Journal of Applied Fluid Mechanics, Vol. 16, No. 10, pp. 1951-1961, 2023.
Available online at www.jafmonline.net, ISSN 1735-3572, EISSN 1735-3645.
https://doi.org/10.47176/jafm.16.10.1869

Pressure Drop Prediction in Fluidized Dense Phase Pneumatic Conveying using Machine Learning Algorithms

J. S. Shijo and N. Behera†

VIT Vellore, Vellore, Tamilnadu,632014, India

†Corresponding Author Email: niranjanabehera@vit.ac.in

ABSTRACT

Modeling of pressure drop in fluidized dense phase conveying (FDP) of powders is a tough work as the flow comprises of various interactions among solid, gas and pipe wall. It is difficult to incorporate these interactions into a model. The pressure drop depends on flow, material and geometrical parameters. The existing models show high error when applied to other pipeline configurations of varying pipeline lengths or diameters. The current study investigates the capability of machine learning (ML) techniques to estimate the drop in pressure in FDP conveying of powders. Pneumatic conveying experimental data were used for training the network and then for predicting the pressure drop. For estimating the pressure drop, four distinct ML algorithms light gradient boosting machine (LighGBM)), multilayer perception (MLP), K-nearerst neighbors (KNN), extreme gradient boosting (XGBoost), and were selected. XGBoost model performed better than other models chosen for the study with ±5% error margin while training and testing the data, and ±10% error margin in validating the data. MLP, XGBoost, KNN, and LightGBM models predicted the data of pressure drop with MAE of 5.05, 1.19, 5.72, and 2.85, respectively, for training as well as testing data. Among the four models considered, the model using XGBoost algorithm performed the best, whereas the model using KNN algorithm performed poorly in predicting the FDP conveying pressure drop.

Article History

Received March 10, 2023
Revised May 5, 2023
Accepted May 19, 2023
Available online July 29, 2023

Keywords:

KNN
LightGBM
Machine learning
MLP
Pressure drop
XGBoost

1. INTRODUCTION

Fluidized dense phase (FDP) conveying of powders involves two layers with dilute and dense flow in the upper and bottom portions of pipeline respectively. At the bottom of the pipeline, the dense phase layer moves like a wave (or dune) (Sanchez et al., 2003). For powders that retain air and in situations where the solids concentration or solids loading is very large, this form of flow is appropriate. Industries use this conveying method because it uses less power, lowers particle attrition, and prevents pipe erosion, especially at bends (Alkassar et al,. 2021a). On the other hand the flow is pulsatile and transient in nature, causing complex interactions of particles, gas, and walls (Shijo & Behera, 2017; Alkassar et al., 2020). This complex disposition of flow makes it challenging to mathematically model the FDP conveying. In an attempt to partially incorporate these interactions in mathematical models that predict pressure drop of conveying through pipelines, researchers have used discrete element method (DEM) and computational fluid dynamics (CFD) (Behera et al., 2013a; Alkassar et al., 2021b). However, for long pipelines, simulation of CFD or DEM models consume more time. Few researchers have also developed models of solid friction factor for pressure drop prediction which was limited to smaller pipeline length and diameters (Datta & Ratnayake, 2003; Behera et al., 2013b). Hence modelling of the pressure drop in FDP conveying is still a difficult work for researchers. In the past few years machine learning (ML) technique has been applied extensively in different fields.

Several authors have applied different optimization techniques to optimize the system parameters. In order to avoid frequent pipeline clogs, pressure loss modeling and bulk transfer system simulation are carried out by Kim and Lee (2020). A genetic algorithm is used to determine ideal spacing between air boosters and the quantity of air boosters required to reduce pressure loss in the pipeline. Liu et al. (2023) optimized the parameters involved in photovoltaic geothermal coupling system. These variables include the configuration of pipes, the nanoparticle type, and the particle concentration in the nanofluids. The multi-objective optimization lowers the life cycle cost (LCC) as well as the overall levelized cost

NOMENCLATURE

AAD%	average absolute deviation%	*MAE*	mean absolute error
B	bend loss coefficient	m_a	air mass flow rate
D	pipe diameter	m_s	solid mass flow rate
d	particle diameter	$m*$	solid loading ratio
FDP	fluidized dense phase	N_b	number of bends
g	acceleration due to gravity	RMSE	root-mean square error
ΔP	total pressure drop	rhobl	loose-poured bulk density
ΔP_b	bend pressure drop	ρ_{ai}	inlet air density
ΔP_v	vertical pressure drop	*s*	activation function
L	pipeline length	slr	solid loading ratio
L_v	length of vertical pipe	u_{ai}	inlet air velocity
$L_s(h)$	training error	w_{ji}	weight

of energy (LCOE) by 32.3% compared to the unoptimized photovoltaic geothermal coupling system. Zawawi et al. (2022) applied the response surface methodology (RSM) quadratic models to optimize operating parameters in automobile air-conditioning (AAC) system that were useful in finding the relation between input parameters and responses. With less compressor work and electrical consumption, the optimization boosted the capacity of cooling and the coefficient of performance (COP). The following paragraph discusses a few of these applications of machine learning.

Using huge data from simulations of gas-particle fluidization, two ML algorithms extreme gradient boosting (XGBoost)) and (artificial neural networks (ANN) were developed to predict drag adjustments (Zhu et al., 2020). Three distinct categorization algorithms: K-nearest neighbors (KNN), multilayer perception (MLP), and random forest (RF) were applied for classifying flow patterns in pulsating heat pipes (Loyola-Fuentes et al., 2022). MLP gave better prediction than other algorithms. A model was developed using a variety of ML techniques, capable of predicting the particle surface charge density in mono-dispersed gas-solid fluidized beds with greater accuracy (Lu et al., 2022). The most effective were MLP and Support vector regression (SVR) models. Machine learning was used to estimate syngas compositions and lower heating values (LHV) utilizing a variety of lignocellulosic biomass feedstocks under varied operating conditions (Kim et al., 2023). The three ML approaches used were RF, support vector machines (SVM), and ANN. The ML approach was applied to choose an optimum mesh size to improve the accuracy of the fluid-particle CFD-DEM model developed for large scale system (Davydzenka & Tahmasebi, 2022). In the recent past, many such applications of machine learning are observed.

The majority of models are only effective at estimating the pressure drop for specific pipeline arrangements (defined by definite pipeline length, diameter, number of bends, etc.) or for specific conveying material type. A few researchers have employed their models for predicting the pressure drop in pipeline length and diameter scale-up conditions (i.e., a longer pipeline or one with a larger diameter than the pipeline's dimensions in the test settings of the conducted experiment), and the results have consistently shown a significant error. The researchers did not use experimental data for very lengthy pipelines (more than 200 m in length) while creating their models. The ML approach has been applied in various fields, but few researchers have applied such technique in pneumatic conveying of powders. Few researches of machine learning in pneumatic conveying have been discussed in the following section. In this study, the straight-line pressure drop in FDP conveying is predicted and compared by using different ML algorithms. The present model would be helpful for prediction of pressure drop in long pipelines. It would also aid in rebuilding the model by addition of more experimental data of different conveying materials pertaining to a generalized model.

1.1 Machine Learning in Pneumatic Conveying

A model was developed for prediction of the pipeline pressure drop using deep neural network (DNN) (Zhang & Lei, 2019). The benefit of this prediction model is that it immediately realizes the prediction of end-to-end pipeline pressure loss without extracting the features of flow parameter data in advance. Solid flow velocity was incorporated in a ML model to predict the meter output voltage, and this yields superior accuracy for a broad range of operating conditions in pneumatic conveying experiments (Kidd et al., 2020). Implementing this model gives a latest calibration function, which, by applying signal processing software produces a precise measurement of mass flow. Flow regime parameters were included as an input for developing a ML model for prediction of solid mass flow rate in pneumatic conveying by applying acoustic emission detection (Zhang et al., 2021). This method reduced the error of prediction in different flow regimes. The signal properties and the particle mass flow rate were correlated using ML models, such as SVM, ANN, and Convolutional Neural Networks (CNN), using training data with selected features (Abbas et al., 2022). Modeling of saturated flow pressure loss and boiling heat transfer in evaporating flow was proposed using new correlations and deep learning techniques. The implemented ideal deep learning (DL) model increased the prediction accuracy for both pressure drop and heat transfer (Chen et al., 2023). SVM model performs better than ANN and CNN. The ANN technique was used for the pressure drop prediction in FDP conveying (Shijo & Behera, 2021). Three different algorithms (Levenberg Marquardt, Scaled Conjugate Gradient and Bayesian

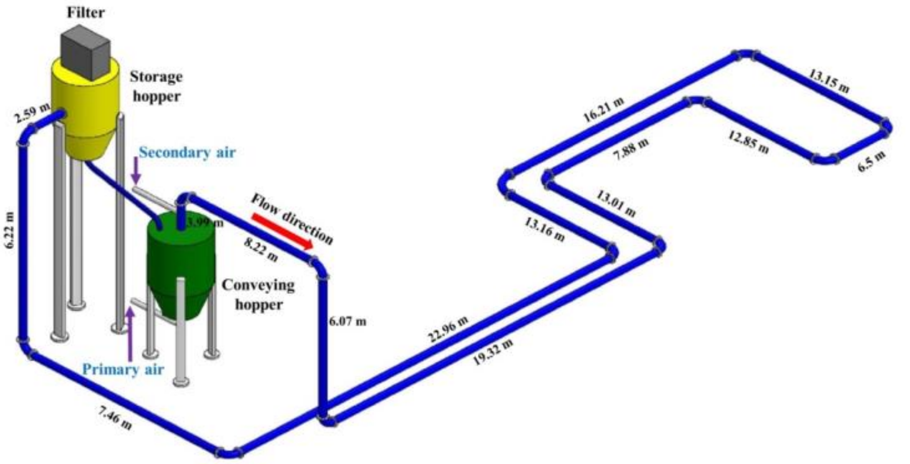

Fig. 1 Schematic diagram of pneumatic conveying system (Behera et al., 2015)

Table 1 Physical properties of conveying materials

Sl. No.	Material	Mean particle diameter (µm)	Particle density (kg/m^3)	Loose-poured bulk density (kg/m^3)	Pipe diameter (mm)	Pipe length (m)
1	EPS Dust	7	3637	610	69	168
2	EPS Dust	7	3637	610	105	168
3	EPS Dust	7	3637	610	69	554
4	Fly ash	30	2300	700	69	168
5	Fly ash	30	2300	700	69	554
6	Cement meal	19	2910	1080	65	254
7	Fly ash	22	2370	660	65	254
8	Fly ash	14.9	2096	724	53	173

Regularization) were applied. Levenberg Marquardt yielded better results in comparison to other two algorithms.

1.2 Objectives of the Study

The present study develops a model involving various input parameters (such as pipeline diameter, pipeline length, air or solid mass flow rate, particle density and particle diameter), influencing the total pressure drop in FDP conveying. Artificial intelligence technology has been applied successfully in various fields of engineering to predict important parameters. Many ML algorithms are available for tackling regression problems. The present study aims to apply different ML algorithms in artificial intelligence for the prediction of total pressure drop (ΔP) in FDP conveying.

2. Experiment

Tests of pneumatic conveying of fly ash were performed on a 173-m long 53-mm uniform-bore pipeline arrangement (Fig. 1). The configuration was having bends, horizontal and vertical sections. A group of sonic nozzles was utilized in the setup to control airflow rate into a blow tank (1 m^3 capacity and top discharge) used for pressurizing the material to be conveyed. Compressed air was delivered using primary air (supplied at the base of blow tank) and secondary air (supplied through the upper zone of blow tank). Supplemental air had to be supplied to prevent a pipeline blockage. For measuring primary and secondary air pressures, two transmitters (accuracy: 0.05%; range: 0–5 BarG) were used. Data of pressure were acquired at sample frequencies ranging between 60 Hz and 100 Hz. Load cells were used to measure the solid mass flow from blow tank to receivers.

The analysis in present study also included the test data obtained by the previous researchers (Mallick, 2009; Setia et al., 2016). Table 1 shows the physical properties of the materials conveyed and the pipeline arrangements employed by these researchers. They employed three different pipeline arrangements. The original pipeline arrangement was as follows: 168-m length, 69-mm inner bore pipe with five 90° bends that had a 1-m radius of curvature and 7-m vertical pipeline section. The second pipeline layout had 10 bends of 90^o with an overall length of 254 m and an ID of 65 mm. The third pipeline configuration was as follows: 554-m length, 69-mm ID pipe with 17 bends of 90^o, each having 1 m radius.

In the present analysis, pressure drop through straight horizontal pipeline was computed from the total pressure drop. The pressure drop across a bend section (ΔP_b) and a vertical section (ΔP_v) was determined applying Eqs. (1) and (2), respectively. The pressure drop through straight horizontal pipeline was computed by deducting the pressure loss across bends (*ΔP$_b$*) and the pressure loss across vertical portions (*ΔP$_V$*) from the total pressure drop found through experimentation.

$$\Delta P_b = (1+m^*) N_b B \frac{\rho_{ai} u_{ai}^2}{2} \tag{1}$$

$$\Delta P_v = m^* \rho_{ai} g L_v \frac{u_{ai}}{u_{si}} \tag{2}$$

3. Machine Learning Models

Supervised learning challenges are the issues with datasets containing both independent and dependent variables. Certain ML methods depend on similarity distance between data points that starts either explicitly or implicitly but fails to scale well in larger dimensions. As a result, a number of ML techniques perform poorly as the complexity of the problem rises, a phenomenon called the “curse of dimensionality."

A learning algorithm consists of a training data set S, obtained from a distribution D, and identified by a target function *f*. The main aim of the algorithm is to obtain a predictor that reduces error w.r.t. the distribution and target function. The learning algorithm is provided with a sample data, known as training dataset. Learning paradigm is also known as Empirical Risk Minimization (ERM) that is applied to identify a predictor which reduces the training error L_s *(h)* as given below (Shalev-Shwartz & Ben-David, 2014):

$$L_s(h) = \frac{\left|\left\{i \in \{1......,n\} : h(x_i) \neq y_i\right\}\right|}{n} \tag{3}$$

The above Eq. (3) is the foundation equation referred to in the learning theory (Vapnik, 1992). Training the data is to memorize the data, but it does not perform well for the unseen data (Zhang et al., 2018). The most common way to handle this issue is to use the ERM rule to a confined area while choosing predictor set called as the hypothesis class. In the following sections, a few of the learning theory-based prediction models are discussed briefly.

3.1 Multi-Layer Perception

The MLP is an artificial neural network technique with fully connected multi-layer neural networks. In a single-layer perception, it is not enough to deliver necessary performance for many complex applications. Hence, MLP with layers stacked on one another is developed in which a signal flows in one direction only, hence termed as feed forward neural (FNN) network. The dependent variable is first extracted as a linear set of input variables, and then it is mapped as a nonlinear function of the features that were generated from the input variables. They are the most basic kind of deep network and are composed of many hidden layer neurons. Any common function can be built by considering networks with more interconnected layers of components. The weighted linear combination of inputs is processed by each layer, which then transforms it employing an activation function. Weights (w_{ji}) are equivalent to regression coefficients. The weight is connected from unit *i* to *j* in the layer 1 and *s* is the activation function. The output from unit *j* can be calculated using Eqs. (4):

$$z_j = s\left(\sum_{i=0}^{d} w_{ji}^{I} x_i\right) \tag{4}$$

3.2 Extreme gradient boosting

In recent years ML techniques are made more effective by boosting that was originally built to be applied for classification problems but efficiently applied in regression problems (Freund & Schapire, 1999). Boosting algorithms drastically improve capacity of weak learners in approximating good predictors for more complicated problems. Boosted trees are versatile and adaptable solution for various problems. The “curse of dimensionality" problem is resolved by tree boosting by learning the association between data points through adaptive neighborhood adjustment rather than depending on the distance measure. In contrast to neural networks, this also renders the model invariant to data modification, therefore scaling the features is no longer necessary. Furthermore, if the trees are built deeper, then the interactions between the elements that can be captured are higher (Nielsen, 2016). A regularization term is included in the Gradient Boosted Decision Trees (GBDT) variation known as XGBoost to prevent overfitting. Furthermore, XGBoost uses the second-order Taylor series of loss function as opposed to the first-order derivative utilized in GBDT (Rashmi & Gilad-Bachrach, 2015).

3.3 K-Nearest Neighbors

Using a variety of distance metrics (such as Euclidean, Manhattan or Minskowski), the KNN-regression attempts to locate points in the dataset that are nearest to the “query" point. At a specified K value, the KNN regression searches for the training data closest to x_0 and categorized by N_0 and can be presented as follows:

$$\hat{f}(x_0) = \frac{1}{K} \sum_{x_i \in N_0} X_i \tag{5}$$

A small value of K allows for a high variance and low bias, while a larger value of K allows for a low variance and high bias. The ideal value of K is calculated by bias-variance trade-off. The dataset is arranged into a tree-type hierarchical data structure applying the KNN algorithm (Kumar et al., 2008).

3.4 Light Gradient Boosting Machine (Lightgbm)

The LightGBM is an algorithm proposed to speed up gradient boosting on decision trees (Ke et al., 2017). This is based on the idea that, while developing leaves, only a subset of possible splits is checked in reality. The Gradient-based One-Side Sampling (GOSS) and Exclusive Feature Bundling (EFB) are two original algorithms that enable this. With GOSS, a definite proportion of data instances with small gradients is neglected, and only the rest of the instances are used to calculate information gain. By grouping together the mutually exclusive features, EFB helps to minimize the number of features.

3.5 Selection of Model Parameters

The choice of parameters and their values in the MLP model is not simple since various factors, including solver type, activation function, learning rate, hidden layer size, maximum number of iterations, and batch size, must be taken into account. The ideal values of the parameters have been determined on the basis of correlation coefficient (R^2) and root mean square error (RMSE) value. The parameter with the greatest influence in this model is the size of the hidden layer.

A lot of parameters are observed in the XGBoost model. Several of these parameters have a more significant effect on the performance of model, and are termed as Hyperparameters. The performance of a model can be considerably enhanced by selecting the ideal values of hyperparameters. Learning rate, maximum depth and number of estimators are the hyperparameters in this model. The present work used the scikit-learn GridSeacrhCV method to perform a cross-validated grid search. In search of the best result, it thoroughly tests each and every potential set of hyperparameters.

In the KNN model of regression, it is necessary to determine "Points in the neighborhood" (k) to achieve the target values of parameters. In order to calculate the target value for the closest "k" points, KNN regression first calculates the "k" nearest points in the space. For the present work, chosen value of the "k" is 2 by assigning equal weight in the neighborhood. "Points in the neighbourhood" is the most important parameter of KNN model. The distance metric 'Minkowski' is chosen for determining separation between two points.

Table 2 Model parameters

Model	Parameter	Value
MLP	Activation function	Relu
	Solver	Lbfgs
	Regularization Parameter, α	0.00001
	Hidden layer size	20
	Maximum number of iterations	15000
	Batch size	Auto
	Learning rate, λ	0.0025
XGBoost	Learning rate	0.015
	Number of estimators	700
	Maximum delta step	0
	Maximum depth	4
	Minimum child weight	0.001
	Scale pos weight	1
	Minimum child samples	20
KNN	Algorithm	Auto
	Leaf size	30
	Metric	Minkowski
	Power parameter	2
	Number of nearest neighbours	2
	Weights	Uniform
LightGBM	Learning rate	0.015
	Number of estimators	700
	Maximum depth	5
	Evaluation Metric	None

In LightGBM model, the most important parameters chosen are the number of estimators, learning rate, number of leaves, maximum depth, etc. In this model, the scikit-learn GridSeacrhCV method was used to carry out grid search to find out the best set of hyperparameters.

The present models used different values or settings of parameters as presented in Table *2*.

3.6 Model Evaluation

In this work, codes for the ML models were developed by using various Python and libraries such as Numpy, Pandas and Scikit-learn. For training the network, 260 numbers of test data were applied. It involved the data of FDP conveying of materials such as EPS dust, fly ash and cement meal. The training and test data split ratio was 80%:20%. The model was validated by using 72 numbers of data. Four statistical parameters Mean absolute error (MAE), RMSE, R^2 and average absolute deviation percentage (AAD%) were calculated for evaluating the performance of each model. solid mass flow rate (m_s), air mass flow rate (m_a), solid loading ratio (slr), pipeline length (L), particle diameter (d), pipeline diameter (D) and loose-poured bulk density (*rhobl*) were chosen as input parameters whereas pressure drop was the output parameter.

4. RESULTS AND DISCUSSIONS

4.1 Feature Importance

The feature selection results of the suggested list of input variables using the mutual information (MI) are displayed in Fig. 2. MI is used to assess similarity between two datasets. The XGBoost technique is used to determine the feature significance. In this method, the value of individual variable has no importance. The relative value between the variables is more significant. According Fig. 2, m_a and m_s are the most significant of all the features mentioned above. However, it appears that m_a with the highest F score has the most significant influence on the pressure drop. On the other hand, compared to other features the loose-poured bulk density is less significant in influencing the pressure drop.

4.2 Performance analysis of ML models

Iterations, learning rate, and error functions affect the wellness performance of a training algorithm. The difference between experimental value and predicted value determines the error function of a network. In order to reduce error functions, weights are updated in ANN. Various statistical measures, such as MAE, RMSE, R^2, and AAD%, can be used to examine the functioning of the network. The value of R^2 should be close to one (1) and the RMSE value should be as close as possible to zero (0) for the best outcomes. The degree of the

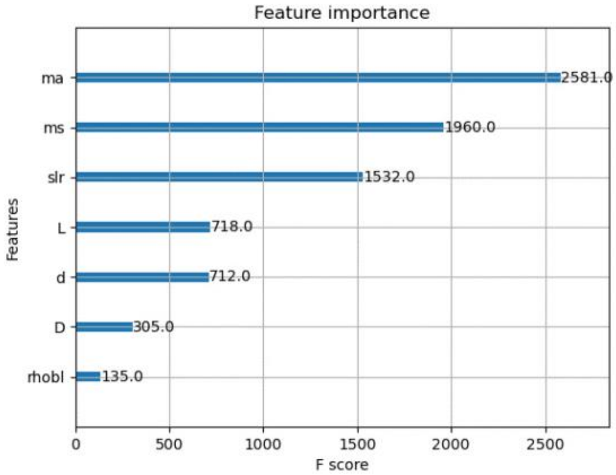

Fig. 2 Feature importance identified by MI technique

correlation between expected and experimental data is judged by the value of R^2 (Chang et al., 2012).

In this study several statistical measures, such as MAE, RMSE, R^2, and AAD%, were used to determine the accuracy of models. These statistical measures are defined as follows:

$$MAE = \frac{1}{N}\sum_{i=1}^{N}\left|X_{\exp} - X_{pred}\right| \tag{6}$$

$$RMSE = \sqrt{\frac{1}{N}\sum_{i=1}^{N}\left(X_{\exp} - X_{pred}\right)^2} \tag{7}$$

$$R^2 = \frac{\sum_{i=1}^{N}\left(X_{\exp} - \overline{X}_{pred}\right)^2 - \sum_{i=1}^{N}\left(X_{\exp} - \overline{X}_{pred}\right)^2}{\sum_{i=1}^{N}\left(X_{\exp} - \overline{X}_{pred}\right)^2} \tag{8}$$

$$AAD\% = \frac{1}{N}\sum_{i=1}^{N}\left(\left|\frac{X_{\exp} - X_{pred}}{X_{\exp}}\right|\right)\times 100 \tag{9}$$

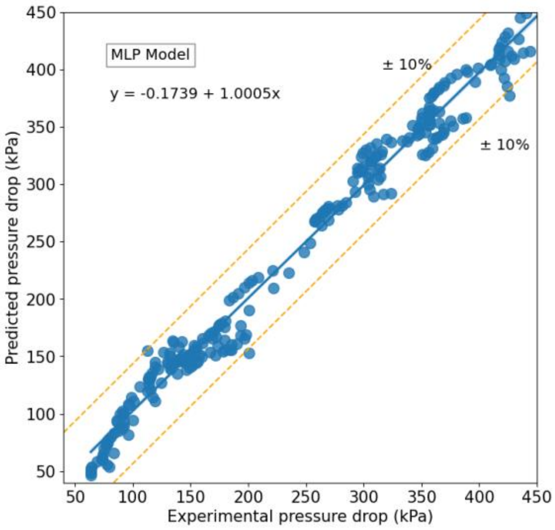

Fig. 3 Predicted pressure drop using MLP model versus the experimental pressure drop

Table 3 Neuron independence test

No. of neurons	MAE	RMSE	R^2	AAD%
5	18.03	23.4	0.9605	7.45
6	59.08	73.04	0.6153	24.37
7	21.67	27.07	0.9471	8.95
8	44.18	54.31	0.7872	18.24
9	15.94	20.4	0.9699	6.58
10	19.8	25.09	0.9545	8.18
11	21.46	26.56	0.9491	8.86
12	21.48	26.53	0.9492	8.87
13	21.46	26.54	0.9491	8.87
14	17.32	22.3	0.9641	7.16
15	21.5	26.51	0.9492	8.88
16	17.14	22.25	0.9643	7.08
17	21.48	26.53	0.9492	8.87
18	17.44	23.17	0.9612	7.21
19	21.46	26.67	0.9486	8.87
20	15.34	19.79	0.9717	6.34
21	21.46	26.49	0.9493	8.86
22	12.28	15.33	0.983	5.07
23	21.53	26.87	0.9479	8.89
24	18.04	23.7	0.9594	7.45
25	19.91	25.18	0.9545	8.23
26	17.8	23.47	0.9602	7.35
27	17.66	23.04	0.9617	7.29
28	17.43	23.19	0.9611	7.2
29	18.27	23.85	0.9585	7.54
30	16.17	22.28	0.9641	6.69

4.3 Neuron Independence Test

By varying the number of hidden neurons in the MLP model from 1 to 30, a neuron independent test was performed. One may choose a network that makes accurate predictions. It is possible to select the network of exact prediction based on the lowest values of MSE or MRE and the maximal values of R^2. Compared to other networks, the network with a single hidden layer made up of 22 neurons performs better as presented in Table 3.

4.4 Comparison of Predicted and Experimental Pressure Drop Using Four Models

Figure 3 illustrates the plot between the experimental and predicted data of pressure drop by the MLP model. A 10% variation in errors was observed. The MAE, RMSE, R^2, and average absolute deviation of the MLP model were 12.22, 15.54, 0.9825, and 5.05%, respectively.

Figure 4 shows a comparison of experimental and predicted pressure drop using the XGBoost model. The margin of error between predicted and experimental data was less than 5%, which is extremely low. The XGBoost model gives an MAE of 2.88, RMSE of 4.13, R^2 value of 0.9987 and AAD of 1.19%. Among the three parameters,

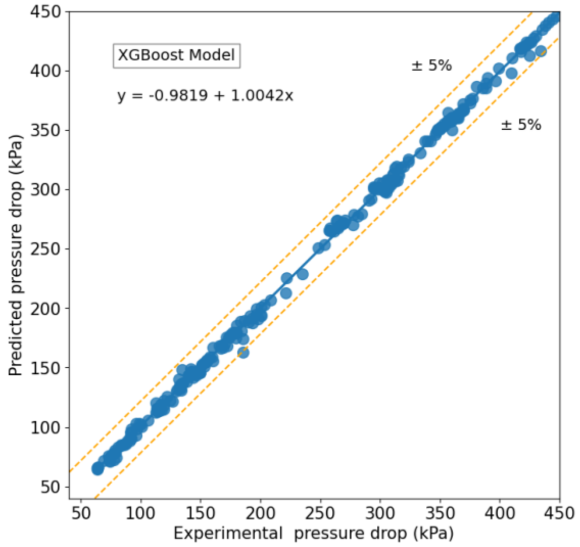

Fig. 4 Predicted pressure drop using XGBoost model versus the experimental pressure drop

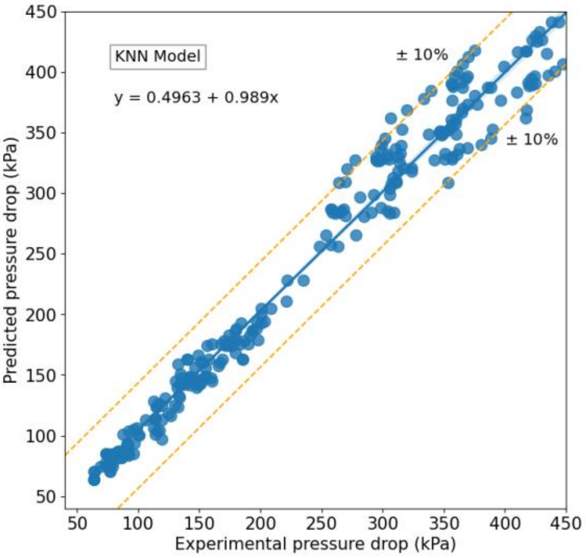

Fig. 5 Predicted pressure drop using KNN model versus the experimental pressure drop

the XGBoost model gives the lowest value for MAE and RMSE, and highest value for R^2.

Plot of the experimental versus the predicted pressure drop applying the KNN model is shown in Fig. 5. In comparison to other models, the deviation between predicted and experimental results was within a 10% margin of error. Table 4 displays the MAE, RMSE, R^2 value, and AAD% values. An MAE of 13.99, RMSE of 19.52, R^2 of 0.9725, and AAD of 5.72% were obtained by using the KNN model. Among the four models, the KNN model yields the highest MAE, RMSE and AAD% values and the lowest R^2 value.

The comparison of experimental and predicted pressure drop results applying the LightGBM model is shown in Fig. 6. The differences in predicted and experimental pressure drop data are limited to 10%. It gave MSE, RMSE, R^2 and AAD values of 6.91, 9.34, 0.9936 and 2.85%, respectively. Figure 7 depicts a comparison between predicted pressure drop data with experimental data using four ML models. It displays that the predicted pressure drop using MLP, XGBoost and LightGBM models and the experimental results are very close to each other. However, the pressure drop prediction using the KNN model significantly differs from the results of the experiment.

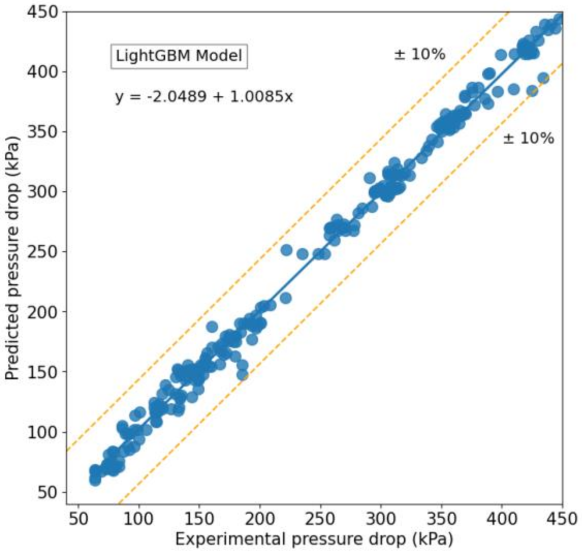

Fig. 6 Predicted pressure drop using LightGBM model versus the experimental pressure drop

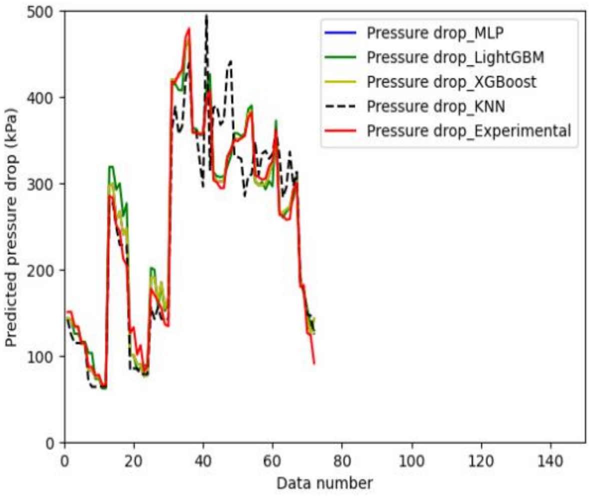

Fig. 7 Comparison between the predicted pressured drop and experimental data using different models

The XGBoost model often is superior to several other ML algorithms as it employs Newton's technique, which applies a higher-order option for the optimization. Further randomization options found in XGBoost are applied to de-correlate each tree and lower the total variance of the model. This caused the improvement of XGBoost, a very effective ML model that considers the trade-off of bias-variance in the learning theory (Nielsen, 2016).

Table 4 Results of ML models

Sl. no.	Model name	MAE	RMSE	R^2	AAD%
1	MLP	12.28	15.33	0.9830	5.07
2	XGBoost	2.88	4.13	0.9987	1.19
3	KNN	13.99	19.52	0.9725	5.72
4	LightGBM	6.91	9.34	0.9936	2.85

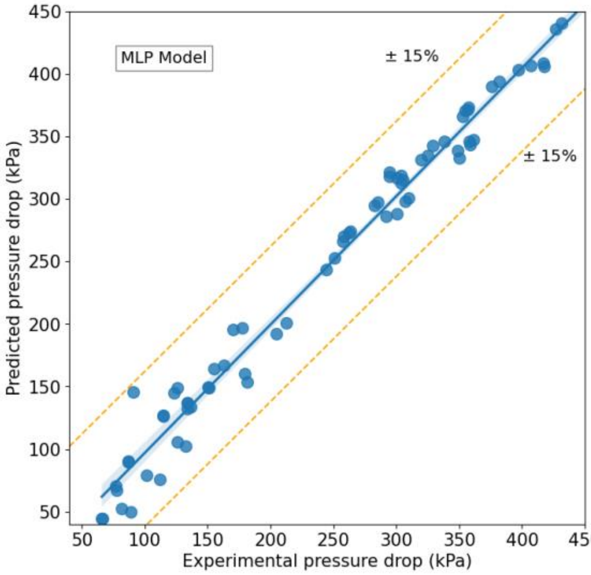

Fig. 8 Validation of the pressure drop data using MLP model

Considering performance, the LightGBM model is little behind the XGBoost model. However, the MAE and RMSE values in LightGBM appear to be higher than in XGBoost. Both grid and random searches were used to find the optimum hyperparameters for the current model, it may be because enough of the best hyperparameter tuning is not available.

The MLP model performs better but marginally lesser than the XGBoost and LightGBM models. MLP is capable of learning intricate connections between input and output parameters as well as the activation function of non-linear type that is found in the hidden layers of the network. They retain their robustness even in the context of noisy data and when the statistical distribution of the input parameters with regard to the response function changes (Memon et al., 2019).

The KNN model does not perform as effectively as done by the other three models. KNN is a distance-based network algorithm. The performance of the algorithm suffers due to the expensive disposition of finding the distance between two points. KNN is also susceptible to outliers and missing values. Therefore, it is necessary to replace missing values and eliminate outliers prior to using the KNN algorithm.

5. Validation of Models

All the four ML models were validated with the experimental data. Figures 8, 9, 10 and 11 show the respective predicted versus the experimental pressure

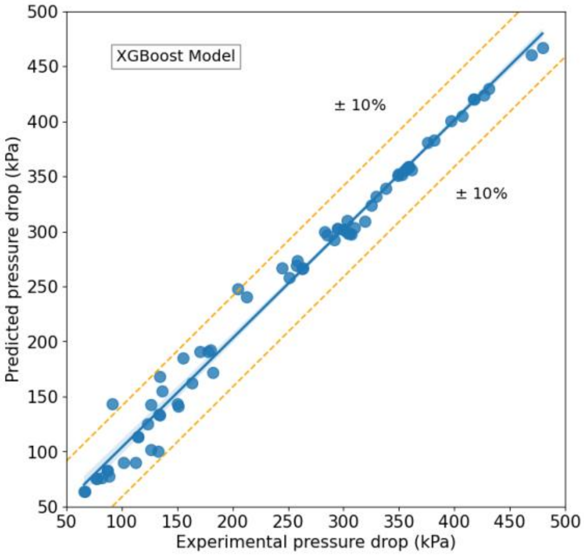

Fig. 9 Validation of the pressure drop data using XGBoost model

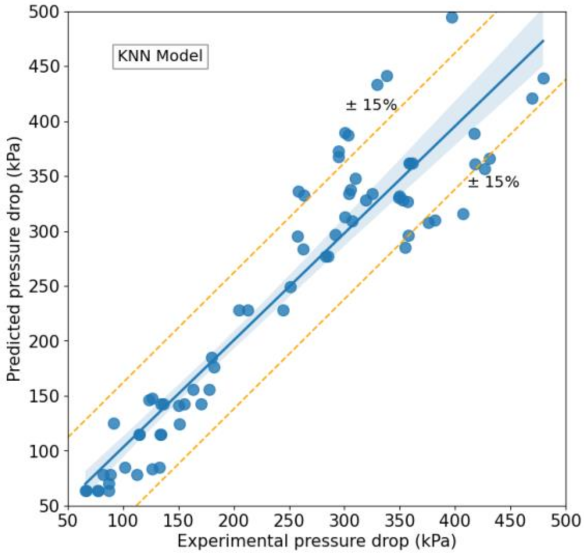

Fig. 10 Validation of the pressure drop data using KNN model

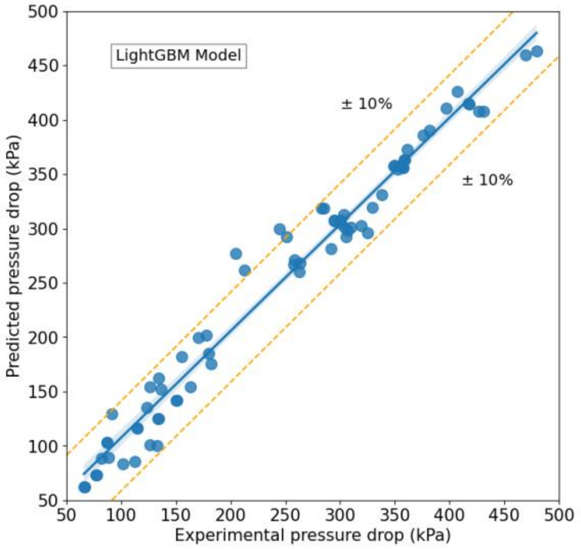

Fig. 11 Validation of the pressure drop data using LightGBM model

Table 5 Validation of ML models

Sl. no.	Model name	MAE	RMSE	R^2	Average absolute deviation %
1	MLP	13.52	16.58	0.9795	5.46
2	XGBoost	8.87	13.74	0.9859	3.56
3	KNN	31.69	42.97	0.8623	12.87
4	LightGBM	14.68	20.05	0.9700	5.83

drop comparative plots using the MLP, XGBoost, KNN and LightGBM models respectively. Predicted errors using The XGBoost and LighGBM models were within ±10% error margin, whereas using other two models it was within ±15%. The accuracy level in validating the models can be judged from the calculated values of statistical parameters given in Table 5. The XGBoost model yields the lowest values of MAE, RMSE and AAD and the highest R^2 value.

6. Conclusions

In this study ML models were developed for predicting the straight line pressure drop in FDP conveying of powders. Four different algorithms such as MLP, XGBoost, KNN and LightGBM were applied for modelling. Experimental data of cement meal, fly ash, and EPS Dust were used to train, test and validate the model. Key findings of this study are outlined below:

(1) ‘Air mass flow rate’ has been found to be the most important feature influencing the output parameter of “pressure drop”.

(2) Both XGBoost and LightGBM models both surpassed MLP and KNN models in accurately predicting the pressure drop (lower MAE, RMSE and AAD% values, and higher R^2 value).

(3) The KNN model showed the highest values of MAE, RMSE and AAD% in training/testing and validating the model.

(4) The MLP, XGBoost, KNN and LightGBM models predicted the pressure drop data with an AAD% of 5.07%, 1.19%, 5.72% and 2.85% respectively, for training and testing data. The XGBoost model showed better results in validating the data.

(5) The MLP model having a single hidden layer showed the optimum number of neurons, equal to 22.

Conflict of Interest

It is declared that there is no financial or non-financial interest in this part. The authors have no conflicts to disclose.

Authors Contribution

J. S. Shijo: Writing, Review and Editing; **N. Behera**: Formal Analysis

References

Abbas, F., Yan, Y., & Wang, L. (2020). *Mass flow measurement of pneumatically conveyed solids through multi-modal sensing and machine learning.* 2020 IEEE International Instrumentation and Measurement Technology Conference (I2MTC), IEEE, New York, NY, 1–6. https://ieeexplore.ieee.org/document/9128576

Alkassar, Y., Agarwal, V. K., Pandey, R. K., & Behera, N. (2020). Experimental study and Shannon entropy analysis of pressure fluctuations and flow mode transition in fluidized dense phase pneumatic conveying of fly ash. *Particuology*, 49, 169-178. https://doi.org/10.1016/j.partic.2019.03.003

Alkassar, Y., Agarwal, V. K., Pandey, R. K., & Behera, N. (2021a). Influence of particle attrition on erosive wear of bends in dilute phase pneumatic conveying. *Wear*, 476, 203594. https://doi.org/10.1016/j.wear.2020.203594

Alkassar, Y., Agarwal, V. K., Pandey, R., & Behera, N. (2021b). Analysis of dense phase pneumatic conveying of fly ash using CFD including particle size distribution. *Particulate Science Technology*, 39(3), 322–337. https://doi.org/10.1080/02726351.2020.1727592

Behera, N., Agarwal, V. K., Jones, M. G., & Williams, K. C. (2013a). CFD modeling and analysis of dense phase pneumatic conveying of fine particles including particle size distribution. *Powder Technology*, 244, 30-37. https://doi.org/10.1016/j.powtec.2013.04.005

Behera, N., Agarwal, V. K., Jones, M., & Williams, K. C. (2015). Power spectral density analysis of pressure fluctuation in pneumatic conveying of powders. *Powder Technology*, 33 (5), 510-516. https://doi.org/10.1080/02726351.2015.1008079

Behera, N., Agarwal, V., Jones, M., & Williams, K. (2013b). Modeling and analysis of solids friction factor for fluidized dense phase pneumatic conveying of powders. *Particulate Science Technology*, 31 (2), 136–146. https://doi.org/10.1080/02726351.2012.672544

Chang, Y., Lin, J., Shieh, J., & Abbod, M. (2012). Optimization the initial weights of artificial neural networks via genetic algorithm applied to hip bone fracture prediction. *Advances in Fuzzy Systems*, 2012, Article ID 951247. https://doi.org/10.1155/2012/951247

Chen, B. L., Yang, T. F., Sajjad, U., Ali, H. M., & Yan, W. M. (2023). Deep learning-based assessment of saturated flow boiling heat transfer and two-phase pressure drop for evaporating flow. *Engineering Analysis with Boundary Elements*, 151, 519-537.

https://doi.org/10.1016/j.enganabound.2023.03.016

Datta, V., & Ratnayake, C. (2003). A simple technique for scaling up pneumatic conveying systems. *Particulate Science Technology*, 21 (3), 227–236. https://doi.org/10.1080/02726350307480

Davydzenka, T., & Tahmasebi, P. (2022). High-resolution fluid–particle interactions: a machine learning approach. *Journal of Fluid Mechanics*, 938, A20. https://doi.org/10.1017/jfm.2022.174

Freund, Y., & Schapire, R. (1999). A short introduction to boosting. *Journal of the Japanese Society for Artificial Intelligence*, 14(5), 771-780. https://cseweb.ucsd.edu/~yfreund/papers/IntroToBoosting.pdf

Ke, G., Meng, Q., Finley, T., Wang, T., Chen, W., Ma, W., Ye, Q., & Liu, T. Y. (2017). *Lightgbm: A highly efficient gradient boosting decision tree*. Proceedings of the advances in neural information processing systems, NeurIPS, 3146–3154. https://proceedings.neurips.cc/paper_files/paper/2017/file/6449f44a102fde848669bdd9eb6b76fa-Paper.pdf

Kidd, A. J., Zhang, J., & Cheng, R. (2020). A low-error calibration function for an electrostatic gas-solid flow meter obtained via machine learning techniques with experimental data. *Energy and Built Environment*, 1(2), 224-232. https://doi.org/10.1016/j.enbenv.2020.02.003

Kim, J. Y., Kim, D., Li, Z. J., Dariva, C., Cao, Y., & Ellis, N. (2023). Predicting and optimizing syngas production from fluidized bed biomass gasifiers: A machine learning approach. *Energy*, 263:125900. https://doi.org/10.1016/j.energy.2022.125900

Kim, Y., & Lee, K. (2020). Pressure loss optimization to reduce pipeline clogging in bulk transfer system of offshore drilling rig. *Applied Sciences*, 10(21), 7515. https://doi.org/10.3390/app10217515

Kumar, N., Zhang, L., & Nayar, S. (2008). *What is a good nearest neighbors algorithm for finding similar patches in images?* Proceedings of the European Conference on Computer Vision, Springer, Switzerland AG, 364–378. https://link.springer.com/chapter/10.1007/978-3-540-88688-4_27

Liu, Z., Yang, X., Ali, H. M., Liu, R., & Yan, J., (2023). Multi-objective optimizations and multi-criteria assessments for a nanofluid-aided geothermal PV hybrid system. *Energy Reports*, 9, 96-113. https://doi.org/10.1016/j.egyr.2022.11.170

Loyola-Fuentes, J., Pietrasanta, L., Marengo, M., & Coletti, F. (2022). Machine learning algorithms for flow pattern classification in pulsating heat pipes. *Energies*, 15(6), 1970. https://doi.org/10.3390/en15061970

Lu, J., Duan, C., & Zhao, Y. (2022). Machine learning approach to predict the surface charge density of monodispersed particles in gas–solid fluidized beds. *ACS Omega*, *7*(11), 9879-9890. https://doi.org/10.1021/acsomega.2c00299

Mallick, S. S. (2009). *Modeling of fluidized dense phase pneumatic conveying of powders*. [Doctoral Thesis, University of Wollongong]. Centre for bulk solid and particulate technologies, Wollongong NSW, Australia.

Memon, N., Patel, S. B., & Patel, D. P. (2019). *Comparative analysis of artificial neural network and XGBoost algorithm for PolSAR image classification*. International Conference on Pattern Recognition and Machine Intelligence, Springer, Cham, Switzerland, 452-460. https://link.springer.com/chapter/10.1007/978-3-030-34869-4_49

Nielsen, D. (2016). *Tree boosting with XGBoost-Why does XGBoost win '"every"' machine learning competition?* [Master Thesis]. NTNU, Trondheim, Norway. http://hdl.handle.net/11250/2433761

Rashmi, K. V., & Gilad-Bachrach, R. (2015). *Dart: dropouts meet multiple additive regression trees.* Proceedings of the Artificial Intelligence and Statistics, PMLR, Microtome Publishing, Brookline, MA, 489–497. https://doi.org/10.48550/arXiv.1505.01866

Sanchez, L., Vasquez, N., Klinzing, G., & Dhodapkar, S. (2003). Characterization of bulk solids to assess dense phase pneumatic conveying. *Powder Technology*, 138, 93–117. https://doi.org/10.1016/j.powtec.2003.08.061

Setia, G., Mallick, S. S., Pan, R., & Wypych, P. W. (2016). Modeling solids friction factor for fluidized dense-phase pneumatic transport of powders using two layer flow theory. *Powder Technology*, 294, 80–92. https://doi.org/10.1016/j.powtec.2016.02.006

Shalev-Shwartz, S., & Ben-David, S. (2014). *Understanding machine learning: from theory to algorithms*. Cambridge University Press, Cambridge, UK. https://doi.org/10.1017/CBO9781107298019

Shijo, J. S. & Behera, N. (2021). Performance prediction of pneumatic conveying of powders using artificial neural network method. *Powder Technology*, 388, 149–157. https://doi.org/10.1016/j.powtec.2021.04.071

Shijo, J. S., & Behera, N. (2017). Transient parameter analysis of pneumatic conveying of fine particles for predicting the change of mode of flow. *Particuology*, 32, 82–88. https://doi.org/10.1016/j.partic.2016.07.004

Vapnik, V. (1992). Principles of risk minimization for learning theory. *Advances in Neural Information Processing Systems*, 831–838.

Zawawi, N. N. M., Azmi, W. H., Redhwan, A. A. M., Ramadhan, A. I., & Ali, H. M. (2022). Optimization of air conditioning performance with Al2O3-SiO2/PAG composite nanolubricants using the response surface method. *Lubricants*, 10(10), 243. https://doi.org/10.3390/lubricants10100243

Zhang, H., Cisse, M., Dauphin, Y. N., & Lopez-Paz, D. (2018). Mixup: beyond empirical risk minimization. *ICLR*, 2018. https://doi.org/10.48550/arXiv.1710.09412

Zhang, P., Yang, Y., Huang, Z., Sun, J., Liao, Z., Wang, J., & Yang, Y. (2021). Machine learning assisted measurement of solid mass flow rate in horizontal pneumatic conveying by acoustic emission detection. *Chemical Engineering Science*, 229, 116083. https://doi.org/10.1016/j.ces.2020.116083

Zhang, X., & Lei, J. (2019). *Study on the optimum design of pneumatic conveying system based on DNN.* Proceedings of the 2019 International Conference on Robotics, Intelligent Control and Artificial Intelligence, 621-625. https://doi.org/10.1145/3366194.3366305

Zhu, L. T., Tang, J. X., & Luo, Z. H. (2020). Machine learning to assist filtered two-fluid model development for dense gas–particle flows. *AIChE Journal*, 66(6), 16973. https://doi.org/10.1002/aic.16973.