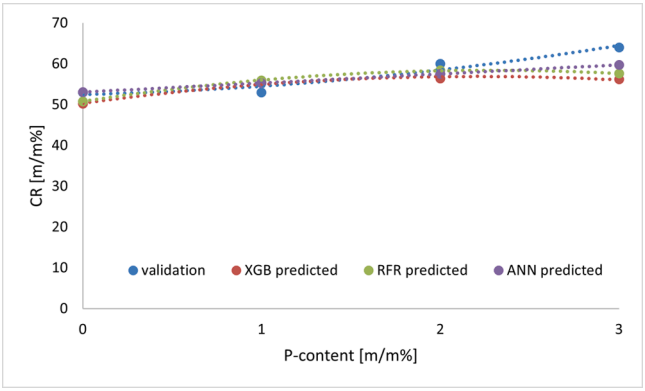

Fig. 10. Correlation between validated and predicted CR values.

number of THR values in the training dataset. Since THR is a single cumulative quantity with relatively low dispersion across the formulations, the models had insufficient signal to learn meaningful patterns, leading to underestimation or constant predictions.

Time to ignition values were well predicted by XGB and RFR algorithms at low FR-content and slightly overestimated at higher concentrations, while the prediction by ANN had a similar issue as in the case of THR prediction: the same values were predicted due to the small training dataset (Table 6, Fig. 9).

The best overall results were obtained when predicting the char residue (Table 7, Fig. 10). All three algorithms estimated the real results with minimal deviation. The superior prediction accuracy for CR is also a noteworthy finding. The physico-chemical explanation is the following: CR is primarily governed by the material's chemical composition (especially P content) and cross-link density, factors that are likely well-captured by the chosen 'atomic ratio' and 'structural ratio' input features and are less dependent on the complex gas-phase combustion processes. This explains the model's relative success with CR, even with limited data.

4.3. Determination coefficients and errors

To evaluate the algorithms consistently, we calculated R^2, MSE and MAE for the complete data set and then for the predicted values. In the latter case, only the test and validation datasets were considered together, as they perform the same function in the algorithm, since they contain unknown output for the constructed model.

Over- or under-fitting can be easily checked by looking at the difference between the corresponding errors in the predicted and the complete data series. Over- or underfitting occurs when the metrics produce very good values when combined, but drastically worse values when predicted only. Of course, these values will always be smaller, but they should be proportional to the ratio of predicted to training data sets.

The learning algorithms adjust the weights based on the training data to be the most accurate fit. Table 8 shows the evaluation of the fit to the complete data set.

Table 8
Determination coefficients and statistical errors of the complete dataset.

Full dataset												
Algorithm	pHRR [kW/m^2]			TTI [s]			THR [MJ/m^2]			CR [%]		
	R^2 [-]	MSE [kW/m^2]	MAE [kW/m^2]	R^2 [-]	MSE [s]	MAE [s]	R^2 [-]	MSE [MJ/m^2]	MAE [MJ/m^2]	R^2 [-]	MSE [%]	MAE [%]
RFR	0.89	3911	41	0.95	24	3.46	0.85	120	7	0.93	109	4
XGBoost	0.96	1355	16	0.95	24	1.94	0.88	96	3	0.95	22	2
ANN	0.57	15.632	69	0.57	199	9.5	0.3	808	22	0.74	31	9

Table 9
Determination coefficients and statistical errors of the test and validation datasets together.

Test and validation datasets together												
Algorithm	pHRR [kW/m^2]			TTI [s]			THR [MJ/m^2]			CR [%]		
	R^2 [-]	MSE [kW/m^2]	MAE [kW/m^2]	R^2 [-]	MSE [s]	MAE [s]	R^2 [-]	MSE [MJ/m^2]	MAE [MJ/m^2]	R^2 [-]	MSE [%]	MAE [%]
RFR	0.30	14.079	94	0.32	358	14	0.53	428	16	0.82	73	6
XGBoost	0.25	15.139	97	0.39	251	12	0.67	220	10	0.81	80	6
ANN	0.13	17.515	98	0.58	158	9	0.33	680	21	0.67	139	9

Typically, XGBoost also performs best in the statistical indicators of the full dataset, alongside RFR: R^2 values are over 0.85, which means quite good prediction. Although it should be considered that with such a small dataset, high R^2 does not guarantee generalisability, so the distribution of the residuals (normality, homoscedasticity and independence) and the potential effects of outliers have to be considered. The metrics of the ANN algorithm are weaker concerning both the full dataset and the test and validation sets (Table 9). However, it is much less capable of mapping validation data. Nevertheless, it can provide a rough estimation of the test values, since they are derived from a database with similar data. Validation values, on the other hand, need to be mapped to new inputs. The metrics of the algorithms concerning the test and validation sets together are lower than in the case of the full dataset, which can be explained by the small amount of data.

Consequently, the RFR and XGBoost algorithms are more effective in predicting flammability parameters, as both models handle smaller, nonlinear data sets better and are more efficient in avoiding overfitting. In contrast, ANN requires much more data and computation to achieve the same performance. A closer analysis of these error metrics reveals a limitation for pHRR predictions on the test and validation sets, significantly higher than the reported experimental standard deviation of our validation samples. This underscores that while the models can successfully identify general trends, their current ability is limited to qualitative trend analysis rather than quantitative prediction.

5. Conclusions

This study focused on the prediction of thermal degradation characteristics of polymers and polymer composites, using three machine learning methods. First, we summarised which machine learning algorithms have been used in recent studies to estimate specific combustion parameters, elaborating on the structure of these algorithms. Then we applied three selected machine learning algorithms to predict four different combustion parameters. These algorithms were XGBoost, Random Forest Regressor (RFR), and Artificial Neural Networks (ANN). The estimated parameters were the peak heat release rate (pHRR), total heat release (THR), time to ignition (TTI), and char residue (CR). The algorithms were trained using a database compiled from data obtained through mass loss type cone calorimetry (MLC) experiments. We attempted to determine how accurately these models could predict the parameters when only the composite's chemical structure and the fibre content were used as input parameters.

For model validation, we prepared four composite samples made from a pentaerythritol-based epoxy resin (PER), a cycloaliphatic amine-type crosslinking agent, a phosphorus-containing flame retardant additive (RDP), and carbon fibre. One of the samples served as a reference, containing no flame retardant, while the remaining three samples contained 1 %, 2 %, and 3 % phosphorus by mass. We performed MLC measurements on the prepared samples and compiled the validation dataset from these results.

We wrote the machine learning models in Python using Jupyter Notebooks. After processing the data, we created the models and determined the key hyperparameters that controlled their operation. We identified the most critical decision-making parameters for each output. Finally, we calculated the performance metrics for both the entire dataset and the predicted set (MAE, MSE, and R^2). Among the models used, XGBoost was the most successful for predicting pHRR with an R^2 value of 0.96, while RFR performed best at predicting TTI (0.95 R^2). These algorithms were able to determine the nature of the values correctly. Although ANN produced reasonable metrics, it did not achieve the accuracy of the other two algorithms. Therefore, XGBoost and RFR provided more successful solutions than ANN, as these models handle smaller, nonlinear data sets better and are more efficient in avoiding overfitting. The ANN's performance was likely also limited by the manual selection of hyperparameters. Future work should explore automated hyperparameter tuning and implement Early Stopping to improve the ANN's robustness and prevent overfitting. Our future studies will prioritize the collection of larger, more comprehensive datasets to improve model robustness and generalisability. Besides the international collaboration of researchers, alternatively, we are encouraged to explore techniques specifically designed for limited data scenarios, such as small-sample learning approaches or deep learning methods optimized for sparse datasets. These strategies may help mitigate the limitations posed by small sample sizes and enhance the performance and applicability of models in similar settings.

Statement

During the preparation of this work, the authors used ChatGPT, DeepL and Grammarly in order to improve the language and readability of the paper. After using this tool, the author(s) reviewed and edited the content as needed and take full responsibility for the content of the published article.

CRediT authorship contribution statement

Ákos Pomázi: Writing – review & editing, Writing – original draft, Conceptualization. **Gergely Magyar:** Writing – review & editing, Data curation. **Andrea Toldy:** Writing – review & editing, Writing – original draft, Supervision, Project administration, Conceptualization.

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.

Reports a relationship with that includes:. Has patent pending to. If there are other authors, they declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgements

This research was funded by the National Research, Development and Innovation Office (NKFIH K142517 and STARTING 150473). Project no TKP-6–6/PALY-2021 has been implemented with the support provided by the Ministry of Culture and Innovation of Hungary from the National Research, Development and Innovation Fund, financed under the TKP2021-NVA funding scheme. Ákos Pomázi is thankful for the support of the Michelberger Master Prize of the Hungarian Academy of Engineering. This research was partly supported by the Doctoral Excellence Fellowship Programme (DCEP) is funded by the National Research Development and Innovation Fund of the Ministry of Culture and Innovation and the Budapest University of Technology and Economics, under a grant agreement with the National Research, Development and Innovation Office.

Data availability

Data will be made available on request.

References

[1] M.Z. Naser, Mechanistically informed machine learning and artificial intelligence in fire engineering and sciences, Fire Technol. 57 (2021) 2741–2784, https://doi.org/10.1007/s10694-020-01069-8.

[2] H. Vahabi, M.Z. Naser, M.R. Saeb, Fire protection and materials flammability control by artificial intelligence, Fire Technol. 58 (2022) 1071–1073, https://doi.org/10.1007/s10694-021-01200-3.

[3] B. Mahesh, Machine learning algorithms – A review, Int. J. Sci. Res. 9 (1) (2020) ART20203995, https://doi.org/10.21275/ART20203995.

[4] D. Dhall, R. Kaur, M. Juneja, Machine learning: a review of the algorithms and its applications, in: Proceedings of ICRIC 2019, 2020, pp. 47–63, https://doi.org/10.1007/978-3-030-29407-6_5.

[5] N.R. Smalheiser, Chapter 13 – Correlation and Other Concepts You Should know" in *Data Literacy*, Academic Press, 2017, pp. 169–185, https://doi.org/10.1016/B978-0-12-811306-6.00013-0.

[6] R. Wang, T. Fu, Y.-J. Yang, X.-L. Wang, Y.-Z. Wang, Deeper insights into flame retardancy of polymers by interpretable, quantifiable, yet accurate machine-learning model, Polym. Degrad. Stab 230 (1) (2024) 110981, https://doi.org/10.1016/j.polymdegradstab.2024.110981.

[7] H.T. Nguyen, K.T.Q. Nguyen, T.C. Le, L. Soufeiani, A.P. Mouritz, Predicting heat release properties of flammable fiber-polymer laminates using artificial neural networks, Compos. Sci. Technol. 215 (14) (2021) 109007, https://doi.org/10.1016/j.compscitech.2021.109007.

[8] J.F. Fatriansyah, E. Kustiyah, S.N. Surip, A. Federico, A.F. Pradana, A. S. Handayani, M. Jaafar, D. Dhaneswara, Fine-tuning optimisation of poly lactic acid impact strength with variation of plasticiser using simple supervised machine learning methods, Express. Polym. Lett. 17 (9) (2023) 964–973, https://doi.org/10.3144/expresspolymlett.2023.71.

[9] Z. Zhang, Z. Jiao, R. Shein, P. Song, Q. Wang, Accelerated design of flame retardant polymeric nanocomposites via machine learning prediction, ACS Appl. Eng. Mater. 1 (1) (2023) 596–605, https://doi.org/10.1021/acsaenm.2c00145.

[10] J. Xiao, J. Hobson, M. Haranczyk, D.-Y. Wang, Machine learning framework to predict instantaneous heat release rate of polymer nanocomposites in cone calorimetry, Polym. Degrad. Stab 218 (2023) 110563, https://doi.org/10.1016/j.polymdegradstab.2023.110563.

[11] D.N. Phan, A.B. Morgan, L. Poudel, R. Bhowmik, A machine learning platform for polymer flammability prediction, Polym. Degrad. Stab 240 (2025) 111411, https://doi.org/10.1016/j.polymdegradstab.2025.111411.

[12] P. Jafari, R. Zhang, S. Huo, Q. Wang, J. Yong, M. Hong, R. Deo, H. Wang, P. Song, Machine learning for expediting next-generation of fire-retardant polymer composites, Compos. Commun. 45 (2024) 101806, https://doi.org/10.1016/j.coco.2023.101806.

[13] Á. Pomázi, A. Toldy, Predicting the flammability of epoxy resins from their structure and small-scale test results using an artificial neural network model, J. Therm. Anal. Calorim. 148 (2023) 243–256, https://doi.org/10.1007/s10973-022-11638-4.

[14] A. Bifulco, A. Casciello, C. Imparato, S. Forte, S. Gaan, A. Aronne, G. Malucelli, A machine learning tool for future prediction of heat release capacity of in-situ flame retardant hybrid Mh(OH)2-Epoxy nanocomposites, Polym. Test. 127 (2023) 108175, https://doi.org/10.1016/j.polymertesting.2023.108175.

[15] C. Yan, X. Lin, X. Feng, H. Yang, P. Mensah, G. Li, Advancing flame retardant prediction: a self-enforcing machine learning approach for small datasets, Appl. Phys. Lett. 122 (25) (2023) 251902, https://doi.org/10.1063/5.0152195.

[16] A. Toldy, Á. Pomázi, B. Szolnoki, The effect of manufacturing technologies on the flame retardancy of carbon fibre reinforced epoxy resin composites, Polym. Degrad. Stab 174 (2020), https://doi.org/10.1016/j.polymdegradstab.2020.109094, 109094/1-109094/10.

[17] Gy. Marosi, K. Bocz, Fibers and safety in fire retarded polymer systems: are we just at the beginning of the road? Express. Polym. Lett. 18 (2) (2024) 116–117, https://doi.org/10.3144/expresspolymlett.2024.9.

[18] A. Toldy, Challenges and opportunities of polymer recycling in the changing landscape of European legislation, Express. Polym. Lett. 17 (11) (2023) 1081, https://doi.org/10.3144/expresspolymlett.2023.81.

[19] A. Toldy, Safe and sustainable-by-design: redefining polymer engineering for a greener future, Express. Polym. Lett. 19 (4) (2025) 350, https://doi.org/10.3144/expresspolymlett.2025.25.

[20] Á. Pomázi, B. Szolnoki, A. Toldy, Flame retardancy of low-viscosity epoxy resins and their carbon fibre reinforced composites via a combined solid and gas phase mechanism, Polym. (Basel) 10 (10) (2018) 1081, https://doi.org/10.3390/polym10101081.

[21] A. Toldy, Development of environmentally friendly epoxy resin composites. Doctoral thesis, Hung. Acad. Sci. (2017). ISBN 978-963-313-262-3.