AISTech 2019 — Proceedings of the Iron & Steel Technology Conference
6–9 May 2019, Pittsburgh, Pa., USA
DOI 10.1000.377.227

Development of a Predictive Tundish Open Eye Model Using Artificial Neural Network (ANN)

Alvin Ma[1], Raghav Mittal[2], Saikat Chatterjee[2,3], Kinnor Chattopadhyay[1,3]

[1]Process Metallurgy Research Labs (PMRL), University of Toronto,
184 College Street, Suite 140, Toronto, ON, Canada, M5S3E4
Email: kinnor.chattopadhyay@utoronto.ca

[2]M. N. Dastur & Co. (P) Ltd.
P-17 Mission Row Extension
Kolkata 700013, India
Email: Raghav.M@dastur.com

[3]Dastur Innovation Labs
250 Yonge Street, Suite 2201
Toronto, ON M5B 2L7, Canada
Phone: +(1) - 647-6-DASTUR, +(1) - 647-632-7887
Email: Chatterjee.Saikat@dastur.com

Keywords: Steelmaking, Tundish Open Eye, Machine Learning, Artificial Neural Network

INTRODUCTION

Tundish Open Eye (TOE) formation is a phenomenon that occurs due to the sweeping off of the overlying slag layer by the strong recirculatory flows generated by the upwelling argon gas-liquid steel plume[1-4]. The exposed layer of liquid steel readily reacts with ambient air to form harmful inclusions that may directly pass on to the molds, thereby degrading the quality of semi-finished products such as billets, blooms or slabs. The inclusions can also get attached on the inner walls of the Submerged Entry Nozzles (SENs) and clog them, resulting into replacing them prematurely. Apart from SEN life reduction, the overall productivity of the continuous casting process also suffers. A schematic representation of the process of formation and evolution of TOE along with its detrimental effects is shown in Figure 1. Since these issues have a direct impact on steel quality, productivity and thus, profitability of steel making process, an efficient and practically implementable solution is the need of the hour.

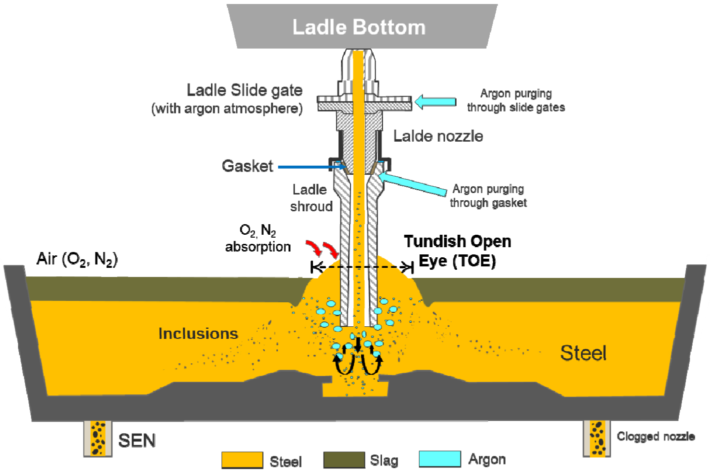


Figure 1: Schematic diagram depicting the formation of tundish open eye during the continuous casting operation.

Previous research on TOEs has focused on physical[1,2], mathematical[1,4] and mechanistic[3] modeling. Although these models can fairly reproduce the phenomenon of TOE and provide us a fundamental understanding of the problem, they are unable to perform on-line and dynamic predictions. In the recent past, machine learning has seen great success due to its versatility and strength in making predictive models. These models can perform predictions in real-time, once they are trained to do so. Artificial neural networks (ANN) based models have flexibility in data input and variable relationships makes it an ideal model for many different applications. There are just a handful of cases[5-9], where machine learning algorithms based on neural network, have been utilized. The present work demonstrates the viability of using machine learning techniques (ANN) for predicting open eye formation during the continuous casting operation.

TUNDISH OPEN EYE MODELING

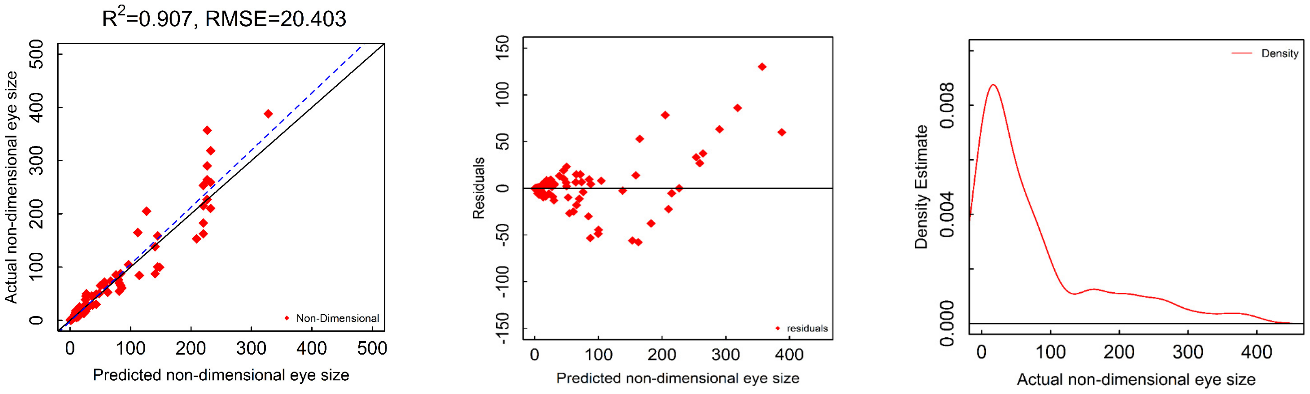
Computational fluid dynamic (CFD) modeling

Multiphase flows encountered during steelmaking and continuous casting operations are often modeled using Computational Fluid Dynamics, where the turbulence in flow and interaction between different phases such as liquid steel, slag, argon and inclusions can be simulated in great details. While the turbulence is generally modeled using *k-ω*, *k-ε* [10] type of models, the interfaces between steel-slag or steel-argon bubbles can be precisely tracked using the Volume of Fluid (VOF) method[11]. Discrete phase modeling[12] is used to track the trajectories of inclusions or argon bubbles within the liquid steel phase. Recently, CFD modeling of tundish open eye and ladle to tundish transfer operations has been performed by various researchers[4,13] in order to gain fundamental understanding of the problem. Although these models are able to accurately predict open eye area size as a function of various process parameters such as tundish dimensions, argon gas flow rate, steel and slag height etc, they are not suited for making predictions in real time.

Non-dimensional regression

In a recent study by Chatterjee and Chattopadhyay[2], experimental results from water model tests were considered for development of a regression-based correlation using dimensional analysis. A correlation for predicting dimensionless tundish open eye area was obtained as a function of various dimensionless numbers such as Froude number, Reynolds number and density ratio as follows:

$$A_e^* = 282.289\left(\frac{U_p^2}{gh}\right)^{1.766}\left(\frac{\Delta\rho}{\rho_l}\right)^{1.588}\left(\frac{V_s}{hU_p}\right)^{0.089} \tag{1}$$



(a) Accuracy of predictive correlation

(b) Residuals of correlation

(c) Density of data utilized

Figure 2. Calculated results obtained from Chatterjee and Chattopadhyay's[2] correlation

The correlation predicts the eye area quite accurately at lower values, as observed from the results shown in Figure 2(a). However, the predictive ability seems to suffer at larger open eye areas. The spread in the residuals at smaller and larger areas, shown in Figure 2(b), also proves the inability of Chatterjee and Chattopadhyay's[2] model to make accurate predictions at larger areas. These inconsistencies can be attributed to the following factors:

- lack of data at larger areas, which is evident from Figure 2(c) and
- inadequate model formulation

Additionally, these empirical models are developed based on certain experimental dataset, thus making it very difficult to get accurate predictions over the entire operating regime. These models also require many model (empirical) parameters, thus negatively affecting their predictive ability outside the tested range.

MODEL DEVELOPMENT: ARTIFICIAL NEURAL NETWORK

Data Visualization

The data in this study has been taken from Chatterjee and Chattopadhyay's[2] work on non-dimensional regression modelling of non-dimensional tundish open eye sizes. In its entirety, the dataset includes 20 predictors such as slag depth, gas flow rate, physical properties of different phases etc. and 1 target variable: non-dimensional open eye size.

The host of variables which was used in this model represents those used to form Chatterjee and Chattopadhyay's[2] model. Akin to describing an apple through its colour, its texture and taste, these variables seek to describe the process from multiple different angles. This type of engineering, coined feature engineering, is a well developed method which can significantly increase the performance of models with existing data[14].

$$\text{Correlation} = \frac{\text{Covar}(x,y)}{\sqrt{\text{Var}(x)\times\text{Var}(y)}} \quad\text{..(2)}$$

where

$$\text{Var}(i) = \frac{\sum(i-\bar{\imath})^2}{n}$$

$$i = x,y \quad\text{..(3)}$$

$$\text{Covar}(x,y) = \frac{\sum(x-\bar{x})\times(y-\bar{y})}{n} \quad\text{..(4)}$$

Table 1: Variable legend for convenient model formulation

Slag height (m)	x1	H (Bath height)/d_b	x12
Bath height (m)	x2	Dimensionless U_p	x13
Water flow rate (litre/min)	x3	U_p, m/s	x14
Gas flow rate (litre/min)	x4	Fr = U_p^2/gh	x15
avg bubble diameter(mm)	x5	$\Delta\rho / \rho_l$	x16
pct gas injection	x6	1 / Re = ν_{slag}/hU_p	x17
Density of slag (ρ_{slag})	x7	log_Fr = U_p^2/gh	x19
Density diff ($\rho_l - \rho_{slag}$)	x8	log_$\Delta\rho / \rho_l$	x20
Dynamic Viscosity of upper phase	x9	log_1 / Re = ν_{slag}/hU_p	x21
Kinematic Viscosity of upper phase	x10	Actual Area	Y
$gd^3/\mu_l^2\rho_l^2$ *10^8	x11		

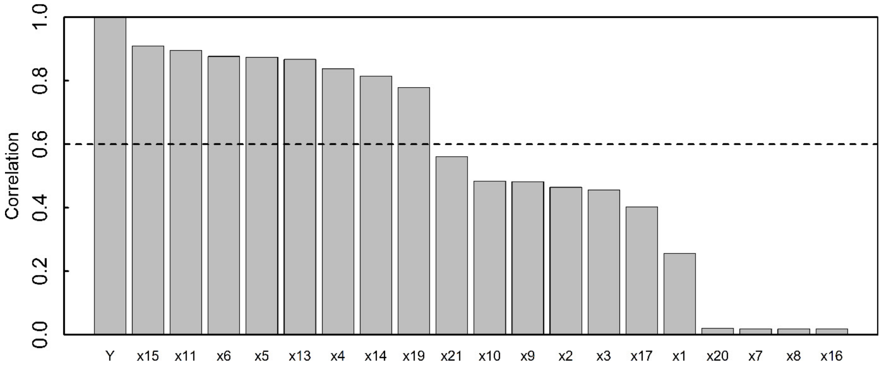


Figure 3. Correlation values for all variables in Chatterjee and Chattopadhyay's[2] dataset

Because the dataset contains many variables, each with its own relationship with respect to the open eye size, the dataset needs to be trimmed to allow for easier model computation. Using correlation as defined in Equation 2, variables which do not appear to be strongly correlated to the eye size can be effectively removed from further model formulation. For ease of viewing, figures in this section have been renamed through Table 1. To ensure that only the variables which have high correlation are used, a correlation cut-off of 0.6 is defined and only those above will be utilized for model formulation. The threshold along with variables with and without sufficient correlation have been plotted in Figure 3. A scatterplot of the qualified variables, exhibiting their inter-relationships, have also been included in Figure 4.

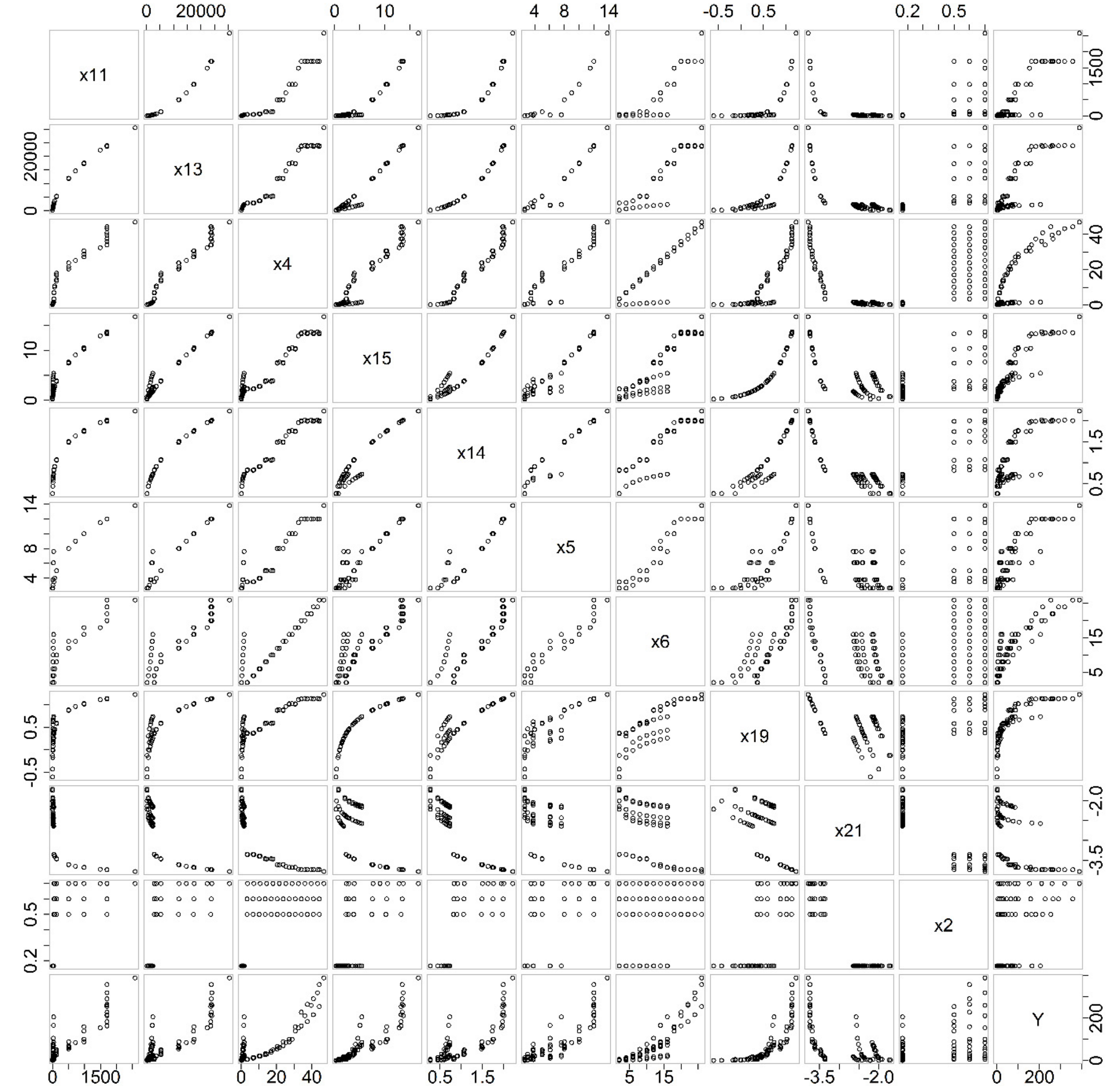


Figure 4. Relationship between highly correlated variables from Chatterjee and Chattopadhyay's[2] dataset

Optimization of Artificial Neural Networks

Background

Linear regression models, though reasonable for capturing simple relationships between variables, do not perform well when there is a complex interplay of relationships between different parameters. For more intricate functions, ANN models are highly capable of capturing the non-linear relationship in between the inputs and the output. The goal in the modeling process was to obtain the correct weights for which the error between model predicted values and actual experimental values are minimized (Equation 5).

$$\text{Cost function} = \frac{1}{2}\sum\left(Y-\hat{Y}\right)^2 \text{ ………………………..(5)}$$

In the case of neural networks, the predicted value can be described by a series of successive relationships which describe the connections between nodes, synapses, outputs and inputs. For a single layer structure this can be described as the following:

$$f^{(2)}\left(f^{(1)}\left(\begin{bmatrix} X_{1,1} & \cdots & X_{1,i} \\ \vdots & \ddots & \vdots \\ X_{n,1} & \cdots & X_{n,i} \end{bmatrix} \times \begin{bmatrix} W_{1,1}^{(1)} & \cdots & W_{1,k}^{(1)} \\ \vdots & \ddots & \vdots \\ W_{i,1}^{(1)} & \cdots & W_{i,k}^{(1)} \end{bmatrix}\right) \times \begin{bmatrix} W_{1}^{(2)} \\ \vdots \\ W_{k}^{(2)} \end{bmatrix}\right) = \hat{Y} \qquad (6)$$

where *f* is a logistic function which introduces non-linearity into the model and is applied twice, once prior to and once post nodes. Visually, a single hidden layer architecture can be seen in Figure 5.

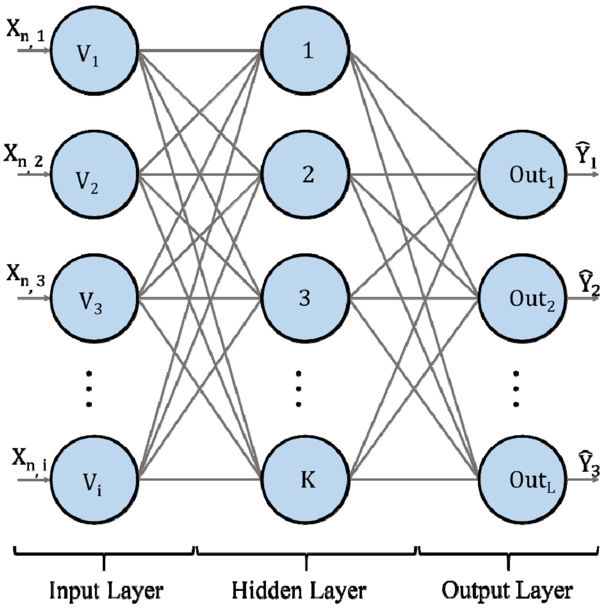


Figure 5. A generic Neural Network

Obtaining the weights for all the nodes in the network architecture is a simple exercise in identifying the minimum point of the cost function. However, it is the high dimensional problems that suffer from the curse of dimensionality. Thus, solving for all domain and range at sufficient resolution through utilizing brute force techniques, becomes impossible for complex systems. To address this, a gradient descent algorithm is typically employed to aid in converging to the final solution. Here an analogy can be drawn with a ball placed at a random point on a hypothetical curved plane. With its trajectory determined by gravity, the ball naturally rolls towards the minimum. The gradient descent algorithm functions similarly and finds the direction of minimum slope iteratively. This gives way to a simpler and much faster convergence of the cost function. Step size and learning rates of the algorithm must be carefully selected to avoid errors associated with the use of this type of algorithm, as mentioned in previous works[15].

Neural network creation and validation

The present work has employed the open source programming language *R*[15] along with an artificial neural network model for regression using the logistic function, developed by Anastasiadis[16]. Adaptive learning, wherein the learning rates are adaptive and hence do not require advance knowledge of optimal learning parameters, is used by this method for getting results with better convergence and stability[17,18]. Figure 5 shows the key-points pertaining to model formulation via a simple schematic diagram.

The entire dataset has been split into two subsets: Training Set and Test Set. While 60% of the data has been used for training of the neural network and fitting relation between predictors and the target, the remaining has been used for validation. The training and test set split was selected arbitrarily in order to avoid any bias or overtraining. A simple flow chart detailing the entire process has been included in Figure 6.

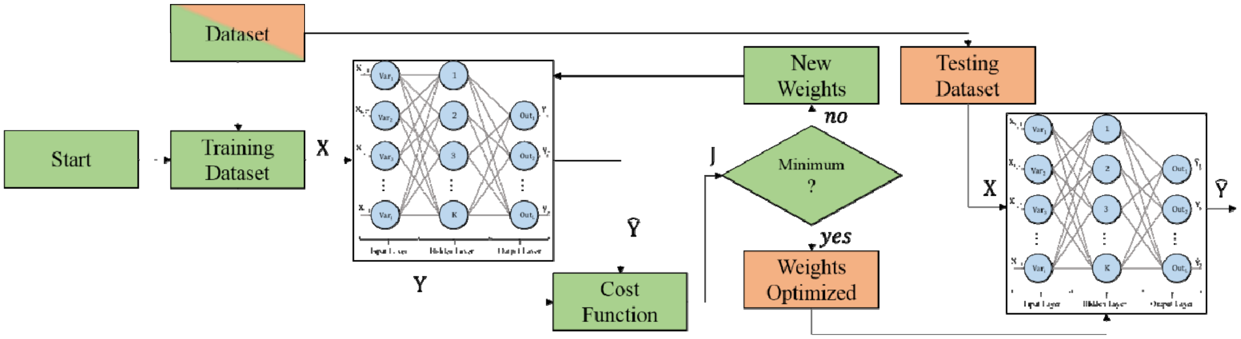


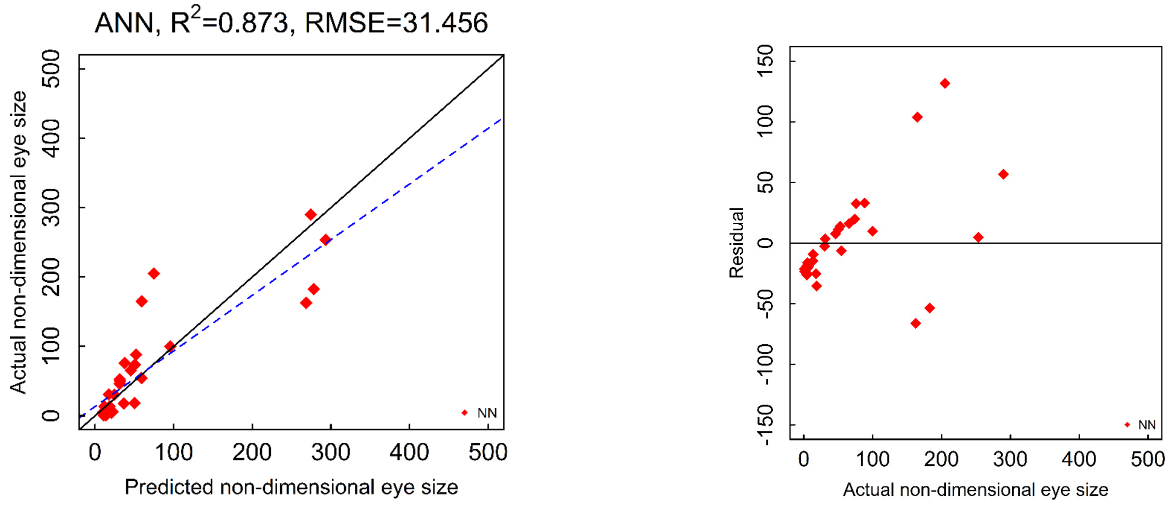
Figure 6. Flow chart of a single neural network model

RESULTS & DISCUSSION

Performance of ANN Models

Single Model Neural Network

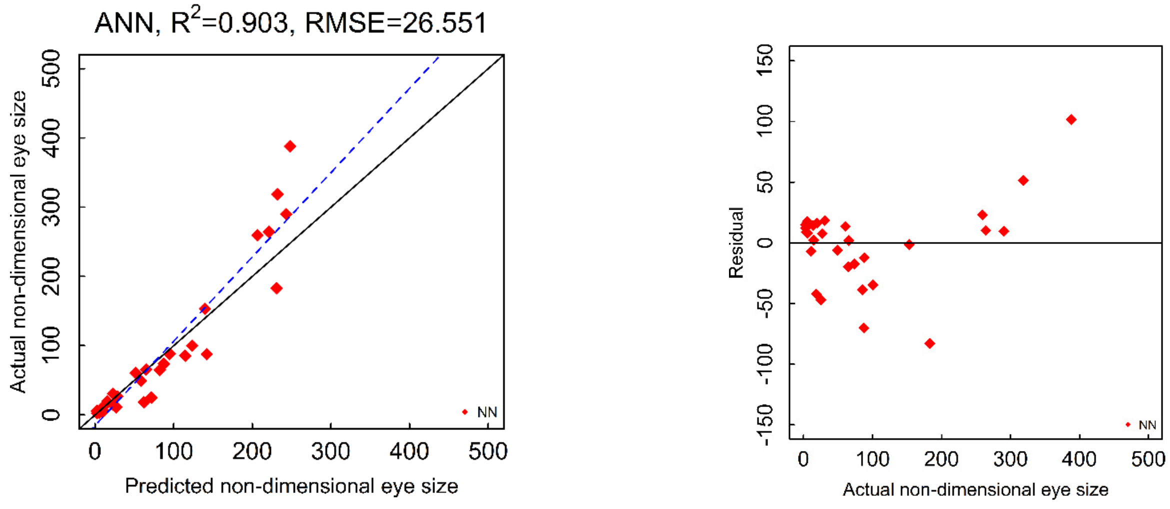
A comparative analysis was carried out based on the above criteria, for various ANN models with a single hidden layer comprising 4, 6 and 8 nodes. After training and deploying the model, the results obtained have been recorded in Table 2. The R^2, and RMSE metrics chosen for performance evaluation are similar to those seen in Chatterjee and Chattopadhyay's[2] work. Unfortunately though, quite similar to Chatterjee and Chattopadhyay's[2] work, increased variance was seen at higher open eye sizes for all cases. This observation suggested that it was most likely that a single layered (hidden) neural network may not be ideal for capturing the complex process. Plots showing actual versus predicted non-dimensional eye sizes for 4, 6, and 8 nodes have been included in Figure 7 to Figure 9 respectively.



(a) Accuracy of prediction

(b) Residuals

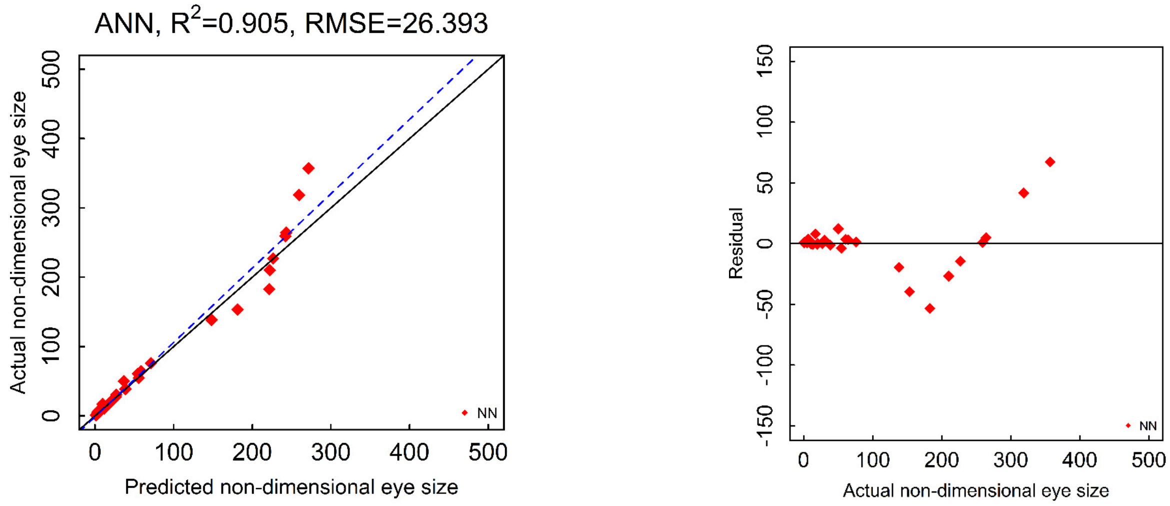
Figure 7. Calculated results for a 4-node neural network



(a) Accuracy of prediction

(b) Residuals

Figure 8. Calculated results for a 6-node neural network



(a) Accuracy of prediction

(b) Residuals

Figure 9. Calculated results for a 8-node neural network

Table 2. Number of nodes and corresponding performance metrics

	Number of nodes		
	4	6	8
R^2	0.873	0.903	0.905
RMSE	31.456	26.551	26.392

As can be observed from the plots, it is quite obvious that the upper limit of predictive power was reached for this type of architecture. However, further refining is required to stabilize the prediction of larger eye sizes is needed for any useful neural network model. Even though they are marginally better, 8 nodes perform the best and hence they have been used for further model augmentation.

As an aside, previous research on optimized neural network architectures where the number of hidden nodes could be represented by the number of input nodes (i.e. input variables) gave rise to the current 4-8 node range[19-22]. As the current research focuses on the utility of this novel modelling technique, any further original development in architecture and optimization specifically for will be discussed in future works.

Ensemble prediction: bootstrap aggregating

Generally, it is ideal to obtain more data points to strengthen the predictions. However, in practice, there are several constraints that make it difficult. There are other statistical techniques which can strengthen predictive models and thus help in addressing the severe limitations in data points at the higher eye sizes in the current study.

Bootstrap aggregating, as shown in Figure 10, is a two-step statistical technique employed to improve stability and accuracy of machine learning algorithms. The first step, bootstrapping, samples, with replacement, the original data set *m* number of times. This will create *m* number of datasets which are formed from fragments of the original dataset, with possibility of observing repeating segments. Once formed, separate models, totaling *m* number of models, are trained using dataset *m* and deployed against the original dataset. The final step consists of aggregating the predictions. Aggregating predictions can take many forms depending on the goals of the model (i.e., classification versus regression). For regression, average or median of the aggregated predictions are natural choices.

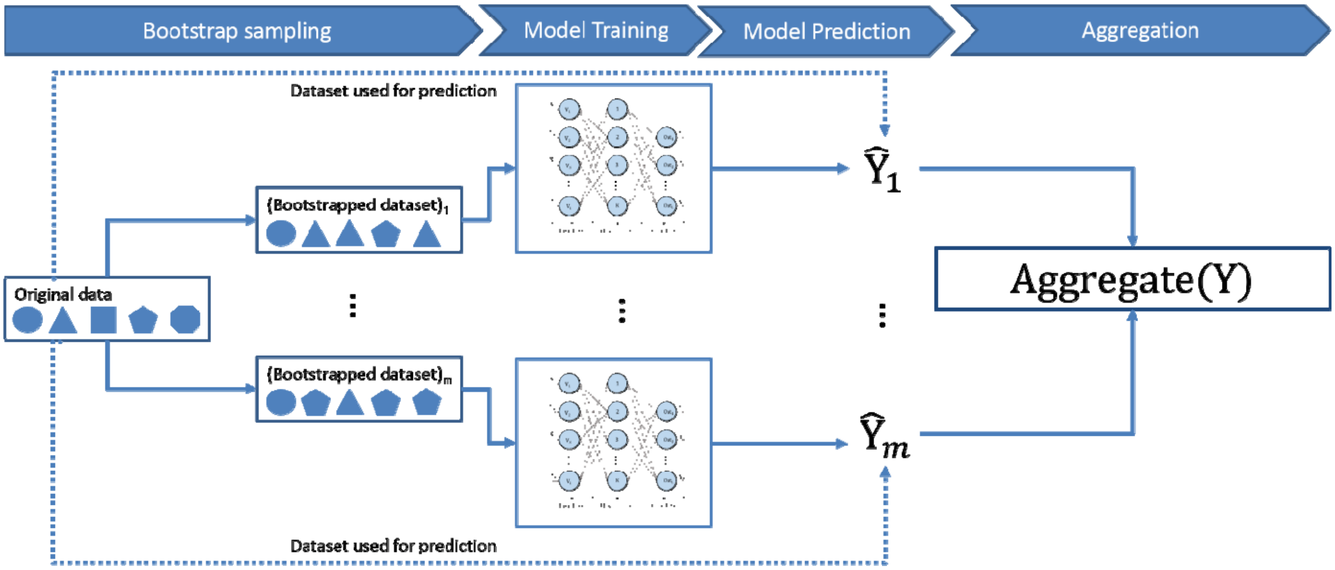
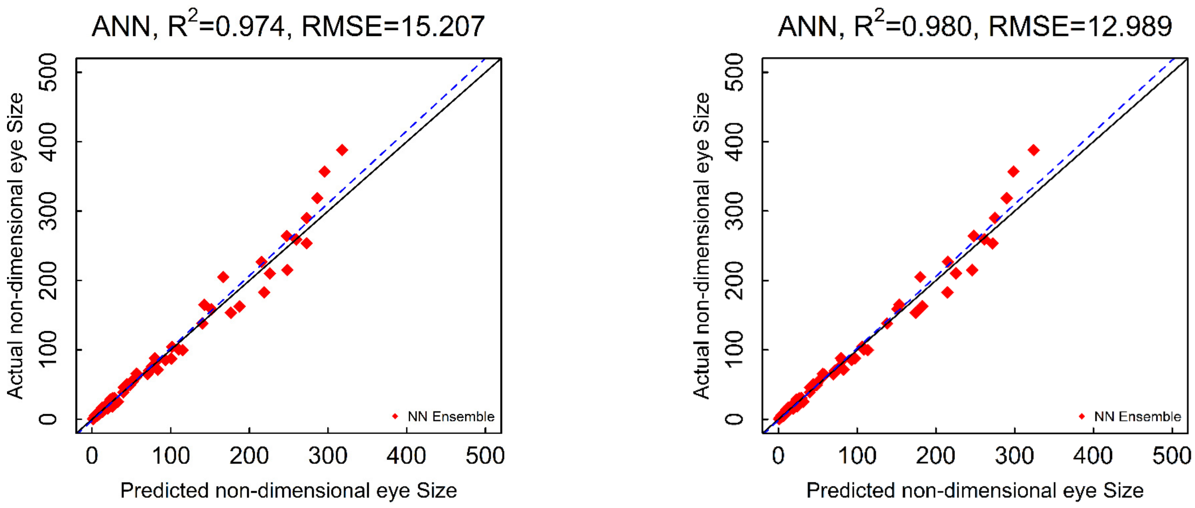

Bootstrap sampling
Model Training
Model Prediction
Aggregation
Dataset used for prediction
{Bootstrapped dataset}$_1$
Original data
{Bootstrapped dataset}$_m$
$\hat{Y}_1$
$\hat{Y}_m$
Aggregate(Y)
Dataset used for prediction

Figure 10. Flow chart of the bootstrapping aggregation process part

In this work, an initial 1000 dataset has been created through sampling, with 60% replacement of the original dataset. An equal number of models will be formulated, and the results aggregated comparing between median and averaged predictions. The remaining 40% was used to test the models.

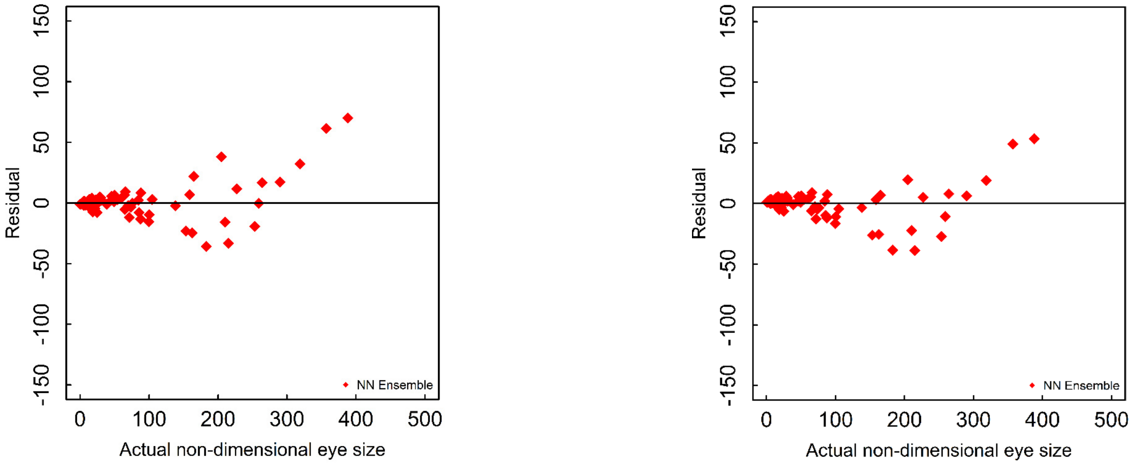
A thousand distinct artificial neural networks were formulated from 1000 unique data sets, in total. Favorable results were observed for the model augmentation through bootstrapping aggregation. A 35% improvement in accuracy is observed in Figure 11. Furthermore, increased stability can also be observed in the residuals, as seen in Figure 12. Sampling and resampling of sparse data points, allowing the model to learn from segments of the data previously learnt poorly, can help explain the reduced spread of this model augmentation method. Despite the stronger accuracy, an element of randomness can still be observed at the higher open eye areas.



(a) Aggregated average

(b) Aggregated median

Figure 11. Calculated results of ensemble neural network



(a) Aggregated average

(b) Aggregated median

Figure 12. Residuals of ensemble neural network

Optimum number of neural networks

This ensemble comprised the generation of 1000 unique neural network models from equally different datasets. Aggregating these models gave us the predicted values. This method has been demonstrated to be much more accurate than regression and single neural network models. However, it must be noted that developing 1000 neural networks still may not be feasible for larger datasets, although such a system may be simple to parallelize. Hence, it is important to define a point where additional neural networks do not contribute to improving the overall accuracy or add negligible value to the predictions. A running calculation of RMSE was tracked for both aggregate techniques (Figure 13) during the formation of the 1000 models. Two features can be observed from the running RMSE plot: 1) performance of the model was observed to be consistently better for the median aggregate technique than the average and 2) past 600 neural networks the RMSE stabilizes for both techniques. This suggests that for this particular system, for a stable prediction, over 600 different models need to be formulated.

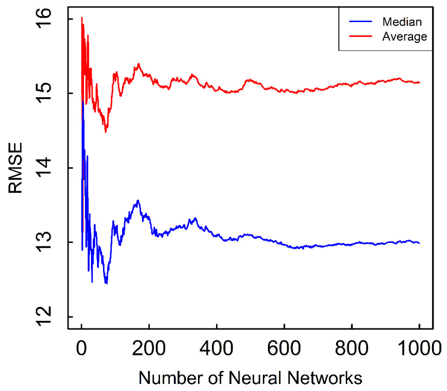


Figure 13. RMSE stability over n-trained Neural Networks

CONCLUSION

The tundish open eye is a complex phenomenon whose effect on steel has not been fully understood yet and is an ongoing endeavor. Equally complex is the process in which it forms because of its inherent multi-phase and multi-physics nature. Researchers have sought to understand the process through fundamental analysis, experiments, physical model formulation, and CFD modelling. While impactful within the realm of fundamental understanding, a static model is not pragmatic in an industrial setting.

Conversely, the modelling method shown in this work, while relatively simple to implement, has profound predictive capabilities. The base data was gathered and clustered to identify those having high correlation. From here a model was built and augmented through boot strap aggregation. With enough data, some basic feature engineering, and a widely available machine learning framework, a complex phenomenon such as the tundish open eye can be better anticipated.

In a plant setting, this methodology can be further augmented via two avenues: 1) with the addition of sensor data, a more robust estimation of process parameters will become a reality via methods such as Kalman filters and 2) parallelizable model training and execution can sustain a population of models capable of making predictions in near real-time. The combination of both streams will bolster any prediction and by extension better prescription at the shop floor.

NOMENCLATURE

ANN	Artificial neural network
$\mathrm{Covar}(x,y)$	Covariance function
$f^{n}(\ )$	Activation function where *n* is the layers of synapses between nodes
$f_{S^{2-}}$	Activity coefficient of ionic sulphur
Fr	Froud number
g	Acceleration due to gravity (ms^{-2})
h	Depth of upper fluid phase (m)
J	Cost function for the gradient descent algorithm
Log	Base 10 logarithmic function
$\Delta\rho$	Density difference between lower and upper fluid phase (kgm^{-3})
ρ_l	Density of bulk fluid phase (kgm^{-3})
R^2	Coefficient of multiple determination
Re	Reynolds number
RMSE	Root mean square error
U_p	Plume velocity (m/s)
V_i	Variable count (I = 1,2,3…)
V_s	Kinematic viscosity of upper fluid phase (m^2s^{-1})
$\mathrm{Var}(i)$	Variance function

$W_{i,k}$	Matrix of synapse weights prior to hidden node where *i* is the number of input variables and *k* is the amount of hidden nodes
W_k	Matrix of synapse weights post-hidden nodes where k is the number of hidden nodes
$X_{n,i}$	Dataset matrix where *n* represents the amount of data points ($\eta = 1,2,3\ldots$) and *i* the amount of variables
$\overline{x}$	Average of predictor
Y	Actual non-dimensional open eye size
$\hat{Y}$	Predicted non-dimensional open eye size
$\overline{y}$	Average non-dimensional open eye size

REFERENCES

1. S. Chatterjee and K. Chattopadhyay, ISIJ Int., 55 (2015), 1416.
2. S. Chatterjee and K. Chattopadhyay, Metall. Mater. Trans. B, 47 (2016), 508.
3. S. Chatterjee and K. Chattopadhyay, Metall. Mater. Trans. B, 47 (2016), 3099.
4. S. Chatterjee and K. Chattopadhyay, Metall. Mater. Trans. B, (201), DOI https://doi.org/10.1007/s11663-018-1177-z.
5. A. González-Marcos, J. Ordieres-Meré, F. Alba-Elías, F.J. Martínez-De-Pisón, and M. Castejón-Limas, Ironmak. Steelmak., 41 (2014), 262.
6. J. Mori and V. Mahalec, Expert Syst. Appl., 49 (2016), 1.
7. J. Mori and V. Mahalec, Comput. Chem. Eng., 79 (2015), 113.
8. E. Palaneeswaran, G. Brooks, and X.B. Xu, Metall. Mater. Trans. B Process Metall. Mater. Process. Sci., 43 (2012), 571.
9. L.Z. Yang, R. Zhu, K. Dong, W.J. Liu, and G.H. Ma, Study of optimizing combined-blowing in EAF based on K-medoids clustering algorithm, in: Proc. *3rd Int. Conf. Chem. Metall. Eng., Zhuhai, China* (1540)(2014), p. 1540.
10. B.E. Launder and D.B. Spalding, *Comput. Methods Appl. Mech. Eng*., 3 (1974), 269.
11. C.. Hirt and B.. Nichols, *J. Comput. Phys*., 39 (1981), 201.
12. Y.A. Buevich, *Fluid Dyn*., 1 (1966), 119.
13. D. Mazumdar , P.K Singh, R.K.Tiwari: ISIJ International, vol 58 (2018), p.1545-1547.
14. R. Szeliski, *Computer vision: algorithms and applications*, Springer Science & Business Media, (2010).
15. A. Ma, S. Mostaghel, and K. Chattopadhyay, ISIJ Int., advpub (2016).
16. A.D. Anastasiadis, G.D. Magoulas, and M.N. Vrahatis, Neurocomputing, 64 (2005), 253.
17. R. Liu, G. Dong, and X. Ling, A convergence analysis for neural networks with constant learning rates and non-stationary inputs, in: Decis. Control 1995 Proc. 34th IEEE Conf. On, IEEE, (1278–1283)(1995), pp. 1278–1283.
18. M. Gori and A. Tesi, *IEEE Trans. Pattern Anal. Mach. Intell.*, 14 (1992), 76.
19. J. Li, T. Chow, and Y. Yu, Estimation theory and optimization algorithm for the number of hidden units in the higher-order feedforward neural network, *Neural Networks*, 1995 Proc. IEEE Conference on Neural Networks(1229-1233)(1995)
20. S. Tamura and M. Tateishi, *IEEE Transactions on Neural Networks*, 8 (1997) 251.
21. K. Shibata and Y. Ikeda, Effect of number of hidden neurons on learning in large-scale layered neural networks, *Proc. of the ICROS-SICE International Joint Conference* (5008–5013)(2009)
22. G. Sheela, S N. Deepa *Mathematical Problems in Engineering*, 6 (2013)