Artificial neural network models for wall parameters on plug-1 flow characteristics through pipelines

DOI : 10.36909/jer.ICMET.17203

A. K. Sethi[*], A. Rawat[**], V. Srivastava[**], A. K. Sharma[***]

[*]Galgotias College of Engineering & Technology, Greater Noida

[**]Motilal Nehru National Institute of Technology Allahabad

[***]Institute of Engineering & Technology Lucknow

*Corresponding Author: anilkrsethi@galgotiacollege.edu

ABSTRACT

Dense phase pneumatic conveying is a preferable method for transportation of a friable, fragile, abrasive, or agglomerated in nature material through pipeline with comparatively less wear and tear of the system and product as well. A fine particulate material moves as a single entity due to its cohesion in vertical or thin horizontal pipes in Plug-1 flow. A model based on mechanics was created and later modified using experiments for this type of flow. In this work Artificial Neural Networks (ANNs) models are used to study the effect of wall friction coefficient and coefficient of wall cohesion on the pressure drop. Three different datasets having 50000, 250000, and 500000 data points were used to test 19, 21, and 25 ANN architectures respectively. The best architecture was found to be t50-t40-r1 architecture with Adamax optimizer, with mean absolute percentage error (MAPE) being close to 0.00402% when tested on the 500000 samples dataset with 25000 test values, 0.0043% when tested on the dataset with 250000 samples and 25000 test values, and 0.0035% on the 50000 samples dataset with 10000 test values. The s20-s20-r1 architecture with Adam optimizer was quick and gave second best results with MAPE being close to 0.009% when tested on the 500000 samples dataset, 0.00988% when tested on the dataset with 250000 samples, and 0.00408% when tested on the 50000 samples dataset. The t40-t40-t40-t40-r1 with Adamax optimizer was slow but gave the third best results with MAPE being close to 0.0166% when tested on the 500000 samples dataset, 0.00496% when tested on the dataset with 250000 samples, and 0.00480% when tested on the 50000 samples dataset.

Keywords: Adamax optimizer; ANN; Magnesium Oxide; Plug-1 flow; Pressure drop

INTRODUCTION

The energy loss in plugs is mainly due to friction and developed a model for the associated friction force (Muschelknautz and Krambrock, 1969). A theoretical model is developed for horizontal plug flow in a pipe describing the mechanics of plug motion inside a pipe where a stationary layer of particles between plugs is present (Konrad, 1980). During plug movement, some of the particles in the stationary layer get picked up due to acceleration and that same number of particles is left behind. So, he balanced the forces on a single plug, using a similar model (Janssen, 1895) and also estimated the gravitational force by using (Wilson et al., 1972) technique. Mi and Wypach, 1995 developed a semi empirical model based on experimental investigations and moving slug force balance, to predict pipeline pressure drop of low velocity slug flow in horizontal pipe having no cohesion. Pan and Wypach, 1997 developed a model for pressure drop for the low velocity slug flow. This model worked well for particles with irregular shapes as well. They also split the radial stress into two parts but modified the previous Mi and Wypach model by directly applying radial stress to the force balance. Yi, 2001 conducted research to predict the pressure drop during conveying of granular materials in the form of a slug with a stationary layer and found that the weight of granular material in the slug on pressure drop should also be accounted for. They developed a modified equation using the momentum balance of the stationary layer's resistance to accelerating particles for the frontal force of the slug. Due to this resistance, the values of the pressure drop calculated were also higher and closer to experimental results. The pressure drop model used in l-in vertical pipe considering the air velocity, plug length, particle size, the cohesive properties andpowder moistureis in good agreementwith the2-inand 4-in vertical pipes (Hong and Klinzing, 1989; Borzone&Klinzing, 1987; Aziz &Klinzing, 1990).Rabinovich et al., 2012 found that plug friction force is a function of pipe diameter, wall friction coefficient, stress ratio and plug length and also found that a plug will move as one stiff entity if the stress ratio corresponds to the active case. Shaul and Kalman, 2015 defined Plug-1, Plug-2 and Plug-3 kinds of classical plugs in which the pressure drop was a function of a number of physical and geometrical properties of the plug material. Rawat and Kalman, 2019 constructed a flow regime chart with Archemedes Number and Reynolds Number as the x and y coordinates respectively. They also developed two new kinds of plugs, Plug-3*, an extension of Plug-3 where the bed is also moving, and Plug-2* where the layer between two plugs is moving. They also found that Plug-1 exists for fine particles of C and A types having Archemedes number (Ar) less than 100 (Geldart, 1973). Aziz and Klinzing, 1988 studied a flow using pneumatic conveying of coal at low velocity at an angle of 45 degrees. The flow was stable as the cohesion among the particles was more than the stresses exerted by air. Chen et al., 2002 used silica and kaolin powder to create a two-layer flow, where the inner plug-like flow was treated as a coulomb solid and the shear layer was treated as a frictional fluid. Mi and Wypach, 1995 discovered that it is easier to transport finer particles from dilute to dense phase as there is a smaller chance for pressure variations and vibrations to occur. They also presented semi empirical correlations to model pressure drop. The Hausner’s Ratio (HR) and cohesion are the important parameters that can establish the existence of plug-1 and plug-2. As with $Ar < 100$ and $HR > 1.25$, plug-1 exists; otherwise, plug-2 is expected (Rawat & Kalman, 2019; Hausner, 1967). The multilayer perceptron (MLP) neural network (Offor & Alabi, 2016) for predicting friction factor in turbulent flow of water with two hidden layers having 30 neurons each had relative error up to a maximum of 0.004% when compared with the Colebrook equation.In order to generate the training set for the ANN model, they solved the Colebrook equation iteratively.Also, Sablani et al., 2003 was able to predict the pressure drop for bingham plastic fluids using two input parameters: Bingham Reynolds Number and Hedstrom Number. He used the Regula Falsi Method to solve the relations developed by Govier and Aziz, 1972.The dataset consisted of 1177 training samples. 150000 iterations were carried out with 1 to 3 hidden layer configurations and 2-16 nodes. Total number of networks used was 24. Mean relative error (MRE), mean absolute error (MAE), and the standard deviations of the relative (STDR) and absolute (STDA) errors, and R2 score were used as the accuracy metrics. The mean relative error of the best architecture was close to 2.01% by a 2-8-8-1 configuration. An ANN structure (2-6-8-6-8-6-1) which is obtained using the NN-SVG tool is used for the current study. In these 2 neurons are selected in the input layer, 1 neuron in the output layer and from left to right 6-8-6-8-6 neurons in the hidden layers are selected to construct the 2-6-8-6-8-6-1 ANN structure.

DATASET GENERATION AND PRE-PROCESSING

Three datasets generated using Python code consisting of 50000, 250000 and 500000 samples were used. Datasets were generated by solving the modified pressure drop equation (Rawat & Kalman, 2019) and HR. Magnesium Oxide was taken as flow material and initial values were taken from the experiment conducted by Rawat and Kalman and the stress transmission coefficient was found using the relations given by Rabinovich. The coefficient of wall friction (μw) range from 0.1 to 0.9 and coefficient of wall cohesion (Cw) range from 700 to 2000 were taken as input parameters for this problem. The flowchart for the dataset generation is given in figure-1.

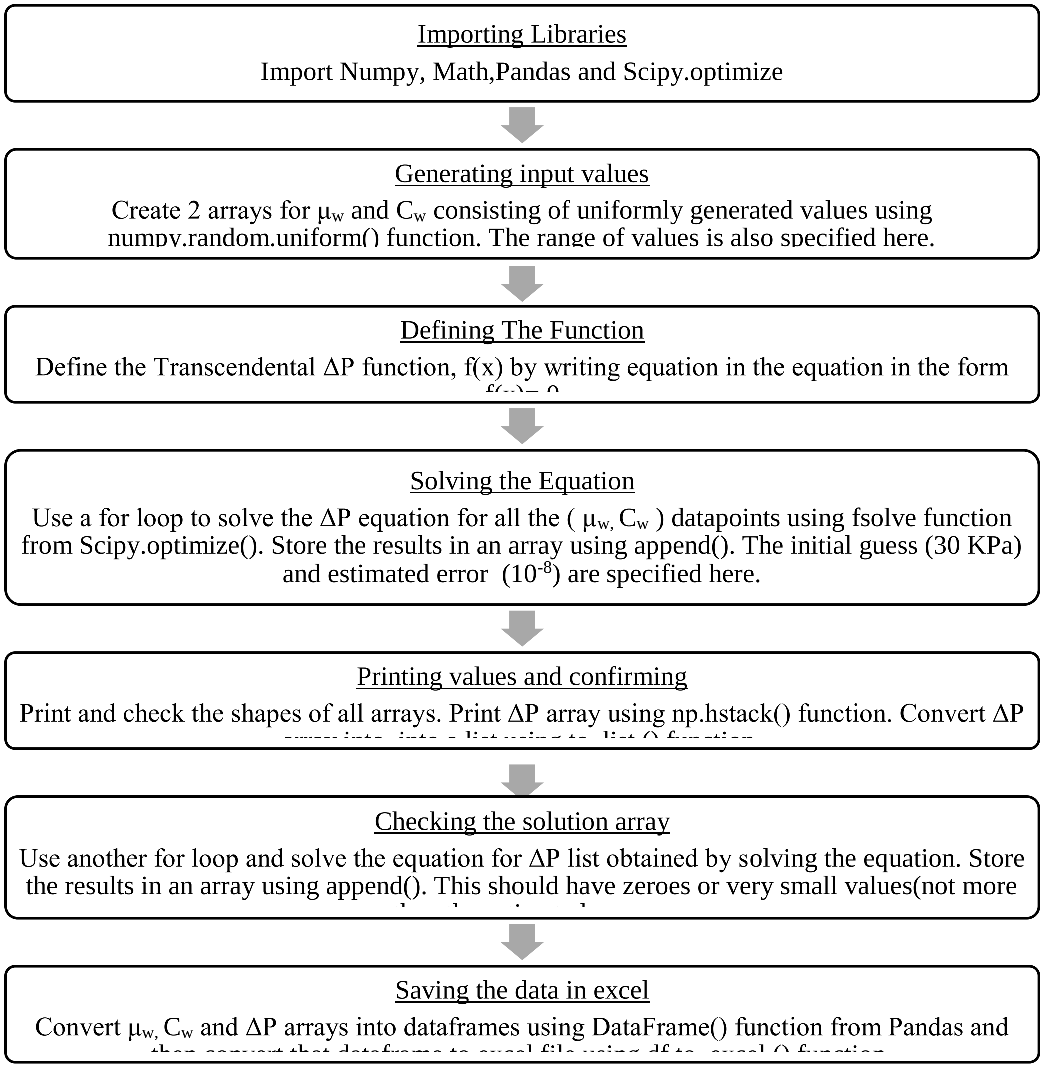

Figure 1. Flowchart for dataset generation

The test set split was 0.2 for a dataset of 50000 samples, 0.1 for 250000 samples and 0.05 for 500000 samples to cover the complete range of pressure values. The Python's Scipy Library is used to find the maxima and minima of the function. The 50000 dataset was trained on two random states and 250000 and 500000 datasets were trained on a single random state. Feature scaling was applied to inputs to standardize them at the beginning of the code.

DEVELOPMENT OF ANN CODE AND TRAINING

The ANN code used in this research was generated using Python 3 programming language. Sophisticated machine learning libraries and frameworks like TensorFlow (version 2.1.0), Keras (version2.3.1) and Scikit-learn (version 0.23.2) were used. This resulted in faster vectorized code without involving any 'for' loops. First layer consisted of μw and Cw values as input parameters to the ANN. Feature scaling method was applied so that the large value of one input doesn't overshadow the other input.

Different combinations of hyperparameters like number of layers, number of units in each layer and type of activation functions were tested on sample datasets. Number of hidden layers varied from 1 to 5. "tanh" and "sigmoid" and ''ReLU" activation functions were used for the hidden units while only the "ReLU" function was used in the output layer as it outputs only positive values and pressure drop can't be negative in our case. Different optimizers namely Adam, Adagrad, Adamax and Stochastic Gradient Descent are used and selected depending upon their suitability to different networks; The Mean Square Error was the cost function to be minimized.

Mini Batch sizes were 1024, 32768 and 131072 for 50000, 250000, 500000 samples respectively. The codes were run up to 10000 epochs. Mean absolute percentage error (MAPE), maximum error (ME) , explained variance score (EVS) and R2 score were chosen as the metrics to test the accuracy of the network. A common test set was assigned for a random state in an n- sample dataset so that the ANNs are tested on the same values of friction factor. These many different combinations of network architectures on three different datasets were tried as there is no single established procedure for choosing these parameters and it comes only by prior experience. Hence, the exercise was also important to get a complete idea of the problem. The Python code is written in a Jupyter notebook for the s20-s20-r1 network with 'adam' optimizer.

TESTING NETWORK AND TUNING HYPERPARAMETERS

Choice of optimization algorithm, number of hidden layers, number of neurons and activation function in each hidden layer were the hyperparameters considered in this problem. The approach used to tune hyperparameters was similar to coarse-to-fine sampling scheme. After getting a rough idea of the problem by trying a number of randomly chosen networks for an n-sample dataset (coarse sampling), the hyperparameters were tuned in the region where previously the results were more accurate.

RESULTS AND DISCUSSIONS

Initially 19 random ANN architectures were trained on the dataset with 50000 data points (50000 series) on random state 1. The maximum and minimum values in the test set of this shuffling state were 44756.4048731213 Pa and 13020.3262892055 Pa respectively (Uniformly Distributed data). However, the maxima and minima found using the python code was 12631.94700467 Pa and 44768.99578927 Pa respectively. Hence the minima ranges had some considerable difference. It was decided that any value below 12 KPa would be enough to ensure that the minimum ΔP range has been covered well. Hence after trying 13 random architectures, the random state was changed to 2. Now the maximum and minimum were 44741.28956 and 12.7430722390717 respectively, which are quite close to the extrema. The architectures took 4-13 minutes to train for 10000 iterations and a coarse –to-fine sampling scheme was used to arrive at the result. The Results of the ANN architectures tried on 50000 samples dataset is shown in table-1.

Table 1. The Results of the ANN architectures tried on 50000 samples dataset

S.No.	H1	H2	H3	H4	O	MAPE (%)	ME	EVS	R2 Score	Optimizer
1	s6				r1	0.109587650	0.594366653714359	0.999960605251383	0.9999606051831 38	adamax
2	t16				r1	0.029315792	0.168487223653771	0.999997527831089	0.9999975189547 33	sgd

3	r78				r1	0.016169026	0.066735847745122	0.999999318011189	0.999999316583709	adamax
4	r8	r60			r1	2.738562300	7.079132341053910	0.983247942946736	0.983215477669398	adagrad
5	s7	s7			r1	0.010210403	0.035461048685370	0.999999807457110	0.999999779466354	adam
6	t16	s32			r1	0.051914003	0.137078228319985	0.999995204333510	0.999992172777169	sgd
7	t16	t24	t2		r1	0.554563460	0.641213209937234	0.999741545880185	0.999278844195789	sgd
8	s2	s4	s2		r1	0.057990562	0.154369624941962	0.999993798306446	0.999993755893175	adamax
9	r25	r87	r27		r1	0.756012140	3.415859969807670	0.998068541025304	0.998068303379873	adagrad
10	s4	s4	s4	s4	r1	0.098860730	0.337974809315632	0.999982528639600	0.999982102809787	adamax
11	t3	t6	t3	t6	r1	0.043891046	0.159538530018757	0.999996963005556	0.999996081058202	adam
12	r78				r1	0.020882930	0.052209171097930	0.999998975499467	0.999998967519938	adamax
13	r100				r1	0.014794015	0.052668197903813	0.999999569449065	0.999999566533599	adamax
14	s7	s7			r1	0.023538800	0.087766657700946	0.999998942976857	0.999998893227324	adamax
15	s10	s10			r1	0.005668796	0.030819209902900	0.999999937565604	0.999999928565093	adam
16	s20	s20			r1	0 . 004087003	0.01 7768878748235	0.999999962335078	0.99999996022651	adam
17	r5	t6	s5		r1	0.026617197	0.095385076236795	0.999998654620514	0.999998610901462	adam
18	s7	t6	s7		r1	0.016977996	0.062900562097845	0.999999583086981	0.999999428379691	adamax
19	t10	r20	t10	r20	r1	0.010500674	0.029064861447198	0.999999771862608	0.999999756924024	adamax

where, in columns H heading represents the hidden layer number, O is for output layer. Optimizer used is given in the last column. ‘t’, ‘s’ and ‘r’ imply tanh, Sigmoid and ReLU activation functions respectively. The number of units in the corresponding layer is also given with the letter denoting function (for a layer denoted as t4 means that it has four neurons/ hidden units and the activation function for all those four units is tanh. Together many such layers will make a complete ANN, having notations like t10-r20-t10-r20-r1), whereas each row represents one ANN architecture.

In the dataset 2 with 250000 data points, instead of trying different random states, the test set was increased so that there are sufficient numbers of data points in any random state that the whole range of pressure drop is covered. Which means that the highest (44.7578432453543 kPa) and lowest (12.8115912238585 kPa) values of test set in random state 1 are close to the maxima and minima of pressure drop function found using the python code, which were 12631.94700467 Pa and 44768.99578927 Pa respectively. Here in the next table-2, 21 networks were tested which took 9-150 minutes to run 10000 iterations. The system time depends on a lot of factors impacting system usage and hence was not considered. Also, here we have to train the network only once, i.e., the input data is not dynamic. Once trained, there will be no need to change the input dataset.

Table 2. The Results of the ANN architectures tried on 250000 samples dataset

S.No.	H1	H2	H3	H4	H5	Q	MAPE (%)	ME	EVS	R2 Score	Optimizer
1	s28					r1	0.255051100	1.916739787400200	0.999774839282203	0.9997748009 0897	sgd
2	r135					r1	0.012030075	0.073621981973961	0.999999590661847	0.9999995902 87246	adamax
3	t76					r1	0.027280215	0.274836863816219	0.999996803232912	0.9999968005 58874	adamax
4	s15	s15				r1	0.012559166	0.105567302048640	0.999999634455471	0.9999996344 31033	adamax
5	r32	r44				r1	0.174442200	0.549311961472469	0.999983537182826	0.9999474048 60122	sgd

6	t24	t28				r1	26.778860000	22.801111449089600	0.123798171476729	-1.090427789280870	adagrad
7	s10	s10				r1	0.013191713	0.092006053269344	0.999999603141647	0.999999588340048	adam
8	s15	s16	s15			r1	0.020294227	0.099553986407594	0.999999286417734	0.999999158717243	adam
9	t9	t8	t12			r1	0.014065002	0.039246385717629	0.99999975806l853	0.999999610442067	adam
10	r6	r6	r6			r1	0.081889670	0.431025828659969	0.999979942760884	0.999979234259338	adam
11	r4	s9	t4			r1	0.035440434	0.274405803025203	0.999997457224921	0.999997272738511	adamax
12	s6	s6	s6	s6		r1	0.036697257	0.135077410194100	0.999997674623557	0.999997648248472	adam
13	t9	t7	t9	t7		r1	0.020655535	0.047895512284143	0.999999216488209	0.999999216472540	adamax
14	t16	s24	t16	s24		r1	0.016188933	0.055l69983722047	0.999999516929888	0.999999495978072	adamax
15	t29	t29	t29	t29	t29	r1	0.007262473	0.052786902569457	0.999999859424823	0.999999858706901	adamax
16	r45	r25	r45	r25		r1	0.035377740	0.264964427292781	0.999995131642068	0.999995l01975695	adamax
17	s6	s8	s6	s8	s6	r1	0.016293276	0.270034714557008	0.999999057384543	0.999999057338298	adam
18	s32	s32	s32	s32		r1	0.014759447	0.051795883990602	0.999999504152263	0.999999502219333	adamax
19	s20	s20	s20	s20	s20	r1	0.038427014	0.160963936663941	0.999998918071426	0.999997319827528	adam
20	t40	t40	t40	t40		r1	0.004961211	0.036747206535928	0.999999941976250	0.999999941870834	adamax
21	s60	s60	s60	s60		r1	0.069739390	0.142438445701024	0.999998919268325	0.999993310642672	adam

In the dataset 3 with 500000 data points, a single random state was tried to cover the whole range of pressure drop values due to a large number of values in the test set (25000). Hence, 475000 values were used to train the networks. The maximum and minimum values of the test set were 44.7642095277823 kPa and 12.9147189854906 kPa respectively, which are close to the extrema of the pressure drop function, which are 12631.94700467 Pa and 44768.99578927 Pa. Overall 25 networks were tried with a maximum number of layers 5 this time, and the networks took 22 -293 minutes to run. The results are shown in table-3.

Table 3. The Results of the ANN architectures tried on 500000 samples dataset

S.No.	H1	H2	H3	H4	H5	Q	MAPE (%)	ME	EVS	R2 Score	Optimizer
1	s88					r1	0.018028073	0.283800499593645	0.999998586119223	0.999998580944023	adamax
2	r150					r1	0.553831640	1.172167198812390	0.999923775759280	0.999508594735740	sgd
3	t78					r1	0.043879500	0.391703026449114	0.999993849159885	0.999993797195402	adamax
4	t22	t18				r1	0.007865883	0.050482170736224	0.999999892641000	0.999999864578462	adamax
5	r33	r44				r1	0.032210420	0.290914909994036	0.999995264232565	0.999995176046473	adamax
6	s15	t20				r1	0.009629170	0.085348503744036	0.999999814422981	0.999999797412340	adamax
7	s60	t70				r1	0.009306306	0.097250359212786	0.999999767278212	0.999999766746936	adamax
8	t40	t30				r1	0.004952371	0.046888725912005	0.999999942181792	0.999999939546673	adamax
9	t60	t50				r1	0.005106382	0.054304497396380	0.999999931553378	0.999999927661881	adamax
10	t50	t40				r1	0.004023054	0.032240288412005	0.999999956925613	0.999999956924265	adamax
11	s8	t10	s8			r1	0.019870633	0.112860100423723	0.999999173075410	0.999999169389358	adam
12	t6	t6	t6			r1	0.453410630	0.419621765978121	0.999939270301347	0.999700181143811	sgd
13	t55	r77	t45			r1	0.011517203	0.056761162435443	0.999999724936100	0.999999724581302	adamax

14	r44	r44	r44			r1	2.361609500	7.982511894857310	0.985317939240120	0.985306399337862	adagrad
15	t40	t40	t40	t40		r1	0.039482567	0.071852104818255	0.999998657544442	0.999997789698808	adam
16	r26	r42	r42	r26		r1	0.056255040	0.316641228353411	0.999988204428046	0.999988173675925	adamax
17	s32	s16	s32	s16		r1	0.060928680	0.226358056135346	0.999993167983045	0.999991669828757	adam
18	s57	s55	s57	s55		r1	0.032887783	0.201414482747942	0.999997549456732	0.999997547858433	adamax
19	r8	s16	t32	r64		r1	0.078608215	0.467230217611223	0.999985784484047	0.999983357077932	adam
20	t29	t32	t29	t29	t32	r1	0.011806865	0.048731664995685	0.999999696509332	0.999999696249580	adamax
21	s16	s18	s16	s18	s16	r1	0.067833130	0.177631020612885	0.999997807236903	0.999992722292200	adam
22	t70	t70	t70	t70	t70	r1	0.049979390	0.070800185136626	0.999999423231475	0.999996887921448	adamax
23	s40	s40	s40	s40	s40	r1	0.021841615	0.157369256086518	0.999998915594956	0.999998906702530	adam
24	t9	t12	t9	t12	t9	r1	0.021615993	0.131559746419817	0.999999035776699	0.999999032613591	adamax
25	s12	t14	s12	t14	s12	r1	0.160978930	0.623590843831927	0.999948150656801	0.999947943079733	adamax

The best performing network are s20-s20-r1 with ‘adam’ optimizer in dataset 1, t40-t40-t40-t40-r1 with ‘adamax’ optimizer in dataset 2 and t50-t40-r1 with ‘adamax’ optimizer in dataset 3 in terms of all four accuracy metrics, with MAPE particularly being around 0.004%, 0.005% and 0.004% respectively. Networks which are either too shallow (one hidden layer) or too deep (three, four hidden layers) doesnot perform well. Hence it is not always necessary that deeper networks will perform better. Second best in terms of MAPE and EVS score are s10-s10-r1 with ‘adam’ optimizer, t29-t29-t29-t29-t29-r1 with ‘adamax’ optimizer and in terms of ME, t10-r20-t10-r20-r1 with ‘adamax’ optimizer, t9-t8-t12-r1 with ‘adam’ optimizer in dataset 1, dataset 2 respectively. In dataset 3 second best is t40-t30-r1 in all four metrics with ‘adamax’ optimizer. In general, networks having sigmoid function in all hidden layers with sufficient number of nodes and ‘adam’ optimizer in dataset 1, tanh function and ‘adamax’ optimizer in dataset 2 and dataset 3 did better. The worst performance in all four metrics is shown by r8-r60-r1 with ‘adagrad’ optimizer, with MAPE around 2.74% in dataset 1, t24-t28-r1 with ‘adagrad’ optimizer, with MAPE around 26.77% in dataset 2 and r44-r44-r44-r1 with ‘adagrad’ optimizer, with MAPE around 2.36% in dataset 3. Second worst performance in all four metrics are r25-r87-r27-r1 with ‘adagrad’ optimizer with MAPE around 0.76% in dataset 1, s28-r1 with ‘sgd’ optimizer with MAPE around 0.26% in dataset 2 and r150-r1 with ‘sgd’ optimizer with MAPE around 0.55% in dataset 3.

After finding the best ANN architecture for each dataset, it is also important to see how these networks perform when trained on another dataset. The best architectures from each dataset were tested with another datasets.

CONCLUSIONS

The performance shown by architectures from all datasets adam, adamax optimizers with tanh, sigmoid functions for the hidden layer and ‘ReLU’ only for output layer are recommended and also two to four hidden layers are enough with number of neurons being 10 to 50.

REFERENCES

Aziz, Z. B., & Klinzing, G. E. 1988. Plug flow transport of cohesive coal: horizontal and inclined flows. Powder technology, **55**(2): 97-105.

Aziz, Z. B., & Klinzing, G. E. 1990. Dense phase plug flow transfer: The 1-inch horizontal flow. Powder Technology, **62**(1): 41-49.

Borzone, L. A., & Klinzing, G. E. 1987. Dense-phase transport: vertical plug flow. Powder technology, **53**(3): 273-283.

Chen, D., Klausner, J. F., & Mei, R. 2002. A fluid mechanics approach to describing the behavior of pneumatically conveyed powder plugs. Powder technology, **124**(1-2): 127-137.

Geldart, D. 1973. Types of gas fluidization. Powder technology, **7**(5): 285-292.

Govier, G. W., & Aziz, K. 1972. The flow of complex mixtures in pipes (Vol. 469). New York: Van Nostrand Reinhold Company.

Hausner, H. H. 1967. Friction conditions in a mass of metal powder. Polytechnic Inst. of Brooklyn. Univ. of California, Los Angeles.

Hong, G. U., & Klinzing, G. E. 1989. Vertical plug flow of cohesive coal in 2-and 4-inch pipes. Powder technology, **57**(1):59-67.

Janssen, H.A. 1895. On the measurement of pressures in grain silos. Z. Ver. Dtsch. Ing., **39**: 1045-1049.

Kalman, H., & Rawat, A. 2020. Flow regime chart for pneumatic conveying. Chemical Engineering Science, **211**:115256.

Konrad, K. 1980. Prediction of the pressure drop for horizontal dense phase pneumatic conveying of particles. Proc. of Pneumotrasport 5, Paper El.

Mi, B., & Wypych, P. W. 1995. Investigations into wall pressure during slug-flow pneumatic conveying. Powder technology, **84**(1): 91-98.

Muschelknautz, E., & Krambrock, W. 1969. Vereinfachte Berechnung horizontaler pneumatischer Förderleitungen bei hoher Gutbeladung mit feinkörnigen Produkten. ChemieIngenieur Technik, **41**(21):1164-1172.

Offor, U. H., & Alabi, S. B. 2016. Artificial neural network model for friction factor prediction. Journal of Materials Science and Chemical Engineering, **4**(7): 77-83.

Pan, R., Mi, B., & Wypych, P. 1994. Pneumatic conveying characteristics of fine and granular bulk solids. KONA Powder and Particle Journal, **12**: 77-85.

Pan, R., & Wypych, P. W. 1997. Pressure drop and slug velocity in low-velocity pneumatic conveying of bulk solids. Powder technology, **94**(2): 123-132.

Rabinovich, E., Freund, N., Kalman, H., & Klinzing, G. 2012. Friction forces on plugs of coarse particles moving upwards in a vertical column. Powder technology, **219**: 143-150.

Rawat, A., & Kalman, H. 2019. Modification and validation of particulate plug-I pressure drop model. Powder Technology, **347**: 243-254.

Sablani, S. S., Shayya, W. H., & Kacimov, A. 2003. Explicit calculation of the friction factor in pipeline flow of Bingham plastic fluids: a neural network approach. Chemical Engineering Science, **58**(1): 99-106.

Shaul, S., & Kalman, H. 2015. Three plugs model. Powder Technology, **283**: 579-592.

Wilson, K.C., Streat, M., & Bantin, R.A. 1972, September. Slip model correlation of dense two phase flow. In Proc. 2nd Int. Conf. on Hydraulic Transport of Solids in Pipes, BHRA, Cranfield, UK, Paper B (Vol. 1).

Yi, J. 2001. Transport boundaries for pneumatic conveying.