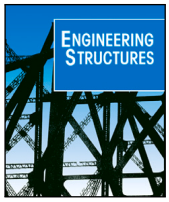
Engineering Structures 276 (2023) 115310

Contents lists available at ScienceDirect

Engineering Structures

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



A comparative study on the most effective machine learning model for blast loading prediction: From GBDT to Transformer

Qilin Li [a], Yang Wang [b], Yanda Shao [a], Ling Li [a,*], Hong Hao [b]

[a] Computing, Curtin University, Perth, Australia
[b] Centre for Infrastructural Monitoring and Protection, Curtin University, Perth, Australia

ARTICLE INFO

Keywords:
Blast loading
Machine learning
Transformer
Neural network
BLEVE

ABSTRACT

In this paper, we present a rigorous comparative study to assess and identify the most effective machine learning model for blast loading prediction. Blast loads are known to produce catastrophic effects including structural collapse and personnel fatality. Accurate and efficient prediction of these extreme loads using empirical methods and numerical solvers remains a challenging problem. Machine learning provides a promising alternative solution, which has been increasingly used in various engineering applications. However, there is seldom any analysis or justification of the selection of machine learning method that would lead to the best performance for such applications. For example, most existing machine learning-based approaches for blast loading prediction utilise the classic multi-layer perceptron (MLP) network with no justifications of their suitability and efficiency nor attempts of leveraging other state-of-the-art neural network architectures. In this study, four well-known machine learning models, including one ensemble tree method and three neural networks of different types, are investigated to demonstrate the effectiveness of different machine learning methods for blast loading prediction. It is showcased using BLEVE (boiling liquid expanding vapour explosion) pressure prediction that the Transformer model achieves the best performance, reaching a relative error of 3.5% and R^2 0.997 that outperforms the existing MLP approach (relative error 6.0%, R^2 0.985) with a clear margin. This study shows that the Transformer network is an effective tool for prediction of blast loading from BLEVE as well as other explosion sources.

1. Introduction

An explosion is a sudden expansion in volume associated with an extremely rapid release of energy, imposing enormous threats to personnel and assets. Catastrophic consequences are known to be generated not only from direct explosion effects, i.e., blast overpressure and fragments, but also from side effects such as structural collapse [1]. Due to the large number of accidental explosions and malicious terrorist attacks, such as the La Mestiza coal mine explosion (2022, 15 fatalities) and Peshawar Mosque bombing (2022, 63 fatalities), it is of significant interest to predict blast loadings for the accurate analysis of structural responses to blast loading, and hence design the effective measures for human [2,3] and infrastructure [4,5] protections.

Accurate and efficient prediction of blast loads remains a challenging problem. Blast loadings by nature appear in an extremely short duration but with very high intensity, resulting in different structural responses compared to those from static and low-rate loads such as earthquake and wind loads [1]. The blast loading is a complex nonlinear function of many factors, such as explosive material, weight, shape, location, etc. The relevant blast parameters, such as pressure and impulse, are primarily dependent on the amount of energy released in the detonation [6]. For different explosive materials, it is a common practice for conventional (semi) empirical approaches [7,8] to convert the charge weight to an equivalent weight of TNT, and leverage the established TNT blast overpressure prediction curves to predict blast parameters. These approaches, however, are reported to be reliable only for far-field, geometrically simple scenarios [1,9], mainly due to the scarcity of experiment data and the difficulty of accurate measurement.

In the recent decades, owing to the development of computational fluid dynamics (CFD) and the increase of computational power, first-principle numerical methods are developed for predicting blast effects, yielding numerous commercial software for blast simulation, such as Air3D, LS-DYNA, FLACS. These CFD-based numerical models, governed by partial differential equations of the underlying physics, can provide accurate simulations of blast dynamics when properly validated using experimental data. It is even preferable to utilise CFD over

* Corresponding author.
E-mail address: l.li@curtin.edu.au (L. Li).

https://doi.org/10.1016/j.engstruct.2022.115310

Available online 2 December 2022

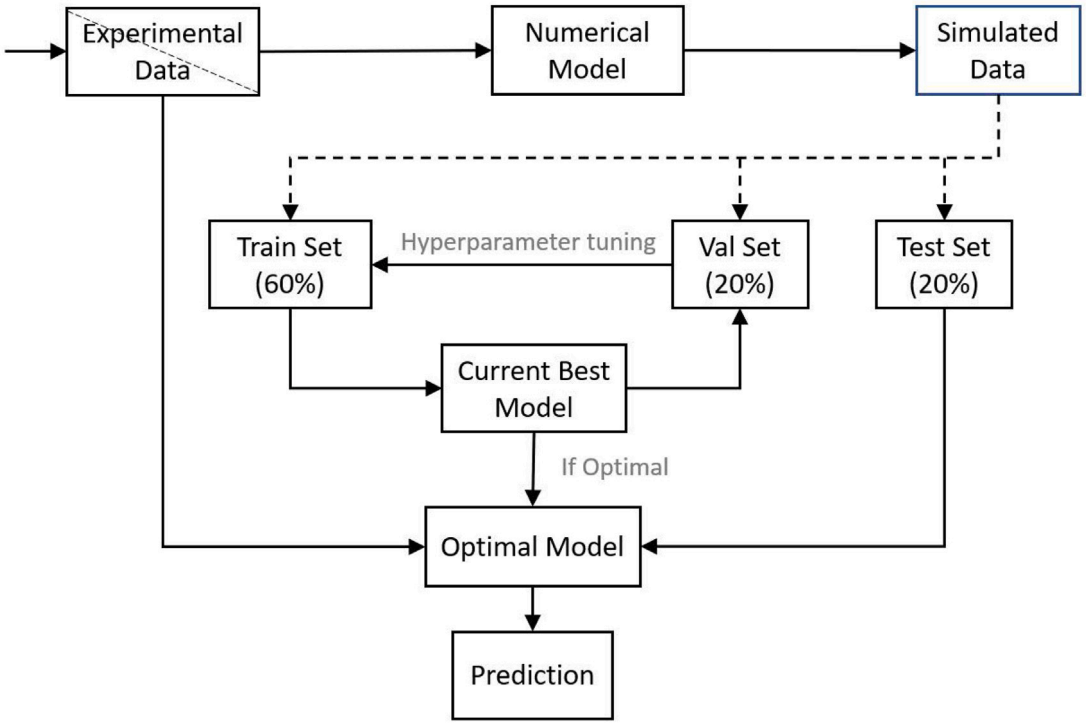


Fig. 1. The flowchart of machine learning based analysis.

semi-empirical approaches in cases of complex environments and non-conventional explosives, such as blast loading in urban landscape [10] and blast loading from gas [11] and vapour cloud explosions [12]. However, to achieve satisfactory accuracy, CFD approaches generally require a fine grid (or mesh) on which the heavy computation can be progressively performed. Complex geometries exacerbate the high computation cost, diminishing the advantage of CFD approaches in such scenarios.

It is therefore of interest to develop a predictive approach for blast effects prediction that has both the high accuracy of numerical models and the low computational cost of semi-empirical methods. Machine learning approaches fit an implicit function that maps input variables to output targets by learning from data, which avoids iterative computations and trivial empirical functions. It is reported in the blast engineering field that these learning-based approaches achieve satisfactory accuracies with computation times comparable to semi-empirical solvers [13]. Among them, neural networks (NNs) remain the dominant machine learning approach. For instance, Remennikov et al. [14] attempted to use NNs to predict the effectiveness of blast wall barriers. Dennis et al. [13] used NNs for the blast loading prediction in an internal environment. Li et al. [15] predicted gas explosion with a mixture of liquid and vapour using NNs. Bui et al. [16] conducted a comparative study of popular machine learning models for blast prediction in open-pit mines and concluded that NNs remain the most effective model.

The popularity of NNs is partially due to the tremendous success of deep learning [17,18] on data domains such as audio, texts, and images. The rapid development of deep learning has resulted in a wide range of network architectural designs, such as ResNet [19] for image recognition and Transformer [20] for natural language processing. The success of deep learning, however, has not yet been replicated for tabular data in which rows represent data instances and columns are features. Tabular data are ubiquitous in blast engineering, e.g., explosives are characterised by the material type, charge weight, location, etc, which prevented blast engineering practitioners from taking advantage of the advances in deep learning. To the best of our knowledge, all applications in blast prediction use the classic fully-connected feed-forward neural network only, commonly referred to as multi-layer perceptron (MLP), with no discussion or investigation on its suitability and effectiveness as compared to the other state-of-the-art NN architectures.

The goal of this study is to provide a rigorous comparative study of various machine learning models for blast loading prediction. Four well-known models, including one ensemble tree approach, lightGBM, and three NNs with different architectures, i.e., MLP, ResNet, Transformer, are selected and compared in a fair benchmark setting. This is achieved by following a comprehensive machine learning model development routine, including data preprocessing, hyperparameter tuning, model selection, model validation and testing. Numerous deep network training techniques are employed to ensure optimal performances, including mini-batch training, Bayesian optimisation-based hyperparameter search, early stopping, etc. For simplicity, the peak pressure prediction of boiling liquid and expanding vapour explosion (BLEVE) [21] in the free field is chosen as a typical example of the blast loading prediction problem. Note that prediction of BLEVE loading using MLP has been attempted in [15]. Following this study, 1000 CFD simulated BLEVE cases are used to train and evaluate the four candidate machine learning models. With a rigorous comparative experiment, it is demonstrated that the Transformer model achieves the best accuracy, significantly outperforms the widely-adopted MLP. An in-depth analysis of the optimal Transformer model is then conducted to comprehensively investigate the model behaviour, including an ablation study to show the model sensibility. Finally, the developed Transformer model is applied to public-available BLEVE field test data on which 25.4% relative error is achieved (compared to 27.6% of MLP [15]), showing its potential for BLEVE and other blast loading predictions. Fig. 1 presents a flowchart of the overall procedure.

The rest of the paper is organised as follows. Section 2 presents the CFD simulation for BLEVE numerical modelling, including the details of model validation and data generation. Section 3 briefly introduces the selected machine learning models and presents the comparative study of their performances. The analysis of the winning Transformer model is presented in Section 4, including an ablation study of design choices

Table 1
Summary of BLEVE cases for the validation of FLACS simulation.

Experiments	Fluid	Failure P (MPa)	Tank V (m^3)	Liquid T (K)	Liquid ratio(%)	Liquid status
02-01 (Birk)	Propane	1.80	2.00	330	51	Superheated
02-02 (Birk)	Propane	1.56	2.00	320	52	Superheated
02-03 (Birk)	Propane	1.81	2.00	325	53	Saturated
BG-2 (Johnson)	Butane	1.52	5.66	374	39	Superheated
BG-3 (Johnson)	Butane	7.50	5.66	323	68	Saturated

of the Transformer. Section 5 shows the prediction of Transformer on field test data, and the conclusion is summarised in Section 6.

2. Numerical modelling

2.1. Validation of FLACS simulation

For numerical analysis of gas explosions, the most commonly used commercial CFD software is FLACS. Based on a three-dimensional code, FLACS can solve Favre-averaged transport equations for mass, momentum, enthalpy, turbulent kinetic energy, etc. The Reynolds-averaged Navier–Stokes equations are used in FLACS simulation by applying the ideal gas law and the standard $k-\epsilon$ model for turbulence modelling [22]. The Euler equations, flux-corrected transport scheme and a second-order flux correction are employed to model the detonation of condensed explosive and blast wave propagation using the FLACS-Blast module. Cell-centred grids are used to solve scalar variables (i.e., pressure, temperature and density), while staggered grids are used to solve velocity components [23].

Five medium-to-large BLEVE field test cases collected by [24,25] are used for the validation of FLACS simulation, and their specifications can be found in Table 1. Based on the pressure and temperature at failure, the NIST fluid properties database (REFPROP) v8 [26] is used to determine if the liquid is superheated. For superheated cases, the liquid correction method [27] is used to modify the failure pressure by adding the liquid flashing energy, resulting in the corrected failure pressures at 2.84 MPa, 2.46 MPa, 2.57 MPa for 02-01 (Birk), 02-02 (Birk), BG-2 (Johnson), respectively. Please refer to [15,27] for the detailed calculation of the liquid correction method.

With the corrected input data, these five BLEVE cases are simulated by FLACS, and 18 peak pressures at various standoff distances (from 10 to 100 m) are used to evaluate the accuracy of FLACS simulation (the peak pressure generated by the first shock wave is the only available data in the literature). Fig. 2 presents the FLACS simulation results and field test recordings. It can be seen from the scatter plot in Fig. 2(a) that the FLACS simulation aligns well with the field test data in terms of the peak pressure. Following the official guideline of the FLACS manual [23], the parabola plot is used to evaluate if the simulation result is "Excellent", "Acceptable" or "Poor", and it is shown in Fig. 2(b) that 72% of the assessment data are within the "Excellent" performance region and 100% data are within the "Acceptable" performance region, indicating that the FLACS CFD model produces accurate simulation and is valid to generate BLEVE data.

2.2. Generation of simulated data

Using the validated FLACS CFD model, 1000 BLEVE cases (500 butane and 500 propane) are simulated by varying 11 input variables as shown in Table 2. Specifically, the shape of BLEVE source is modelled as a rectangular tank to reduce the mass residual problem in the FLACS simulation [15]. Following the European manufacturing standard of LPG tanks by Kadatec [28], the maximum tank dimension is set to $10 \times 3 \times 3$ m for the length, width and height of the tank. Accordingly, the maximum BLEVE height is set to 2.2 m. The tank failure pressure is bounded between 500 kPa and the critical pressure of the liquid (3700 kPa for butane and 4200 kPa for Propane). The tank is assumed to contain both liquid and vapour with the liquid fill level between

Table 2
Summary of 11 input variables of the BLEVE dataset.

Variable	Notation	Range	
		Butane	Propane
Tank failure pressure (kPa)	$Tank_P$	500–3700	500–4200
Liquid fill ratio (%)	R_L	10–90	10–90
Tank width (m)	$Tank_W$	0.2–3	0.2–3
Tank length (m)	$Tank_L$	0.2–10	0.2–10
Tank height (m)	$Tank_H$	0.2–3	0.2–3
BLEVE height (m)	B_H	0–2	0–2
Vapour temperature (°C)	T_V	1–304	1–192
Liquid temperature (°C)	T_L	1–152	1–96
Liquid status	$Status$	Superheated or Saturated	
Vapour height (m)	V_H	$Tank_H \times (1-R_L)$	
Standoff distance (m)	D	5–50	5–50

10 and 90 percent. The failure temperature goes up to the critical temperature of the substance to consider both the superheated and saturated liquid status. The standoff distance is varied from 5 m to 50 m with a one-meter interval, which means 46 peak pressures at different distances will be recorded for each BLEVE simulation. The monitor point is placed on the ground that is modelled as a rigid surface and no other barriers exist to simulate a free-field explosion wave propagation scenario. More details about the FLACS simulation setting can be found in [15].

To better illustrate the simulated data, the pairwise Pearson correlation coefficient is calculated for all input variables and the output variable, resulting in the correlation matrix as shown in Fig. 3. For the output pressure P, the most (negatively) correlated input parameter is the standoff distance D, and the second most correlated parameter is the vapour height V_H.

The 1000 BLEVE cases take around two months of continuous simulation in FLACS using a standard desktop PC, and in total 46000 (46×1000) data instances are collected. The dataset is split into a training set (60%), a validation set (20%), and a testing set (20%) for the training, hyperparameter tuning, and evaluation of machine learning models, respectively. The substantial amount of data will enable the training of deep neural networks with high accuracy and good generalisability.

3. Comparison of machine learning models

The simulated BLEVE data are presented in tabular format, where each row represents a data instance and each column represents a variable (feature). Even in the machine learning community, it is still not clear which machine learning model is best fitted for tabular data [29]. Instead of simply following the existing work in blast loading prediction, where MLP is commonly used, we conducted a rigorous comparative experiment to systematically evaluate the accuracy of four well-known machine learning models for model selection. The candidate models include: Gradient Boosted Decision Tree (GBDT) [30], Multi-Layer Perceptron (MLP) [31], Residual Network (ResNet) [19], and Transformer Network [20].

3.1. Candidate models

The ensemble of decision trees, such as Gradient Boosted Decision Tree (GBDT), is widely used by machine learning practitioners. They

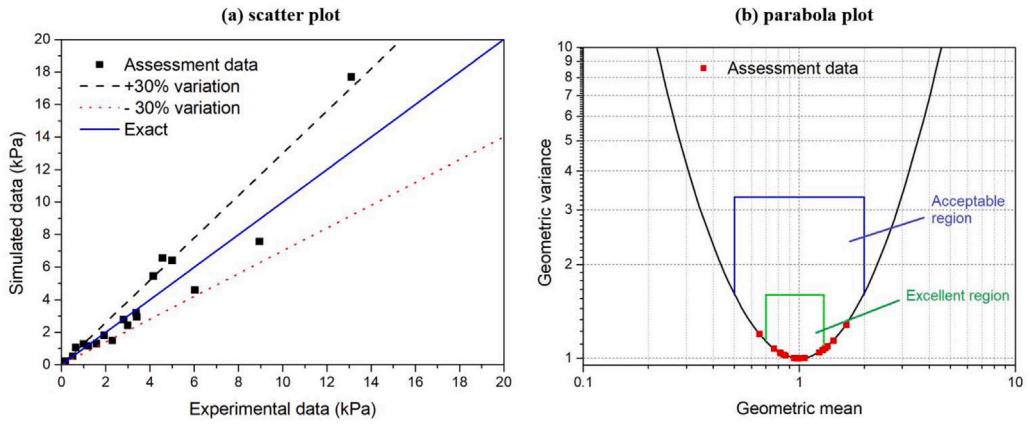


Fig. 2. The comparison of FLACS simulation results and experimental data [15].

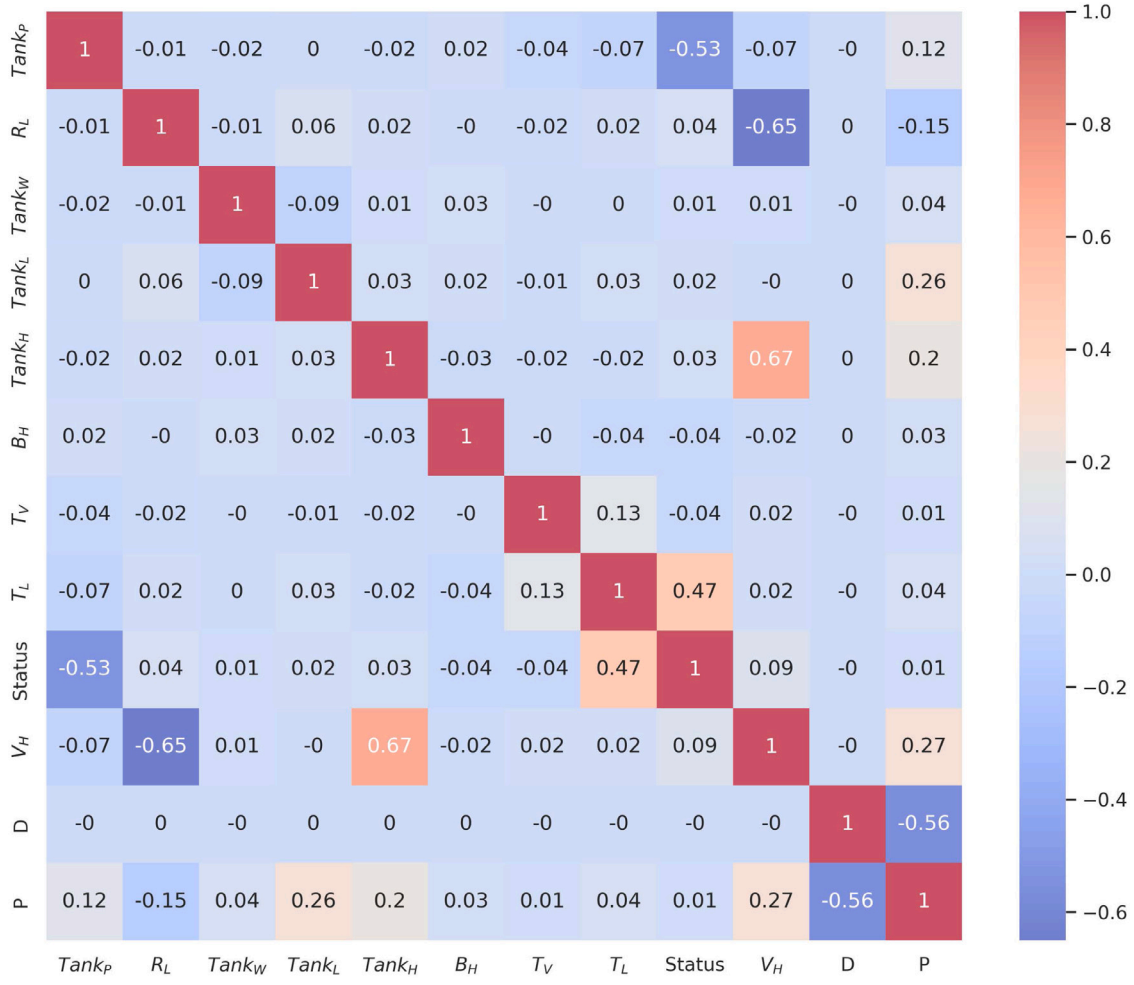


Fig. 3. The correlation matrix between input and output variables in the simulated data.

are still the first choice of many machine learning competition winners due to their simplicity and superior performance. There are several GBDT implementations, including XGBoost [32], lightGBM [33], CatBoost [34], among which the main difference is the processing speed [35]. The lightGBM is used in this study for its efficiency.

The Multi-Layer Perceptrons (MLP) is a standard feed-forward fully-connected neural network which are currently used as the de facto machine learning models in blast engineering, often referred to as neural networks (NNs) or artificial neural networks (ANNs) in the literature of blast engineering. However, it is found out in this study that this default choice is not optimal. The MLP architecture is formulated in Eq. (1):

$$\begin{aligned} \texttt{MLP}(x) &= \texttt{Linear}\,(\texttt{MLPBlock}(...(\texttt{MLPBlock}(x)))) \\ \texttt{MLPBlock}(x) &= \texttt{Dropout}\,(\texttt{ReLU}\,(\texttt{Linear}(x))), \end{aligned} \tag{1}$$

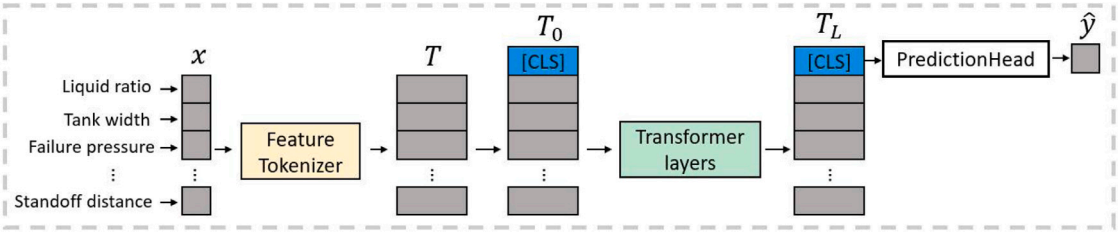


Fig. 4. The architecture of the FT-Transformer. The Feature Tokenizer transforms features to embeddings. The Transformer updates embeddings layer-by-layer. The final embedding of the [CLS] token is used for prediction.

where ReLU is used as the non-linear activation function and Dropout [36] is applied after it as a regularisation.

The Residual Network (ResNet) is proposed in the context of convolutional neural networks (CNNs) for image recognition, where the key idea is to add a skip connection between CNN blocks. ResNet is the most widely used CNN and has been used as the default model for many computer vision tasks. A simple ResNet variant [29] is employed in this study, where the convolution layers are replaced with linear layers to deal with tabular data. Therefore, ResNet in this study can be thought of as a MLP with skip connections and batch normalisations [37]. It is shown later in the ablation study (Section 4.3) that the skip/residual connection is one of the critical designs for the training of deep neural networks. The ResNet architecture is formulated in Eq. (2):

$$\begin{aligned}
&\texttt{ResNet}(x)\\
&= \texttt{PredictionHead}(\texttt{ResNetBlock}(\ldots(\texttt{ResNetBlock}(\texttt{Linear}(x)))))\\
&\texttt{ResNetBlock}(x)\\
&= x + \texttt{Dropout}(\texttt{Linear}(\texttt{Dropout}(\texttt{ReLU}(\texttt{Linear}(\texttt{BatchNorm}(x))))))\\
&\texttt{PredictionHead}(x) = \texttt{Linear}(\texttt{ReLU}(\texttt{BatchNorm}(x))).
\end{aligned} \tag{2}$$

Transformers are novel NN models based on the attention mechanism [20], which have shown state-of-the-art performances on both nature language processing [38] and computer vision [39]. Transformers work on feature embeddings and are not directly applicable to tabular data. There are a few attempts on adapting Transformer networks to tabular data, such as TabFormer [40] and FeatureTokenizer (FT)-Transformer [29], in which the key is to convert features to embeddings. The FT-Transformer is adopted in this work and can be roughly formulated as:

$$\begin{aligned}
&\texttt{Transformer}(x)\\
&= \texttt{PredHead}(\texttt{TFBlock}(\ldots(\texttt{TFBlock}(\texttt{AppendCLS}(\texttt{FT}(x))))))\\
&\texttt{TFBlock}(x) = \texttt{ResidualBlock}(\texttt{FFN}, \texttt{ResidualBlock}(\texttt{MHSA}, x))\\
&\texttt{ResidualBlock}(\texttt{Module}, x) = x + \texttt{Dropout}(\texttt{Module}(\texttt{Norm}(x)))\\
&\texttt{FFN}(x) = \texttt{Linear}(\texttt{Dropout}(\texttt{ReLU}(\texttt{Linear}(x))))\\
&\texttt{PredHead}(x) = \texttt{Linear}(\texttt{ReLU}(\texttt{LayerNorm}(x_{\texttt{CLS}})))
\end{aligned} \tag{3}$$

Specifically, the FT-Transformer converts all features to embeddings (tokens), appends a [CLS] token, and applies a stack of Transformer layers (Multi-Head Self-Attention, MHSA [20]) to the embeddings. The final prediction is obtained by applying a PredHead (prediction head) on the [CLS] token. The overall architecture is given in Fig. 4. The key components, Feature Tokenizer and Transformer layer, are presented in Fig. 5. Note that categorical features are converted to numeric using the one-hot encoding method, and the Feature Tokenizer takes numeric features as inputs and converts them to d dimensional embeddings with trainable linear projections.

3.2. Implementation details

Data preparation. The simulated BLEVE dataset (46000 instances) is randomly split into a training set (80%) and a testing set (20%), and 20% of the training data is further split out as a validation set for hyperparameter tuning. Data pre-processing is critical, especially for neural networks-based models. The "Standardisation" transformation is applied to the input data to convert each feature to mean 0 and standard deviation 1. It is observed that the original target is highly skewed as shown in Fig. 6 and the "Quantile" transformation from the Scikit-learn library [41] is applied to convert the target to be the standard normal distribution (Fig. 6), which is shown to improve the performance (Table 4). Note that the quantile transformation generates invalid negative pressures and it is important to apply inverse quantile transformation to the model prediction to convert them back to positive values.

Table 3
The hyperparameter tuning space of all models.

Model	Hyperparameter	Search space	Distribution
lightGBM	number of leaves	[10, 100]	int uniform
	learning rate	[1e−3, 1.0]	log uniform
	minimum child weight	[1e−5, 1e−1]	log uniform
	minimum child sample	[2, 100]	int uniform
	data sampling fraction	[0.5, 1.0]	uniform
	feature sampling fraction	[0.5, 1.0]	uniform
	regularisation lambda	[1e−5, 1.0]	log uniform
MLP	learning rate	[1e−5, 1e−2]	log uniform
	weight decay	[1e−6, 1e−3]	log uniform
	number of layers	[1, 5]	int uniform
	number of neurons	[1, 512]	int uniform
	dropout probability	[0.0, 0.5]	uniform
ResNet	learning rate	[1e−5, 1e−2]	log uniform
	weight decay	[1e−6, 1e−3]	log uniform
	number of layers	[1, 8]	int uniform
	number of neurons	[64, 512]	int uniform
	dropout probability	[0.0, 0.5]	uniform
Transformer	learning rate	[1e−5, 1e−2]	log uniform
	weight decay	[1e−6, 1e−3]	log uniform
	number of layers	[1, 5]	int uniform
	token dimensions	[64, 512]	int uniform
	dropout probability	[0.0, 0.5]	uniform
	attention scaling factor	[1, 4]	int uniform

Hyperparameter tuning. The performance of a machine learning model depends heavily on an optimal set of hyperparameters. For a fair comparison, an advanced hyperparameter searching strategy based on the Bayesian optimisation is utilised, which is reported to be better than the grid search and random search [42]. The implementation is from the Optuna library [43] using the Tree-Structured Parzen Estimator algorithm. The specification of hyperparameter tuning is summarised in Table 3. Although the number of hyperparameters varies between models, the tuning budget for each model is fixed to 100 trials, i.e., 100 combinations of hyperparameter values are examined and the set of hyperparameter giving the best result on the validation set is chosen.

Training protocol. The lightGBM is trained for a maximal iteration of 2000 with early stopping patience of 50 epochs, i.e., the training will be stopped if the validation performance does not improve in the last 50 epochs. For all three NN models, i.e., MLP, ResNet, and Transformer, the training is not stopped until the validation performance does not

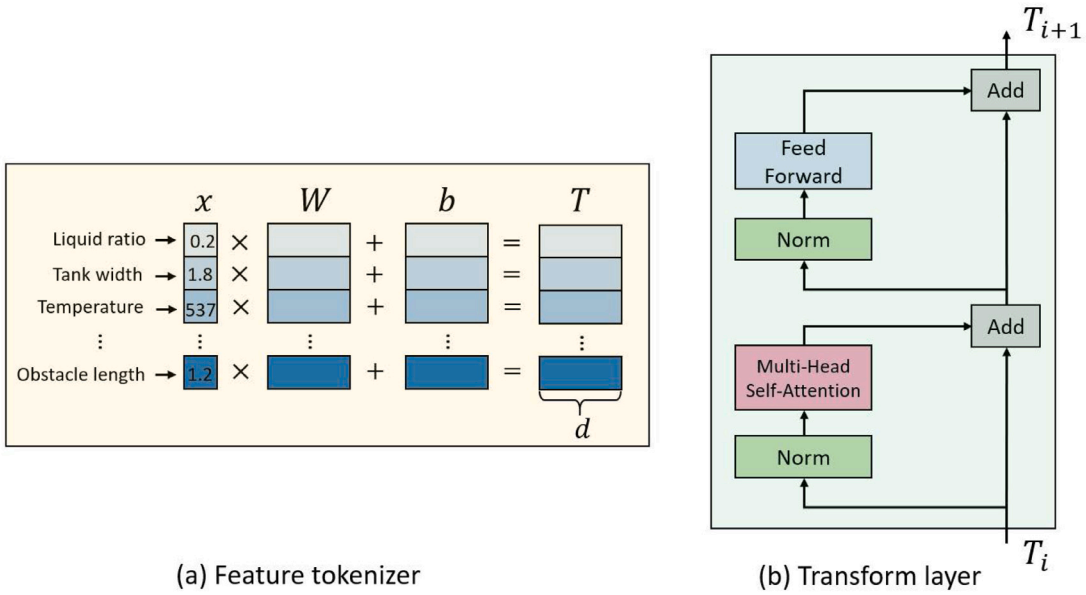


Fig. 5. (a) The Feature tokenizer converts singleton numeric features to d dimensional feature embeddings (tokens) via trainable linear projections. (b) The Transformer layer updates tokens using the multi-head self-attention module.

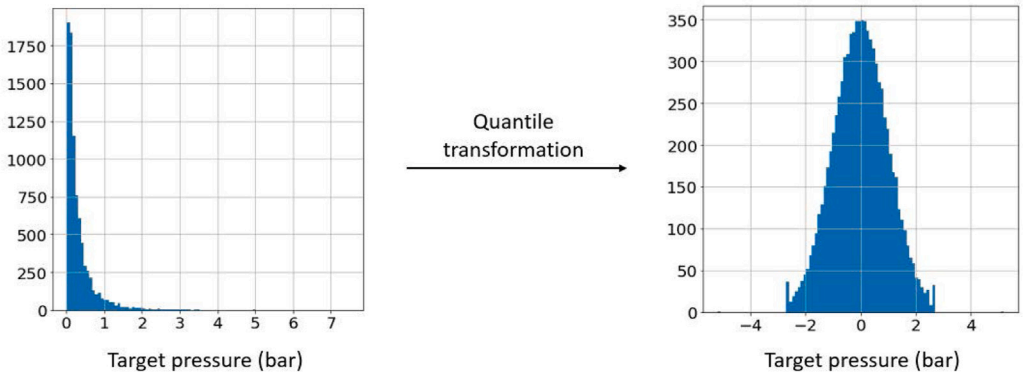


Fig. 6. The effect of quantile transformation on the target pressure.

improve in the last 16 epochs. The AdamW solver [44] with a batch size of 512 is used for the optimisation of all NN models.

Evaluation metrics. The output BLEVE peak pressure is a continuous quantity and its prediction is a typical regression problem. It is hard to tell if a prediction is incorrect for regression problems. Only the difference (error) between the prediction and the ground truth can be quantified with certain regression metrics. Available metrics quantify the error from different perspectives and they can be inconsistent in some cases (Fig. 9). For a thorough comparison of the models, three evaluation metrics are adopted, including Root Mean Square Error (RMSE), Mean Absolute Percentage Error (MAPE), and Coefficient of Determination (or R-squared, R^2).

RMSE is the standard deviation of the prediction error, which measures the "spreadness" of errors around the best fitting line (hyperplane). Given the ground truth y and the prediction $\hat{y}$, RMSE is defined as:

$$\text{RMSE} = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(y_i - \hat{y}_i)^2}, \tag{4}$$

where N is the number of data points. RMSE ranges from 0 to infinity, and the smaller the better.

MAPE measures the relative error between the prediction and the ground truth, and it is defined as a ratio as follows:

$$\text{MAPE} = \frac{1}{N}\sum_{i=1}^{N}\left|\frac{y_i - \hat{y}_i}{y_i}\right|. \tag{5}$$

MAPE also ranges from 0 to infinity, and the smaller the better.

While RMSE and MAPE quantify the error from a local point of view, R^2 measures the error from a holistic distribution view, where it expresses the degree of linear correlation between the ground truth and the prediction. R^2 is defined as:

$$R^2 = 1 - \frac{\sum_{i=1}^{N}(y_i - \hat{y}_i)^2}{\sum_{i=1}^{N}(y_i - \tilde{y})^2}, \tag{6}$$

where $\tilde{y} = \sum_{i=1}^{N} y_i$ is the mean of the ground truth. R^2 normally ranges from 0 to 1 and the larger the better, but it can be negative when the model is even worse than constantly predicting $\tilde{y}$.

3.3. The choice of loss function

The loss function plays a critical role in the training of neural network (NN) models. Existing applications of NN-based blast loading prediction have adopted MSE [13] and MAPE [15] separately without a justification of choice. A comparative experiment is performed in this

Table 4
The effect of target preprocessing and loss function.

Loss function	Metric	Target preprocessing		
		None	Standardisation	Quantile
MSE	RMSE ($\times 10^{-2}$) ↓	2.24	2.07	**1.95**
	MAPE (%) ↓	4.58	4.43	**3.60**
	R^2 (%) ↑	98.6	99.2	**99.7**
MAPE	RMSE ($\times 10^{-2}$) ↓	2.51	7.32	34.5
	MAPE (%) ↓	3.63	5.57	102
	R^2 (%) ↑	99.1	95.0	−11.7

study to analyse the effect of loss functions for blast loading prediction. In particular, the loss function is studied in conjunction with the target preprocessing, as the latter has a significant impact on the scale of the former. Specifically, three target transformations are considered, including (1) "none", which means no processing; (2) "standardisation" which converts the ground truth to be mean 0 and standard deviation 1; (3) "Quantile" which converts the target distribution to be the standard Gaussian. The Transformer model is used in this experiment and it is trained with MSE or MAPE loss function and evaluated using the validation performance for peak pressure prediction. The results are shown in Table 4.

Several conclusions can be drawn from Table 4. (1) The target preprocessing matters. When used with the MSE loss function, preprocessing of the ground truth improves the performance consistently over all three metrics, and the rank is *Quantile* > *Standardisation* > *None*; (2) The MAPE loss function works reasonably well without any target preprocessing, and this is consistent with [15]. In fact, MAPE could be negatively impacted by the target preprocessing, e.g., using MAPE with quantile transformation leads to a relative error larger than 100% and a negative R^2 score; (3) MSE is more stable and the best result is achieved using MSE with the quantile transformation. This combination of MSE with quantile transformation of target is hence used for the subsequent experiments.

3.4. Comparison of model performances on the test set

With the experimental setting and tuning specified in Section 3.2, the best hyperparameters are obtained for lightGBM, MLP, ResNet, and Transformer. Models are trained with their best configurations on the training set (using the validation set for early stopping) and evaluated on the test set. To remove the effect of randomness, 10 runs are performed for each model with the same configuration but different random seeds, and the average performance with standard deviation is reported. The MLP* model studied in [15] is also added for comparison. Note that the main differences between MLP in this work and MLP* in [15] are the usage of target transformation and optimised hyperparameters.

Following [45], a graphical visualisation using the Taylor diagram is presented in Fig. 7 to visually compare the investigated machine learning models. In the Taylor diagram, each model is represented by a point and a better model should be closer to the reference point on the x-axis. Fig. 7 shows the performance of five trained machine learning models on two random testing sets, and it can be seen that Transformer achieves the best accuracy.

Fig. 8 shows the model performances in terms of RMSE, MAPE, and R^2. The ranking of the five models is consistent in terms of all three metrics: Transformer > ResNet ≈ MLP > MLP* > lightGBM. First of all, all NN-based approaches outperform the ensemble tree lightGBM, indicating the advantage of utilising deep learning for BLEVE blast prediction. MLP achieves clearly better results than the existing MLP* model [15], which shows the necessity of target transformation and hyperparameter tuning. ResNet achieves negligible improvements compared to MLP and it is hypothesised that there are too few layers for the residual connection to have an effect (ResNet is tuned to have three layers only). The Transformer model achieves the best performance and outperforms all other models with a clear margin, e.g., it achieves on average 3.5% relative error (MAPE) which is 1.1% lower than the second-best (ResNet 4.6%). Given the simple (open) environment setting and the saturated performance, this is considered a significant improvement. The advantage of Transformer is expected to be greater in complex explosion scenarios, e.g., with obstacles. The configuration of the optimal Transformer model is summarised in Table 5.

Table 5
The final configuration of the Transformer model.

Hyperparameter	Value
learning rate	1e−4
weight decay	2.38e−6
number of layers	4
token dimensions	72
Dropout probability	0.04
attention scaling factor	2.8
Batch size	512
Loss function	MSE
Optimiser	AdamW
Attention heads	8
Activation function	ReGLU

4. Analysis of the transformer performance

For a better understanding of the model performance, an in-depth analysis of the Transformer prediction is conducted. With the optimal configuration presented in Table 5, the Transformer model is retrained for BLEVE peak pressure prediction. The training strategy is the same as the one used for hyperparameter tuning and model selection, that is, the model is trained on the training set with early stopping by monitoring its performance on the validation set. If the model performance does not improve for 16 epochs continuously, the training is stopped and the trained model is applied to the testing set to obtain the final prediction.

4.1. Scatter plots for a holistic overview

The scatter plot between the simulated target and the model's prediction is shown in Fig. 9, where x-axis and y-axis represent the target and the prediction, respectively. The red dashed line represents a perfect model, i.e., the prediction matches the target exactly. The yellow dashed line indicates the envelop of 30% relative error. Blue points denote test data, and a linear regression line (blue) with a 95% confidence interval that best fits these points is also shown. Quantitative performance measurements, including MAPE and R^2, are shown on each plot.

Fig. 9 presents the scatter plots for BLEVE pressure prediction. Fig. 9(a) covers the entire range of pressure (0 to 500 kPa) presented in the test data. It is observed that overall, the Transformer model's prediction is fairly accurate qualitatively and quantitatively. The points are well scattered around the exact match line (red), and the regression line (blue) is almost identical to the red line with no bias to over-prediction or under-prediction. Fig. 9(b) shows the scatter plot for testing data with the target peak pressure ranging from 0 to 30 kPa. This subset accounts for 80% of the testing data and the performance is aligned qualitatively and quantitatively with the overall performance shown in Fig. 9(a). Fig. 9(c) shows the scatter plot for testing data ranging from 30 kpa to 100 kPa. Points in this range are more visually spread out, which is demonstrated by the decrease of R^2 value from 99.6% to 98.5%. An interesting observation is that the corresponding MAPE value, on contrary, improved from 3.56% to 2.98%. The inconsistency comes from the fact that they measure the regression performance from different perspectives: R^2 focuses on the global alignment whereas MAPE measures the local fitness between the target and the prediction.

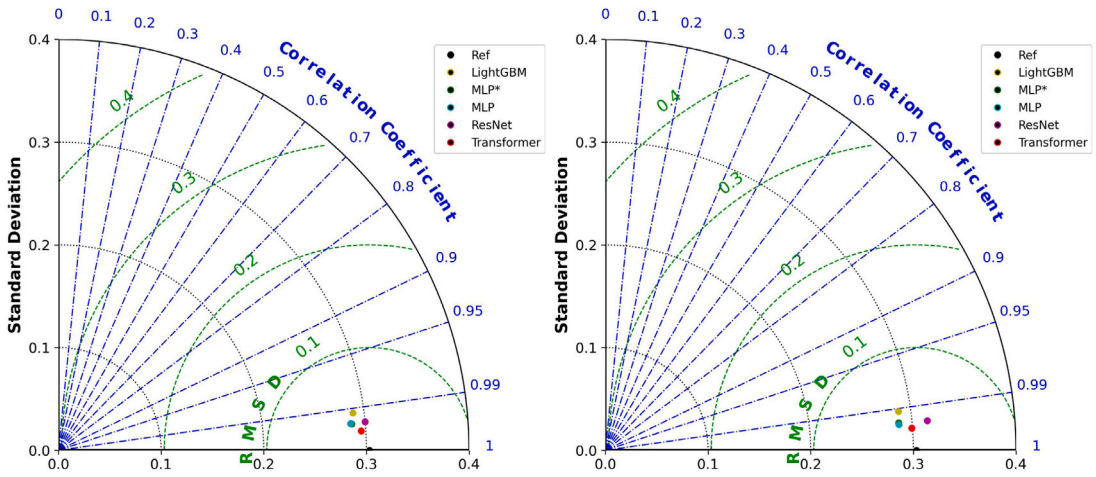


Fig. 7. Performance visualisation on two random testing sets using the Taylor diagram.

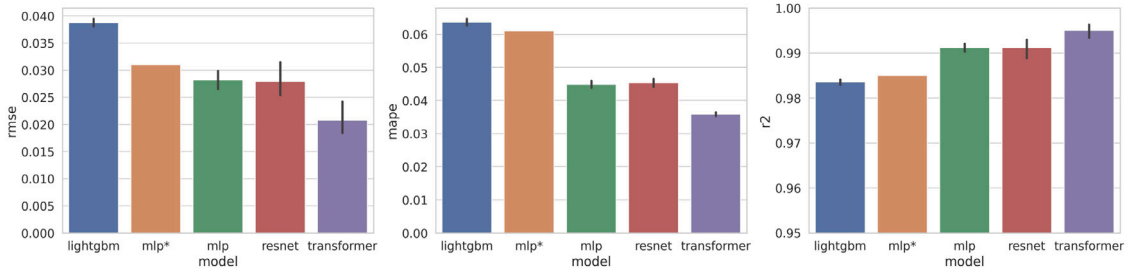


Fig. 8. The comparison of model performances on the test set. Note that for RMSE and MAPE the smaller the better, while for R^2 the larger the better.

While R^2 is more consistent with the "visual spreadness" of the scatter plot, MAPE is much easier to interpret and therefore more suitable for engineering practices. Fig. 9(d) presents the scatter plot for testing data ranging from 100 kpa to 500 kPa. Predictions in this range are slightly worse compared to previous ones as shown by the MAPE and R^2 values. The regression line indicates a slight under-prediction in the range of 400 kpa to 500 kPa. Nevertheless, these predictions are still fairly accurate, achieving a relative error of 3.71% (compared to 4.42% error of the MLP* model studied in [15]). As analysed in the next section, data in this range (100 kpa to 500 kPa) often have a small standoff distance, i.e., near-field data, which are known to be the hardest part of blast loading prediction [1,9].

4.2. Error breakdown with respect to the Standoff distance

The standoff distance to the BLEVE source is a key factor in blast prediction. To thoroughly investigate the behaviour of the Transformer model to the distance factor, the testing set is separated into five subsets according to the distance D. For each subset, the ratios of data with the prediction error (MAPE) less than 1%, 3%, 5%, and 7% are counted to show a detailed breakdown of the prediction error.

Table 6 presents the result of error breakdown. There are no clear variations between the model's performance in different distance ranges except for the "near-field" group: the model performs slightly worse when the standoff distance is less than 10 m. Note that the near-field is obviously where the large peak pressures appear, so the observation is consistent with the result in Fig. 9 that the prediction is slightly worse when the pressure is large. Overall, around 20% predictions have a relative error of less than 1%, more than 50% predictions have less than 3% error and more than 90% have less than 7% error. At the level of ≥7% relative error, the Transformer model performs equally well for all standoff distances from 5 to 50 m. Note that the simulated data has a maximum standoff distance of 50 m, but it is shown later in Section 5 with BLEVE field test data that the Transformer model trained with the simulated data can extrapolate beyond this range, producing accurate peak pressure prediction even at 100 m.

4.3. Ablation study of the transformer model

Next, an ablation study is performed to evaluate the design choice of the developed Transformer model. In the hyperparameter-tuning experiment (Section 3.2), several import hyperparameters (see Table 3) have been tuned to optimal values without showing the sensitivity of the model. Yet, there are other hyperparameters and design choices of the Transformer model that have not been evaluated. Using the optimal configuration (shown in Table 5) as the default choice, a controlled experiment is conducted to show the significance of each design choice. That is, while keeping all others as the default value, the model is trained multiple times with one varying parameter and the performance on the test set (in terms of MAPE) is reported. Results are shown in Fig. 10 and these parameters are grouped into data-related, model-related, and training-related for the sake of discussion.

Categorical embedding (Fig. 10(a)) is a data-related design choice, concerning how to convert categorical features to tokens on which the attention module [20] applies. Setting it to *True* means a look-up embedding table will be used to map each categorical value to an embedding. Setting it to *False* means categorical features are transformed to numerical features simply using "one-hot-encoding" and embedded as tokens using the numerical feature tokenizer described in Fig. 5(a). No improvements are observed by utilising categorical embedding, hence it is turned off for simplicity. **Data preprocessing** (Fig. 10(b)) is another data-related design that deals with the scale and distribution of input data. It is found the widely-used "standardisation" (mean 0, standard deviation 1) offers the best performance for data preprocessing.

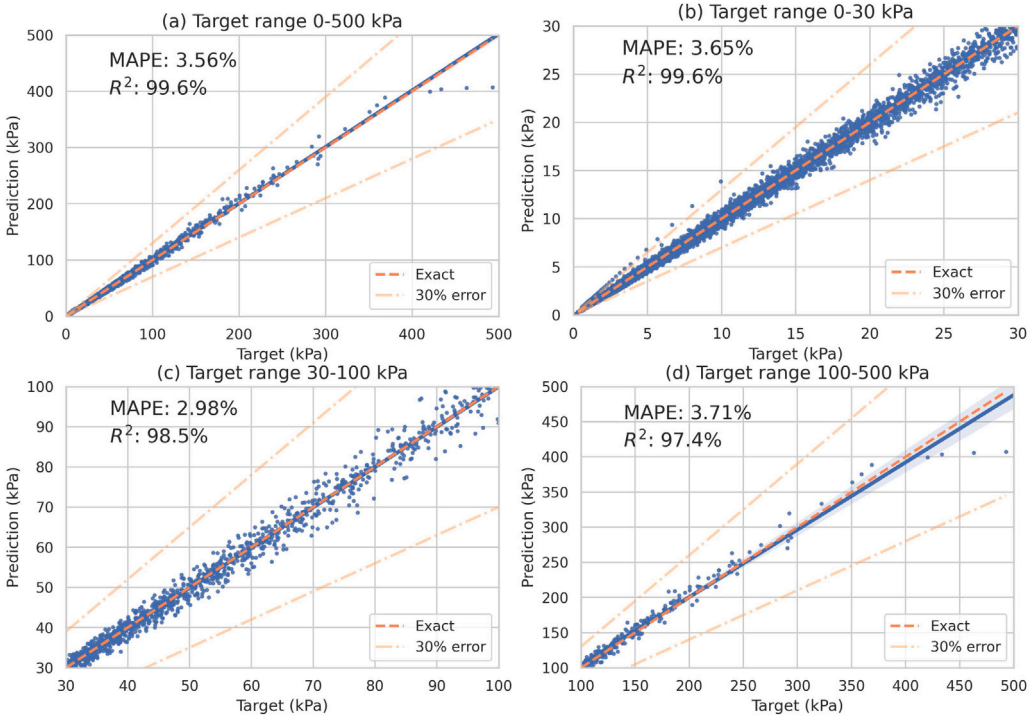


Fig. 9. The scatter plots of the Transformer model prediction on the test set with respect to various target ranges. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

Table 6
The Transformer model error breakdown with respect to the standoff distance.

Distance (m)	Number of data	Subset Error (%)	Data ratio within relative error (%)			
			E ≤ 1%	E ≤ 3%	E ≤ 5%	E ≤ 7%
5 < D ≤ 10	1194	4.0	18.8	50.1	70.3	94.5
10 < D ≤ 20	1998	3.5	21.1	57.6	77.5	95.6
20 < D ≤ 30	2000	3.4	24.4	60.4	78.5	94.4
30 < D ≤ 40	2000	3.4	27.0	59.0	78.2	94.7
40 < D ≤ 50	2000	3.7	25.4	56.4	77.7	94.3

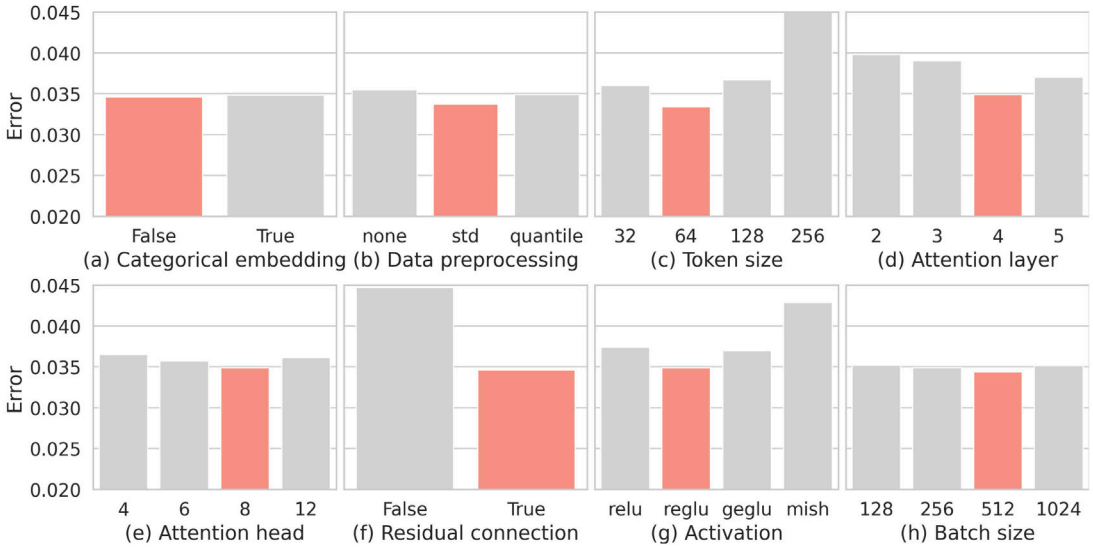


Fig. 10. The ablation study of the Transformer model.

Five model-related design choices are studied. **Token size** (Fig. 10(c)) is the dimension of feature embedding and it is found that a moderate size of 64 is enough for BLEVE pressure prediction. **Attention layer** (Fig. 10(d)) controls the depth of the Transformer model and is the key factor to the model capacity and complexity. It is found that a four-layer Transformer model works the best. Actually, token size and attention layer are two hyperparameters that have been automatically tuned to be 72 and 4 in Section 3.2, and they are added in the ablation

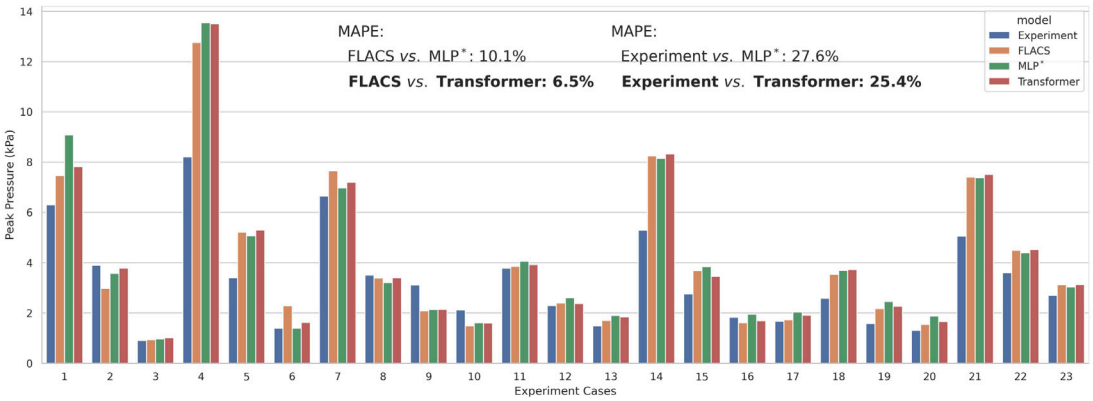


Fig. 11. Results of the peak pressure on 23 BLEVE experiment cases.

study to verify the effectiveness of the hyperparameter-tuning library. The large variation of the model performance with respect to token size and attention layer also indicates the necessity of tuning these hyperparameters. **Attention head** (Fig. 10(e)) is the number of parallel attention computation in each attention layer, and it is found the model is not sensitive to this number and the default 8 heads works the best. **Residual connection** (Fig. 5(f)) is the generic skip-connection introduced in ResNet [19] which is proven to be extremely beneficial to the optimisation of deep neural networks by providing robust gradient flows. By adding residual connections between every two consecutive attention layers, a significant improvement is observed. Note that the improvement is measured with a four-layer Transformer model and it is expected to be more significant for more complex problems using deeper networks. **Activation** (Fig. 5(g)) handles the non-linearity of neural networks and it is confirmed that GLU variants work well with Transformer models as reported in the original paper [46].

Batch size (Fig. 5(h)) is a deep network training-related hyperparameter, which represents the number of data points used in each iteration of network training. The model is not sensitive to this hyperparameter and using a large batch size reduces the training iteration needed for convergence, which results in faster training overall.

In summary, apart from hyperparameters tuned in Table 3 (which are common for all deep neural networks) and the residual connection studied above, the developed Transformer model is robust against other design choices.

5. Prediction on real experimental data

The established Transformer model produces accurate peak pressure predictions on simulated BLEVE data. It is of interest to investigate how the model performs on real BLEVE experimental data. 8 BLEVE experiments conducted by Johnson et al. [25] and Birk et al. [24,47] are used as testing data, each of which has the peak pressure measurement at various standoff distances as shown in Table 7. It is worth noting that these cases have no overlapping with the data used for FLACS validation in Section 2.1, which means they are completely unseen for both the CFD simulator and the Transformer model.

Fig. 11 presents a case-by-case visualisation of the peak pressure values produced by experimental measurement, FLACS simulation, MLP* model [15], and Transformer model. Overall, the machine learning models (MLP* and Transformer) achieve promising results, compared to both the experimental tests and FLACS simulations. Using FLACS simulation as the ground truth, MLP* and Transformer achieve 10.1% and 6.5% relative error, respectively. When using experiment recording as the ground truth, MLP* and Transformer achieve 27.6% and 25.4% relative error. As a reference, the empirical method presented in [48] achieves an error rate of over 70% and the energy-based method studied by [49] generates around 50% error, which demonstrates the potential of machine learning-based approaches for blast loading prediction. Moreover, the developed Transformer model outperforms the MLP significantly, showing that the state-of-the-art neural networks should be adopted for blast loading prediction, replacing the currently widely used MLP.

Table 7
Summary of BLEVE experimental testing data.

Experiments	Fluid	Liquid Status	Distance (m)	Case No.
BG-1 (Johnson)	Butane	Superheated	25, 50, 100	1, 2, 3
BG-4 (Johnson)	Butane	Saturated	25, 50, 100	4, 5, 6
01-01 (Birk)	Propane	Saturated	10, 20, 30, 40	7, 8, 9, 10
01-02 (Birk)	Propane	Superheated	20, 30, 40	11, 12, 13
01-03 (Birk)	Propane	Superheated	10, 20, 40	14, 15, 16
01-04 (Birk)	Propane	Superheated	40	17
01-05 (Birk)	Propane	Superheated	20, 30, 40	18, 19, 20
02-04 (Birk)	Propane	Superheated	20, 30, 40	21, 22, 23

Looking at the prediction of individual cases, it is observed that machine learning models do not align very well with some experiments, e.g., “Experiment cases 4–6 from BG-4 (Johnson)”. Note that FLACS produces relatively large errors in those cases, leading to the similar performance of its surrogate learning models. On the contrary, there are also cases in which machine learning models generate better predictions for the experiment recording compared to FLACS, such as cases 7 and 10. More importantly, in cases 3 and 6 where the standoff distance is 100 meters (while the distance range of training data is 5 m to 50 m), the learning models demonstrate the capability of extrapolating beyond the training domain. The generalisation indicates that these machine learning models have the ability to learn certain implicit knowledge about the blast wave propagation instead of merely ‘‘memorising’’ data.

6. Conclusion

This study challenges the common practice of using the multi-layer perceptron (MLP) for blast loading prediction by conducting a comparative study of four well-known machine learning models, including lightGBM, MLP, ResNet, and Transformer. It is demonstrated that the Transformer network achieves the best performance, outperforming MLP for BLEVE peak pressure prediction with a clear margin.

1000 BLEVE cases are simulated based on the validated CFD model in FLACS, yielding a full dataset of 46000 instances for the training and evaluation of machine learning models. With a rigorous experiment setting, these models are carefully tuned with an advanced hyperparameter searching strategy to achieve their optimal performances. It is established that Transformer significantly outperforms other competitors, predicting BLEVE peak pressure with a relative error of 6.5% and R^2 value of 0.997 when compared to the FLACS simulation result. The trained Transformer model is also applied to 23 BLEVE field test cases, achieving a relative error of 25.4%, which is better than 27.6%

obtained by a tailored MLP from the previous study. In addition, an ablation study of Transformer is conducted to show the significance of critical design choices, such as the residual connection and token size.

Furthermore, the Transformer neural network architecture is tested for BLEVE peak pressure prediction on the real free field experimental data, with satisfactory results achieved. It demonstrates the capability of Transformer for BLEVE loading prediction. It is believed that Transformer can be applied to the prediction of blast loads of other explosives, and its advantage over the MLP baseline will be even larger in complex scenarios, such as confined environments and cityscapes.

The future research directions include: (a) extend the Transformer model to complex environments, (b) extend the Transformer model to multi-output prediction, such as impulse and blast wave duration, and (c) interpret the inference mechanism of the black-box machine learning models in terms of input–output relationships to derive easy-to-use formula.

CRediT authorship contribution statement

Qilin Li: Conception and design of study, Acquisition of data, Writing – original draft, Approval of the final version of the manuscript. **Yang Wang:** Acquisition of data, Writing – original draft, Approval of the final version of the manuscript. **Yanda Shao:** Analysis of data, Writing – original draft, Approval of the final version of the manuscript. **Ling Li:** Conception and design of study, Writing – review & editing, Approval of the final version of the manuscript. **Hong Hao:** Conception and design of study, Writing – review & editing, Approval of the final version of the manuscript.

Declaration of competing interest

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

Data availability

Data will be made available on request.

References

[1] Hao H, Hao Y, Li J, Chen W. Review of the current practices in blast-resistant analysis and design of concrete structures. Adv Struct Eng 2016;19(8):1193–223.
[2] Pope DJ. The development of a quick-running prediction tool for the assessment of human injury owing to terrorist attack within crowded metropolitan environments. Philos Trans R Soc B 2011;366(1562):127–43.
[3] Yokohama H, Sunde J, Ellis-Steinborner ST, Ayubi Z. Vehicle borne improvised explosive device (VBIED) characterisation and estimation of its effects in terms of human injury. Int J Prot Struct 2015;6(4):607–27.
[4] Li J, Hao H, Wu C. Numerical study of precast segmental column under blast loads. Eng Struct 2017;134:125–37.
[5] Rigby S, Akintaro O, Fuller B, Tyas A, Curry R, Langdon G, et al. Predicting the response of plates subjected to near-field explosions using an energy equivalent impulse. Int J Impact Eng 2019;128:24–36.
[6] US Department of Defense. Structures to resist the effects of accidental explosions. Tech. rep., Washington, DC, 2008.
[7] Baker WE, Cox P, Kulesz J, Strehlow R, Westine P. Explosion hazards and evaluation. Elsevier; 1983.
[8] Kingery CN, Bulmash G, US Army Ballistic Research Laboratory. Air blast parameters from TNT spherical air burst and hemispherical surface burst. Tech. rep., Aberdeen, MD, 1984.
[9] Rigby SE, Tyas A, Fay SD, Clarke SD, Warren JA. Validation of semi-empirical blast pressure predictions for far field explosions-is there inherent variability in blast wave parameters? In: Proceedings of the 6th international conference on protection of structures against hazards. Sheffield; 2014.
[10] Smith PD, Rose TA. Blast wave propagation in city streets—an overview. Prog Struct Eng Mater 2006;8(1):16–28.
[11] Li J, Hernandez F, Hao H, Fang Q, Xiang H, Li Z, et al. Vented methane-air explosion overpressure calculation—A simplified approach based on CFD. Process Saf Environ Prot 2017;109:489–508.
[12] Li J, Abdel-jawad M, Ma G. New correlation for vapor cloud explosion overpressure calculation at congested configurations. J Loss Prev Process Ind 2014;31:16–25.
[13] Dennis AA, Pannell JJ, Smyl DJ, Rigby SE. Prediction of blast loading in an internal environment using artificial neural networks. Int J Prot Struct 2021;12(3):287–314.
[14] Remennikov AM, Rose TA. Predicting the effectiveness of blast wall barriers using neural networks. Int J Impact Eng 2007;34(12):1907–23.
[15] Li J, Li Q, Hao H, Li L. Prediction of BLEVE blast loading using CFD and artificial neural network. Process Saf Environ Prot 2021;149:711–23.
[16] Bui X-N, Nguyen H, Le H-A, Bui H-B, Do N-H. Prediction of blast-induced air over-pressure in open-pit mine: Assessment of different artificial intelligence techniques. Nat Resour Res 2020;29(2):571–91.
[17] LeCun Y, Bengio Y, Hinton G. Deep learning. Nature 2015;521(7553):436–44.
[18] Goodfellow I, Bengio Y, Courville A. Deep learning. MIT Press; 2016.
[19] He K, Zhang X, Ren S, Sun J. Deep residual learning for image recognition. In: Proceedings of the IEEE conference on computer vision and pattern recognition. 2016, p. 770–8.
[20] Vaswani A, Shazeer N, Parmar N, Uszkoreit J, Jones L, Gomez AN, et al. Attention is all you need. Adv Neural Inf Process Syst 2017;30.
[21] Eckhoff RK. Boiling liquid expanding vapour explosions (BLEVEs): A brief review. J Loss Prev Process Ind 2014;32:30–43.
[22] Launder BE, Spalding DB. The numerical computation of turbulent flows. In: Numerical prediction of flow, heat transfer, turbulence and combustion. Elsevier; 1983, p. 96–116.
[23] Gexcon AS. FLACS v10.7 user's manual, Norway. 2017.
[24] Birk AM, VanderSteen JDJ. On the transition from non-BLEVE to BLEVE failure for a 1.8 m 3 propane tank. 2006.
[25] Johonson DM, Pritchard M. Large scale experimental study of boiling liquid expanding vapour explosions (BLEVEs). In: The 14th international LNG/LPG conference & exhibition. 1991.
[26] Lemmon EW, Huber ML, McLinden MO. NIST reference fluid thermodynamic and transport properties–REFPROP. NIST Stand Ref Database 2007;23:v8.
[27] Li J, Hao H. Numerical study of medium to large scale BLEVE for blast wave prediction. J Loss Prev Process Ind 2020;65:104107.
[28] Kadatec. LPG road tankers & ISO tank containers, Czech Republic. 2017.
[29] Gorishniy Y, Rubachev I, Khrulkov V, Babenko A. Revisiting deep learning models for tabular data. Adv Neural Inf Process Syst 2021;34.
[30] Freund Y, Schapire R, Abe N. A short introduction to boosting. J Japan Soc Artif Intell 1999;14(771–780):1612.
[31] Rumelhart DE, Hinton GE, Williams RJ. Learning internal representations by error propagation. Parallel Distrib Process: Explor Microstruct Cogn 1986;1:318–62.
[32] 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. 2016, p. 785–94.
[33] Ke G, Meng Q, Finley T, Wang T, Chen W, Ma W, et al. LightGBM: A highly efficient gradient boosting decision tree. Adv Neural Inf Process Syst 2017;30.
[34] Dorogush AV, Ershov V, Gulin A. CatBoost: Gradient boosting with categorical features support. 2018, arXiv preprint arXiv:1810.11363.
[35] Al Daoud E. Comparison between xgboost, LightGBM and CatBoost using a home credit dataset. Int J Comput Inf Eng 2019;13(1):6–10.
[36] Srivastava N, Hinton G, Krizhevsky A, Sutskever I, Salakhutdinov R. Dropout: A simple way to prevent neural networks from overfitting. J Mach Learn Res 2014;15(1):1929–58.
[37] Ioffe S, Szegedy C. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In: International conference on machine learning. PMLR; 2015, p. 448–56.
[38] Devlin J, Chang M-W, Lee K, Toutanova K. Bert: Pre-training of deep bidirectional transformers for language understanding. 2018, arXiv preprint arXiv:1810.04805.
[39] Dosovitskiy A, Beyer L, Kolesnikov A, Weissenborn D, Zhai X, Unterthiner T, et al. An image is worth 16 × 16 words: Transformers for image recognition at scale. 2020, arXiv preprint arXiv:2010.11929.
[40] Padhi I, Schiff Y, Melnyk I, Rigotti M, Mroueh Y, Dognin P, et al. Tabular transformers for modeling multivariate time series. In: ICASSP 2021-2021 IEEE international conference on acoustics, speech and signal processing. ICASSP, IEEE; 2021, p. 3565–9.
[41] Pedregosa F, Varoquaux G, Gramfort A, Michel V, Thirion B, Grisel O, et al. Scikit-learn: Machine learning in Python. J Mach Learn Res 2011;12:2825–30.
[42] Turner R, Eriksson D, McCourt M, Kiili J, Laaksonen E, Xu Z, et al. Bayesian optimization is superior to random search for machine learning hyperparameter tuning: Analysis of the black-box optimization challenge 2020. In: NeurIPS 2020 competition and demonstration track. PMLR; 2021, p. 3–26.
[43] Akiba T, Sano S, Yanase T, Ohta T, Koyama M. Optuna: A next-generation hyperparameter optimization framework. In: Proceedings of the 25th ACM SIGKDD international conference on knowledge discovery & data mining. 2019, p. 2623–31.
[44] Loshchilov I, Hutter F. Decoupled weight decay regularization. 2017, arXiv preprint arXiv:1711.05101.

[45] Bardhan A, Biswas R, Kardani N, Iqbal M, Samui P, Singh M, et al. A novel integrated approach of augmented grey wolf optimizer and ann for estimating axial load carrying-capacity of concrete-filled steel tube columns. Constr Build Mater 2022;337:127454.

[46] Shazeer N. Glu variants improve transformer. 2020, arXiv preprint arXiv:2002.05202.

[47] Birk AM, Davison C, Cunningham M. Blast overpressures from medium scale BLEVE tests. J Loss Prev Process Ind 2007;20(3):194–206.

[48] Laboureur D, Heymes F, Lapebie E, Buchlin J, Rambaud P. BLEVE overpressure: Multiscale comparison of blast wave modeling. Process Saf Prog 2014;33(3):274–84.

[49] Hemmatian B, Planas E, Casal J. Comparative analysis of BLEVE mechanical energy and overpressure modelling. Process Saf Environ Prot 2017;106:138–49.