Engineering Science & Technology
https://ojs.wiserpub.com/index.php/EST/

UNIVERSAL WISER PUBLISHER

Research Article

Development of an Automated Property Prediction System for the Pelletizing Process

Pranav Kumar Tripathi*, Sunal S, Manuraj Dadwal, D. Satish Kumar, Prabhat Kumar Ghorui

JSW Steel Ltd, Vijayanagar Works, Bellary, India
E-mail: pranavkumar.tripathi@jsw.in

Received: 3 January 2025; **Revised:** 3 June 2025; **Accepted:** 10 June 2025

Abstract: In an integrated steel plant, the iron ore pellet properties affect the productivity and efficiency of the iron-making units. The pelletizing process is very dynamic and is affected by many raw materials and process variables, which cause fluctuations in the cold and high-temperature properties of the pellets. This necessitates an increase in the frequency of sampling and testing in case of process anomalies, which involves tedious and lengthy testing procedures leading to delays in corrective actions by the operators. To address this issue, an online fully automated machine learning-based prediction system was developed and implemented for the instantaneous prediction of the key pellet quality parameters-Cold crushing strength, Reducibility index, and Tumbler index on a continuous basis. The prediction system was developed using a Gaussian gradient boosting algorithm with more than 1,000 datasets as inputs in Python and *R* language. The developed model showed good accuracy with $R^2 > 0.95$ for the pellet properties. The developed prediction system was implemented on a plant scale through dynamic dashboards with timed automated predictions without manual intervention. A novel 'incremental updating dataset' approach was incorporated, which enabled the model to 'self-learn' from the deviations and ongoing trends in the fresh datasets to deliver more accurate predictions. Post-deployment, the model assisted in taking corrective actions to reduce process deviations and helped reduce sampling frequency by 50% resulting in better manpower utilization and lower annual maintenance costs of the testing equipment, leading to significant monetary savings.

***Keywords*:** iron-ore pelletization, machine learning, Gaussian gradient boosting, incremental updating, self-learning, property prediction model

1. Introduction

Iron ore pelletizing is one of the most commonly used agglomeration techniques in the steel industry to convert the ground iron-ore into iron through iron-making units like blast furnaces, COREX® (developed by Siemens VAI)s, and Direct Reducing Iron (DRI) furnaces, etc. The pelletization process involves a complex array of process steps and is affected by raw material composition, size distribution, mixing, heating, and induration parameters [1]. With the non-availability of good raw materials and frequent changes in the raw material characteristics, significant variation in product quality and productivity is observed. A dynamic control in the process and the induration parameters are required for consistent control of the process. Irregularities in the variables affect the subsequent iron-making processes [2]. As per traditional practice, frequent sampling and analysis are carried out on a routine basis, often multiple times

DOI: https://doi.org/10.37256/est.6220256348

during the day, to continuously monitor the changes in the pellet quality. Pellet quality parameters such as the Tumbler Index (TI), which indicates the resistance of the material to breakage or degradation by impact, the Reducibility Degradation Index (RDI), which is the measure of the ability to generate fines during reduction, and the Cold Crushing Strength (CCS) which quantifies the durability of the pellet during the shipment or transfer in the shaft in the later iron making units, are frequently measured batch to batch to ensure consistent quality of pellets for the next stage of operation [1], [2]. Such high frequency of sampling is tedious, requires large manpower, usage of testing facilities, and suffers from testing delays leading to delays in quality feedback which affects the process control. Additionally, some high-temperature tests for RDI require long hours and there is a large time lag of an average of 6 to 8 hours between sampling and results, which hinders real-time feedback that may lead to non-conforming pellet quality. The use of predictive modeling techniques with proper mathematical modeling equations and algorithms is an efficient and effective way for the determination of the quality parameters [1].

Data analytics is an area of examining, arranging, and transforming the raw process data to obtain useful information about the process. Such analysis is carried out through descriptive, diagnostic, predictive, and prescriptive models, which are based on regression, classification, clustering, and pattern recognition [3]. The most commonly used algorithms are linear regression, neural networks, tree-based algorithms (ensemble machine learning techniques), support vector machines, and *K*-means clustering. A model can be developed with any of these algorithms which would yield values in close agreement with the actual running conditions. Compared to the process-based models, the statistical models implement various data mining techniques, which have low computational requirements and a faster response time [4].

Several attempts have been made earlier for the predictive analysis of the quality parameters in different production processes. A Kernel Principal Component Analysis-Radial Basis Function (KPCA-RBF) algorithm-based CCS prediction model was done where the neural network was trained with the simulated annealing algorithm and fuzzy *c*-means clustering algorithm which eliminated the problems of local minimum [5]. A conventional Generalized Feed Forward neural network with a back propagation error correction technique was carried out for the prediction of the pellet CCS. A total of 12 parameters were used in the model, of which, the CCS was found to be quite sensitive to bentonite concentration, basicity, FeO, and green pellet moisture. The model thus developed had an error percentage of 3% [2]. A model was developed using artificial neural networks by tracing the non-linearity effects to determine the bed permeability by predicting the pellet bed backpressure. The model optimized the standard deviation of pellet diameter, bed height, and sphericity which were the key parameters affecting the bed back pressure and hence the permeability. A neural network model using the conventional back propagation technique predicted the TI, which gave < 3.5% margin of error [6]. A comprehensive model using the various combinations of regression models and selection of input variables by recursive feature elimination, stability selection, and random forest selection was developed, which depicted a good accuracy of 88% [7]. Numerous models have been developed using both the first-principle-based techniques and also neural networks for prediction of properties of iron-ore agglomerates [5]-[12]. As these models are more linearized, they do not account for the operational fluctuations, which has an effect on the actual agglomeration in plant scale operation resulting in lesser accuracy, in the longer run. Due to the increased complexity of heat and mass transfer involving solid to solid and gas to solid reactions, the process is complex and conventional static models have limited utility [10], [13]. Therefore, a real-time forecasting model enables better utility for the effective control of the process.

The majority of the models developed use a static input dataset without any self-learning feature. Moreover, the deployment strategies for such predictive models have often been a user-entry-based interface, which is not user-friendly, as it requires manual input from the user. This involves time and effort from the user and is not conducive to a high production rate system. In the present study, a robust algorithm, based on an ensemble machine learning technique of boosting, has been deployed for the prediction of the pellet properties. The model was enabled to self-learn using the latest data recorded in the plant server, achieving very good prediction accuracy levels under dynamically varying process conditions. A real-time property prediction system was developed and deployed at the plant level, which nearly has zero human interference. The system provides accurate predictions with automated updates and requires minimal manual intervention and monitoring.

1.1 *Process description*

Pelletization of iron ore concentrate is carried out by blending the ore, with limestone, carbon sludge, iron ore

slurry, and additives like bentonite. The ore concentrate, fluxes, and carbon of appropriate composition and size fraction are used. All the ingredients are mixed with appropriate moisture and the binder is made into pellets using a disc pelletizer for the preparation of green balls of optimum size. The green pellets are fired in a seven-zone indurating furnace, at the end of which, hard pellets with the required mechanical, physical, and metallurgical properties are obtained. Some significant properties of the pellet include high iron content (around 63%), particle size variation ranging between 9 mm to 15 mm, high and even porosity of 25-30%, and uniform mineralogical structure with a low tendency to abrasion [1]. The testing was done based on sampling plans as per ISO 3271 and ISO 4700 standards. The quality targets for the pellets are shown in Table 1. These parameters are essential for achieving > 75% shaft metallization, which minimizes pressure drop and ultimately enables high production yields. A brief overview of the pelletization process at JSW Steel Ltd. Vijayanagar Works is shown in Figure 1.

Table 1. Quality targets for the process

Output parameter	Ideal specification limit
Cold Crushing Strength (CCS)	210-240
Reducibility Degradation Index (RDI)	10-16
Tumbler Index (TI)	90-94%

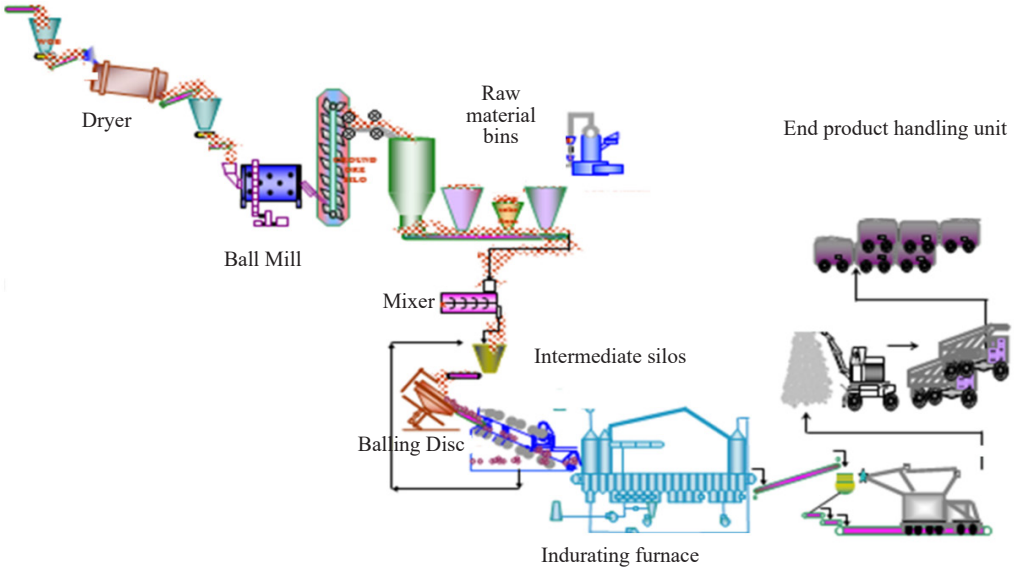

Figure 1. Overview of the pelletizing process

2. Material and method

In this section, the detailed procedure followed for the model development and its performance evaluation have been discussed.

2.1 *Model development based on gradient boosting*

The use of advanced machine learning algorithms involves the usage of historical data to develop prediction models by identifying the patterns and relationships within the data, which are then used to make predictions on new,

unseen data. While linear models, such as linear regression assume a linear relationship between the input features and predictor variable, complex algorithms like Artificial Neural Networks (ANN) and ensemble-based algorithms create and combine multiple decision trees to capture complex, non-linear relationships in the data [11], [12], [14]-[17].

Ensemble models are machine learning algorithms that combine various predictions and give a final prediction based on the majority, thereby reducing the bias and variances and improving accuracy. Also referred to as meta-algorithms, these can be sub-divided based on the way the learning takes place, such as bagging, which has parallel learning, or boosting, which has sequential learning [15], [16], [18].

Bagging involves the generation of base learners in sequence to generalize their dependency which results in a reduction of error [16], [19]. Mathematically, the M different decision trees can be trained using the data, which are randomly selected with the replacement, which infers to the fact that M different trees are used to reduce the variance by creating data subsets for training the base learners, which uses averaging and can be represented as a function as shown in equation (1).

$$f(x) = \frac{1}{M}\sum_{m=1}^{M} f_m(x) \tag{1}$$

These are generally applied in conglomeration with decision trees rather than linearized models [19], [20]. Examples for these models include decision trees and Random Forest.

Boosting on the other strata can convert weak learners to strong learners by fitting a bunch of weak learners, which are usually small decision trees to a weighted version of the data, and just like any ensemble technique, the weighted majority vote is taken as the prediction. Since the present model incorporates the boosting (specifically gradient boosting) algorithm, the mathematical interpretation is as follows [18], [20].

Consider Y_i to be the dependent variable which is a function of the vector of all predictor variables (X_i). Since the predictor cannot fully describe the outcome, there is a bias function which is the error (e_i), as denoted by equation (2).

$$Y_i = F_1(x_i) + e_{1i} \tag{2}$$

In boosting, the weak learners are converted into strong learners and the second iteration (model) will be trained with the error term. The corrected second model is given below (equation (3)).

$$Y_1 - F_1(x_i) = e_{1i} = h_1(x_i) + e_{2i} \tag{3}$$

From equations (1) and (2),

$$F_2(x_i) = F_1(x_i) + h_1(x_i) \tag{4}$$

After fitting the model for the m^{th} boosting iteration step,

$$F_m(x_i) = F_{m-1}(x_i) + h_{m-1}(x_i) \tag{5}$$

Where, $h_{m-1}(x_i)$ is a weak learner, which, on training, becomes a part of a stronger tree. The residual error is expressed as loss function $L[Y, F_m(x)]$ which signifies the difference in actual and predicted value and the ultimate aim is to reduce this loss. This happens by taking the derivative of this loss function concerning the predictor value.

The gradient γ_{im} is defined as:

$$\gamma_{im} = -\left\{\frac{\partial L\left[y_i,\, F_{m-1}(x_i)\right]}{\partial F_{m-1}(x_i)}\right\} \tag{6}$$

Where 'i' is the index for observations, which will have one gradient value. Therefore, the weak learner $h_m(x)$ is fit

by computing γ_{im}, where,

$$\gamma_m = \arg_\gamma \min \sum_{i=1}^{m} L\left[y_i,\ F_{m-1}\left(x_i\right) + \gamma \cdot F_m\left(x_i\right)\right] \tag{7}$$

The solution is,

$$\gamma_m = \frac{\sum_{i=1}^{n} h_m\left(x_i\right)\left[y_i - F_{m-1}\left(x_i\right)\right]}{\sum_{i=1}^{n} h_m\left(x_i\right)^2} \tag{8}$$

The final equation that gives the prediction can be generalized and is updated with

$$F_m(x) = F_{m-1}(x) + \gamma_m h_m(x) \tag{9}$$

Inferring from equation (9), $F_m(x)$ is the final value of the prediction obtained.

A number of different machine learning algorithms have been reported in the literature, each with respective pros and cons. Table 2 provides a detailed comparison of the major machine learning algorithms which can be used for regression tasks [18], [21].

Table 2. Comparison of different machine learning algorithms

ML algorithm	Advantages	Disadvantages	Applications
Multiple Linear Regression (MLR)	• Simple and interpretable • Fast training and prediction • Good baseline model • Handles linear relationships well	• Assumes linear relationships • Sensitive to outliers • Poor performance with non-linear data • Requires feature scaling • Multicollinearity issues	• Linear relationship modeling • Feature importance analysis • Small datasets • Baseline comparisons
Artificial Neural Networks (ANN)	• Handles non-linear relationships • Universal function approximator • Flexible architecture • Good for complex patterns • Can learn feature representations	• Black box model (low interpretability) • Requires large datasets • Prone to overfitting • Computationally expensive • Sensitive to initialization	• Image recognition • Complex non-linear problems • Large datasets • Pattern recognition
Support Vector Machine (SVM)	• Effective in high dimensional spaces • Memory efficient • Versatile (different kernel functions) • Good for small datasets • Good generalization capability	• Slow on large datasets • Sensitive to feature scaling • No probabilistic output • Choice of kernel and parameters critical • Poor performance with noisy data	• Text classification • High-dimensional data • Small to medium datasets • Binary classification
Random forest	• Handles overfitting well • Feature importance ranking • Handles missing values • No need for feature scaling • Robust to outliers • Parallel processing	• Can overfit with very noisy data • Memory intensive • Less interpretable than single trees • Biased toward categorical variables • May not perform well on very small datasets	• Mixed data types • Feature selection • Medium to large datasets • Good accuracy with limited interpretability
Gradient boosting	• Superior predictive performance • Handles various data types • Built-in feature importance • Robust to outliers • Handles missing values • Sequential learning reduces bias	• Prone to overfitting if not tuned • Computationally intensive • Requires hyperparameter tuning • Sensitive to noise • Less interpretable	• Structured/tabular data • Superior performance • Complex feature interactions • High accuracy required

Gradient boosting builds models sequentially, where each subsequent model corrects the errors of previous models. This iterative error correction leads to superior predictive performance compared to individual models or parallel ensemble

methods like Random Forest which primarily reduces variance whereas gradient boosting effectively reduces both bias and variance through its boosting mechanism, leading to better overall model performance. Moreover, it is much more computationally efficient as compared to algorithms like SVM and ANN. Recent research also shows that gradient boosting and Random Forest consistently outperform traditional algorithms like SVM, ANN, and linear regression, with gradient boosting often achieving accuracy rates above 95% compared to approximately 85-90% for other methods in structured data prediction tasks [18]. Henceforth, this technique was used to develop the prediction model because of its inherent ability to improve prediction accuracy by minimizing bias and variance errors through regularization techniques. This leads to a robust model which is capable of capturing the complex non-linear relationships present in the datasets thereby making it more reliable for real-world predictions and often resulting in maximum accuracy when tuned properly [18], [19], [21].

2.2 *Data mining and feature extraction*

For model building, ten months of data from pellet plant 1, JSW Vijayanagar Works was collected. Initially, about 50 variables were selected with more than 1,000 data points to start with. After the data was fetched, it was filtered and subjected to a data wrangling process involving the removal of blank values, NaN (non-numerical) values, and outliers in the data, which account for the erroneous data points that might be due to the fault in the sensor, instrument or manual error. The next important step is normalization, where all the input variables are scaled down to a common scale usually between 0 and 1 to cater to an efficient model building. A typical min-max type of normalization was used. Mathematically, it can be represented as,

$$X_{norm} = \frac{X - X_{min}}{X_{max} - X_{min}} \tag{10}$$

Where X is the value to be normalized, X_{min} and X_{max} refer to the minimum and maximum values of the dataset, and X_{norm} is the scaled-down output value. Once the normalization is done, it is required to find out the strongest predictors that influence the independent variables that are correlated to each other by using relative importance analysis by the principles of feature extraction. Mathematically, the variables are chosen based on a loss function, which is the error function that helps to decide the weights of the inputs for evaluation. Therefore, the drop in the error at each split is calculated. The greater the drop in error, the more is the importance of the variable. However, the inputs from the process control were also taken into account before arriving at the final list of parameters that were used for model building as shown in Figure 2.

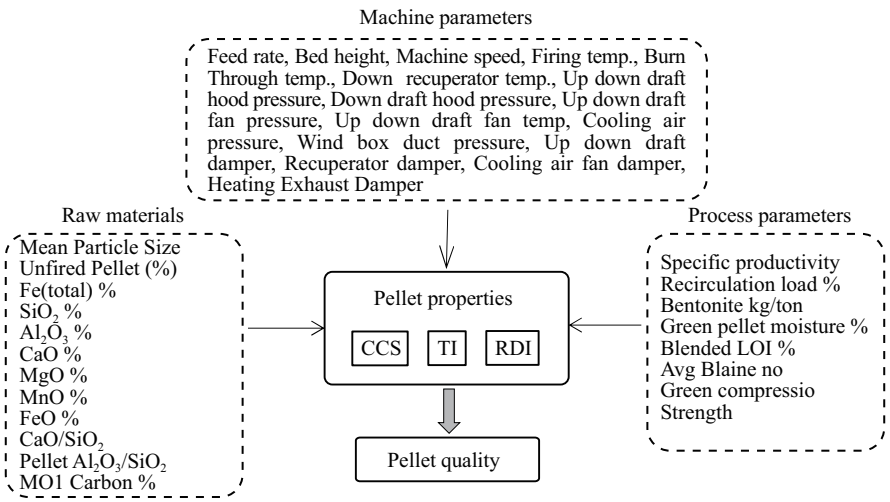

Figure 2. Parameters affecting pellet quality

After collecting all the recorded data and applying filters, by using feature selection and process-based inputs, the final list of variables that were used for building the model after feature selection is shown in Table 3.

Table 3. Ranges of model input parameters

Input parameter	Minimum	Maximum	Mean	Standard deviation
Mean particle size	9.46	12.41	11.35	0.48
Unfired pellet (%)	4.53	18	5.80	1.19
Fe(total) %	58.94	62.76	60.96	0.61
SiO_2 %	4.12	7.94	5.81	0.78
Al_2O_3 %	2.93	4.69	3.53	0.34
CaO %	1.58	2.33	1.80	0.11
MgO %	0.34	0.93	0.47	0.14
MnO %	0.04	1.39	0.66	0.29
FeO %	0.45	1.99	0.67	0.22
CaO/SiO_2 ratio	0.2	0.47	0.32	0.05
Pellet Al_2O_3/SiO_2 ratio	0.46	0.91	0.62	0.09
MO1 carbon (%)	0.93	1.61	1.20	0.14
Feed rate (tph)	222	573	492.2	72.65
Bed height (mm)	337	553	506.5	37.21
Machine speed (m/min)	1.55	2.33	2.1	0.15
Firing temp. (ºC)	793	1,277	1,210	67.02
Burn through temp. (ºC)	217	437	321	49.11
Down recuperator temp. (ºC)	556	923	831.6	59.41
Up down draft hood pressure (mmwc)	-51	-32	-38	2.59
Down draft hood pressure (mmwc)	-15.45	14.5	-3.56	8.37
Up down draft fan pressure (mmwc)	105	226	188	26.43
Up down draft fan temp. (ºC)	41	267	193	28.56
Cooling air pressure (mmwc)	136	435	297	62.86
Hood above WB#22 (mmwc)	375	713	529	78.94
Wind box duct pressure (mmwc)	-562	-366	-386	23.35
Up down draft damper (mmwc)	4	99	80.3	13.19
Recuperator damper (mmwc)	47	136	97.4	7.96
Cooling air fan damper (mmwc)	36	101	62.8	8.58
Heating exhaust damper (mmwc)	47	87	73	4.79
Specific productivity	9.8	25.09	21.65	3.22
Recirculation load (%)	10.82	37.36	21.13	6.10
Bentonite (Kg/ton)	2.87	7.33	6.31	0.83
Green pellet moisture (%)	9	10.6	9.57	0.25
Blended LOI (%)	3.07	5.21	3.89	0.45
Average blaine number	1,907	3,245	2,595.66	261.94
Green compression strength	1.03	1.6	1.37	0.10

From Table 3, the upper and lower values of different parameters signify the range of the model input variables. This highlights the dynamic nature of the process, wherein the input raw material composition and process parameters change on a routine basis. Many of these input variables bear complex non-linear relationships with the pellet properties and hence, are difficult to model using simpler machine learning techniques like linear regression or simple decision trees, etc. New ensemble learning techniques are quite powerful in capturing such underlying trends in the datasets and provide accurate predictions [4], [16], [18], [19], [21], hence are more suitable to be used while modeling a complex process like iron-ore agglomeration.

The model development approach is shown in Figure 3. After model development, the predicted values were stored back into the central database using Python libraries.

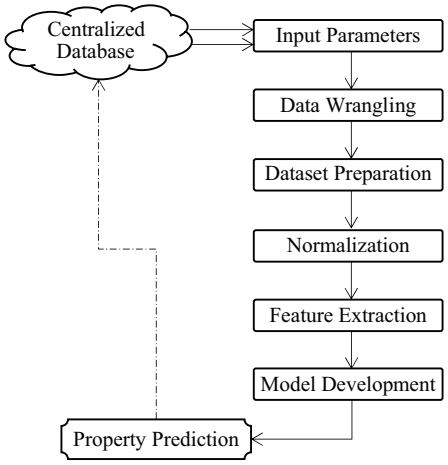

Figure 3. Prediction model development approach

2.3 *Performance evaluation of model*

The performance criteria of a regression-based data analytics model can be interpreted in several ways, one of which includes a coefficient of determination (R^2) as listed in equation (11).

$$R = \frac{\sum_{i=1}^{n}\left(C_{pi(act)} - C'_{p(act)}\right)\left(C_{pi(pred)} - C'_{p(pred)}\right)}{\sqrt{\sum_{i=1}^{n}\left(C_{pi(act)} - C'_{p(act)}\right)^2 \sum_{i=1}^{n}\left(C_{pi(pred)} - C'_{p(pred)}\right)^2}} \tag{11}$$

Where, $C_{p(pred)}$ and $C_{p(act)}$ represent the mean of the actual and predicted values.

The accuracies of the model were evaluated in two phases—one during the model-building phase to arrive at the right algorithm and the second after deployment in the production scenario for real-time checking of the performance. The coefficient of determination (R^2) for the model was calculated using a user-defined library in *R*-platform and is listed in Table 4.

Table 4. Coefficient of determination for predicted values

Parameters	Coefficient of determination (R^2)
CCS	0.96
RDI	0.95
TI	0.96

Inferring from the above table, considering a parameter, for example, CCS, for a pellet plant the R^2 is 0.96, which signifies that 96% of the predicted values match the actual values. The same was mapped to the other parameters. Certain unfired pellets were neglected during the process as these can provide erratic results.

3. Model validation and self learning approach

During the model-building process, a high R^2 value indicates good prediction accuracy within the randomized testing dataset, which represents a portion of the original input data. However, for a prediction model to be useful, it is mandatory to check for the prediction accuracy for fresh data points outside the input dataset. Tested values for pellet CCS, RDI, and TI for 100 days were collected for model validation. For prediction, a new innovative approach was followed. Initially, the prediction was carried out for the first dataset. The predicted value was compared with the plant-tested value. Thereafter, this predicted dataset with input values and the actual tested property value was appended into the initial dataset with 'n' values to get a new dataset with 'n + 1' values. Then, the model was retrained and rebuilt before the prediction of 2nd value set. This type of 'predictive and incremental dataset updating' approach was followed for the full 100 datasets. This approach allowed the model to be self-learning from successive data points after prediction and is shown in Figure 4.

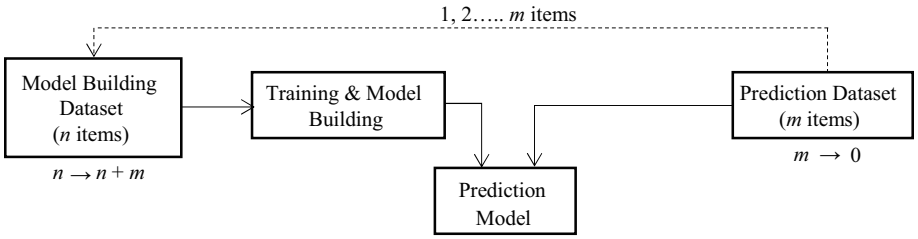

Figure 4. Model self-learning approach

Finally, the model-predicted values were compared with the actual tested values of pellet properties for the full 100 datasets as shown in Figure 5. Error threshold limits of ± 5%, denoted as acceptable error limits (Upper Error Limit (UEL), Lower Error Limit (LEL)), were selected as the criteria for classifying the predictions as accurate or inaccurate as per the operational requirements. The Upper and Lower Specification Limits (USL, LSL) were also plotted to gauge the process abnormalities. It was observed that the model was capable of tracking the process variations and providing fairly accurate predictions. The predictions for CCS and TI were found to be near accurate with an average percentage errors of 0.26 and 0.15, respectively, whereas the predictions for RDI had an average percentage error of 4.26. This translates into an average accuracy of > 99% for CCS and TI and > 95% for RDI. The relative accuracy levels for RDI predictions were slightly less because of the fewer data points available to train the model. RDI testing of pellets is comparatively a tedious and time-consuming process since it involves administration of heat along with CO, N_2 gas inside a furnace, followed by cooling and performing a tumbler test routine. This leads to low sampling frequency and fewer data points to train the model resulting in lower accuracy levels.

The new 'incremental updating dataset' approach for pellet property prediction helped to achieve high accuracy levels for the predicted variables. This is evident in Figure 5 for the RDI graph, where predictions initially showed deviations but subsequently the model learned from these process deviations and retrained itself, eventually becoming capable of delivering more accurate predictions. The actual physical testing of pellets for CCS and TI is comparatively faster and requires less manual effort, resulting in a sampling and testing frequency nearly 6-10 times higher than that for RDI, which provides sufficiently large datasets. Hence, the self-learning model was fairly accurate in predicting CCS and TI.

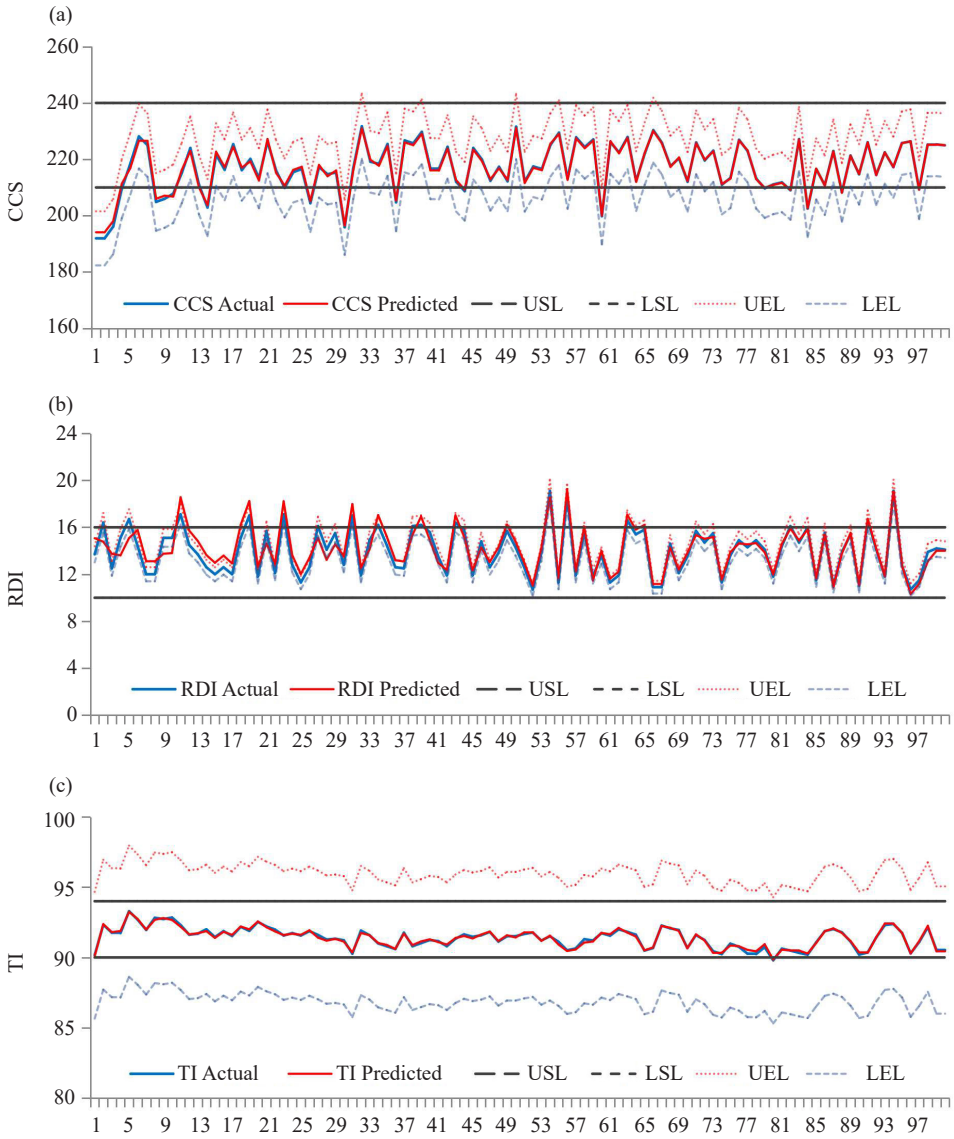

Figure 5. Validation of model predictions with actual plant data from pellet plant

4. Plant scale implementation

The prediction system was deployed in the plant scale operation through user input web-based prediction system and automated prediction & mail notification system.

4.1 *Web-based on-demand prediction system*

A web-based prediction system for delivering instantaneous on-demand predictions was deployed using Hyper Text Markup Language (HTML) as the Graphical User Interface (GUI) frontend and *R* as the backend for predictions. The interface has provisions for accepting manual input from the user in the form of Comma-Separated Values (CSV) file and giving predictions that can be downloaded or exported in different spreadsheet-type file formats as shown in Figure 6. This prediction system can be accessed anywhere using a web browser through a Uniform Resource Locator (URL) over the internet connection.

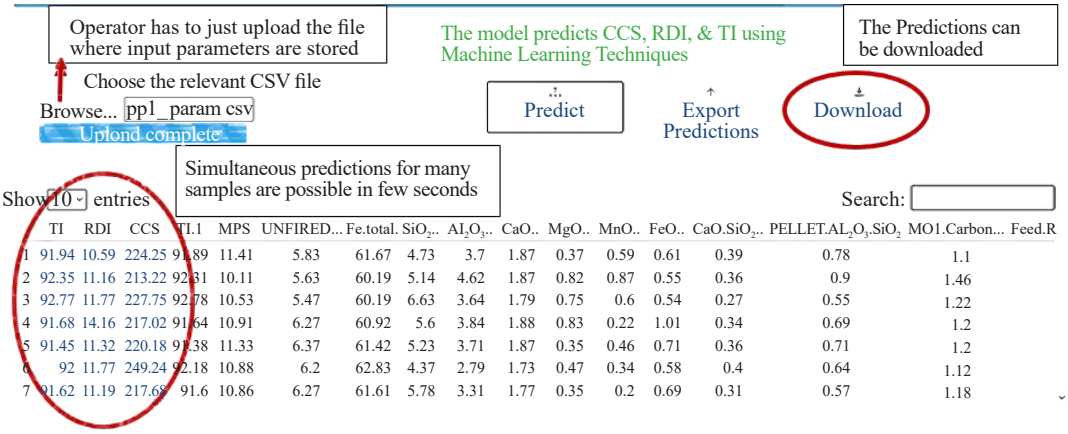

Figure 6. User entry-based on-demand prediction system interface

4.2 *Automated prediction and reporting system*

An automated prediction system was deployed, which can fetch the input parameters from the production server and deliver automated timed predictions as per the defined schedule as shown in Figure 7. The predicted values of CCS, RDI, and TI were updated on a web-based dashboard which gets auto-updated daily and can be accessed anywhere over the local intranet to track the pellet properties and monitor model performance. An automated mailing system was developed using Python to send daily emails for the predicted values and model performance to pellet plant operators, quality control team, and concerned authorities.

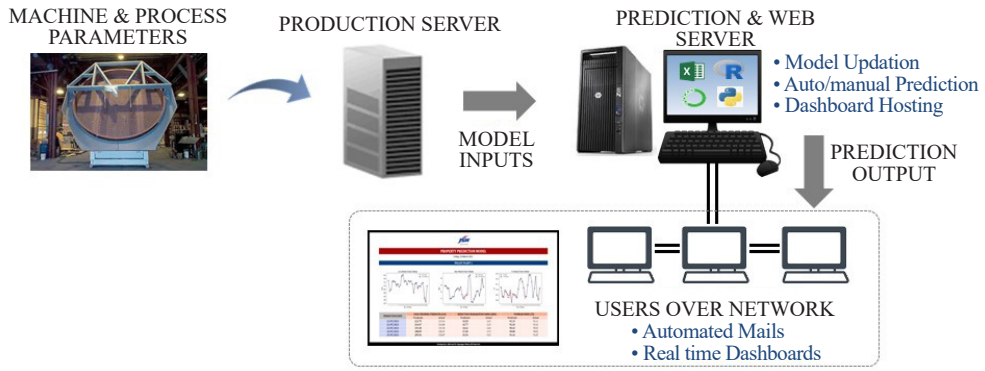

Figure 7. Automated prediction and reporting system

4.3 *Post-deployment benefits*

The model helped in getting accurate and timely predictions, which enabled the operator to manually intervene and control any process deviations on time. This improved the process control as is evident from the reduction in property variations and standard deviations for the pellet properties as shown in Figures 8 and 9.

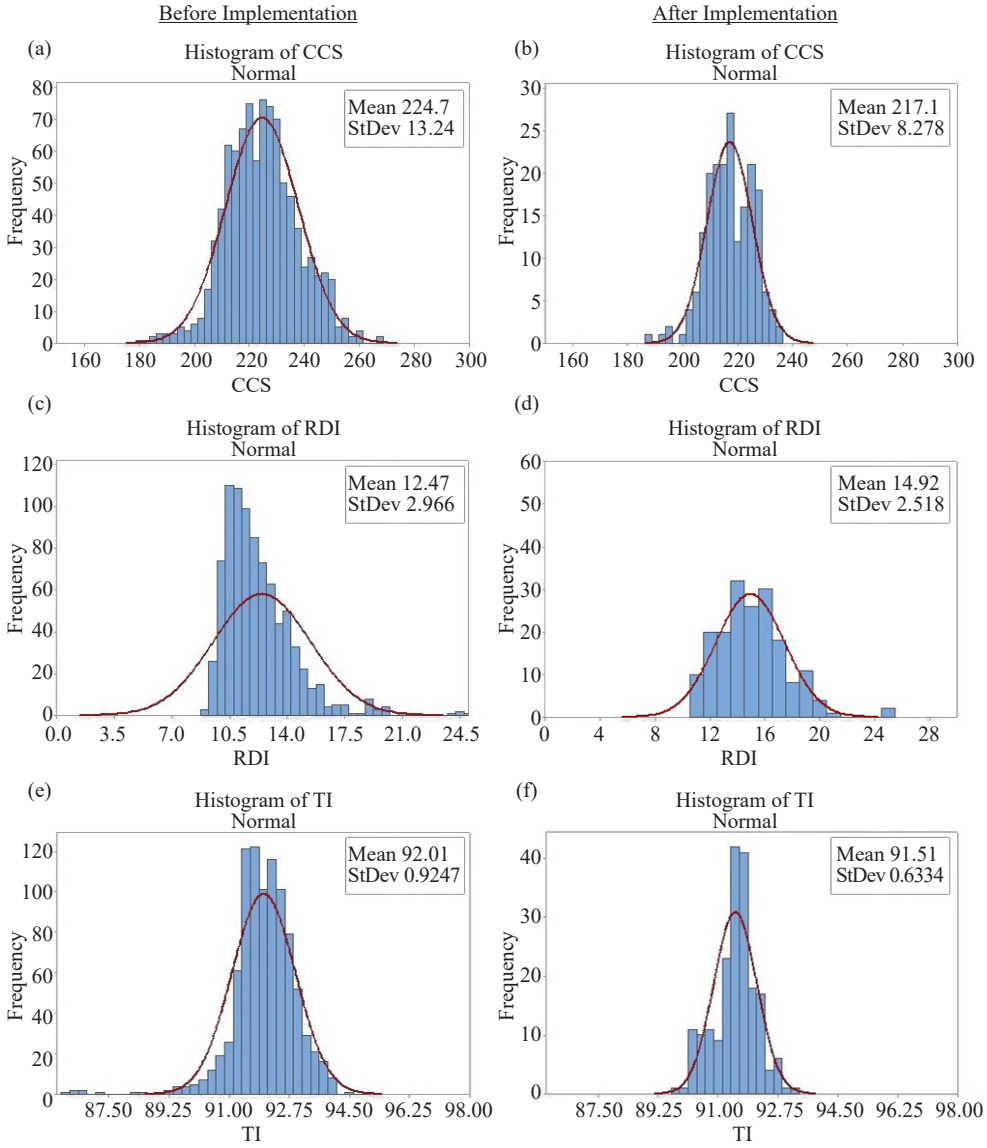

Figure 8. Distribution of CCS, RDI, and TI

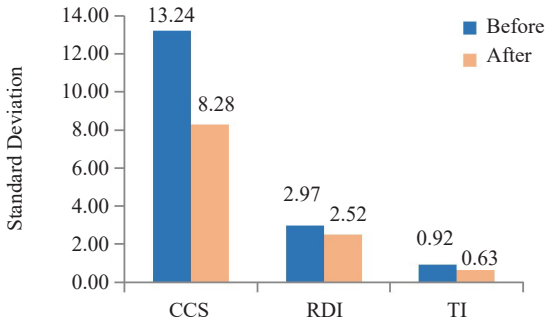

Figure 9. Standard deviation of CCS, RDI, and TI before and after model deployment

Initially, one sample/shift was taken for measuring the CCS and the TI, and one sample/day for testing the RDI of the pellets produced. After deployment of the model for three months, the sampling frequency was reduced by 50%

as shown in Figure 10a. With a reduction in sampling frequency, the associated cost of consumables, manpower, and machine maintenance was also reduced proportionately as reported in Figure 10b.

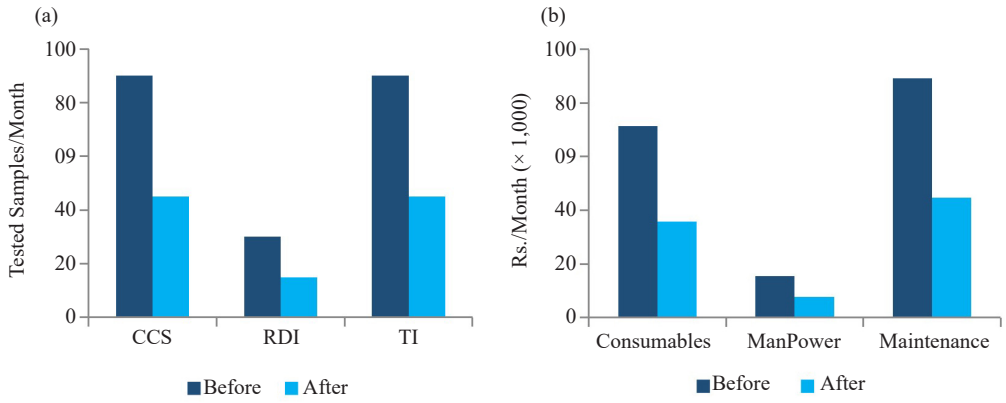

Figure 10. Reduction in (a) Sampling and (b) Sample testing costs before and after model deployment

5. Way forward

The present model has been developed and deployed to carry out predictive analysis in the pellet plant at JSW Steel Ltd. During routine plant operations, the charge mix input and the process variations are quite common, which often lead to deviations from the desired specifications' limit. In such abnormal scenarios, it is planned to incorporate a prescriptive feature in the current model, which can alert the operator in case of process deviations through suitable alarms and suggest possible corrective actions to minimize process deviations. This can be achieved by combining the presently developed prediction system with optimization algorithms to predict and suggest the most optimized operating conditions. This qualifies as a further enhancement feature of the present model and requires further research.

6. Conclusions

1. An ensemble machine learning-based Gaussian Gradient Boosting Algorithm was developed for the prediction of output quality parameters namely TI, RDI, and CCS for pelletizing units.

2. The final deployed model was developed using 36 input process variables. The dataset was split into 80 : 20 ratios for model training and testing.

3. The new 'incremental updating dataset' approach incorporated enabled the model to 'learn' the deviations and ongoing trends from the most recent datasets, thereby improving both prediction accuracy and operational control of the pelletizing process.

4. This self-learning prediction system was capable of predicting CCS and TI with > 99% accuracy and RDI with > 95% accuracy.

5. After deployment in the plant, the sampling frequency was reduced by 50%, leading to savings of consumables, man-hours, and maintenance cost of the testing equipment.

Acknowledgements

The authors would like to extend sincere thanks to the operations, quality control, and technical excellence teams at pellet plant 1 of JSW Steel Ltd. Vijayanagar Works for the help and support extended during this study. The authors would also like to thank the library at JSW Steel (JKC) for providing referenced material for the present work.

Conflict of interest

The authors declare no competing financial interest.

References

[1] R. P. Bhagat, *Agglomeration of Iron Ores.* Boca Raton, FL, USA: CRC Press, 2019.

[2] S. Dwarapudi, P. K. Gupta, and S. M. Rao, "Prediction of iron ore pellet strength using artificial neural network model," *ISIJ International*, vol. 47, no. 1, pp. 67-72, 2007.

[3] T. A. Runkler, *Data Analytics: Models and Algorithms for Intelligent Data Analysis*. Wiesbaden, Germany: Springer Vieweg, 2020.

[4] X. Wu and V. Kumar, *The Top Ten Algorithms in Data Mining.* Boca Raton, FL, USA: Chapman and Hall/CRC, 2009.

[5] W. Yukun, W. Lin, and L. Yunqi, "Iron ore pellets compressive strength prediction model based on KPCA-RBF," In Proceedings of the Chinese Control and Decision Conference (CCDC), Yinchuan, China, 2016, pp. 3080-3083.

[6] T. Umadevi, D. K. Naik, R. Sah, A. Brahmacharyulu, K. Marutiram, and P. C. Mahapatra, "Studies on parameters affecting sinter strength and prediction through artificial neural network model," *Mineral Processing and Extractive Metallurgy*, vol. 125, no. 1, pp. 32-38, 2016.

[7] L. Song, L. Qing, L. Xiaojie, and S. Yanqin, "Synthetically predicting the quality index of sinter using machine learning model," *Ironmaking & Steelmaking*, vol. 47, no. 7, pp. 828-836, 2019.

[8] F. Patisson, J. P. Bellot, D. Ablitzer, E. Marliere, C. Dulcy, and J. M. Steiler, "Mathematical modeling of iron ore sintering process," *Ironmaking & Steelmaking*, vol. 18, no. 2, pp. 89-95, 1991.

[9] W. Yang, R. Changkook, S. Choi, E. Choi, D. Lee, and W. Huh, "Modeling of combustion and heat transfer in an iron ore sintering bed with considerations of multiple solid phases," *ISIJ International*, vol. 44, no. 3, pp. 492-499, 2004.

[10] K. Kucukada, J. Thibault, D. Hodouin, G. Paquet, and S. Caron, "Modeling of a pilot scale iron ore pellet induration furnace," *Canadian Metallurgical Quarterly*, vol. 33, no. 1, pp. 1-12, 1994.

[11] V. Kumar, S. D. S. S. Sairam, S. Kumar, A. Singh, D. Nayak, R. Sah, and P. C. Mahapatra, "Prediction of iron ore sinter properties using statistical technique," *Transactions of the Indian Institute of Metals*, vol. 70, no. 6, pp. 1661-1670, 2016.

[12] Z. P. Li, X. H. Fan, G. Chen, G. M. Yang, and Y. Sun, "Optimization of iron ore sintering process based on ELM model and multi-criteria evaluation," *Neural Computing and Applications*, vol. 28, no. 8, pp. 2247-2253, 2016.

[13] S. Caron, G. Ouellet, D. Roy, and G. Paquet. "Simboul: A simulator of the iron ore pellet induration process," In Proceedings of the International Symposium on Control and Optimization in Minerals, Metals and Materials Processing, Quebec City, Canada, 1999, pp. 235-238.

[14] C. Sheppard, *Tree-Based Machine Learning Algorithms: Decision Trees, Random Forests, and Boosting*. North Charleston, SC, USA: Createspace Independent Publishing, 2017.

[15] B. A. Salami, S. M. Rahman, T. A. Oyehan, M. Maslehuddin, and S. U. Al Dulaijan, "Ensemble machine learning model for corrosion initiation time estimation of embedded steel reinforced self-compacting concrete," *Measurement*, vol. 165, pp. 108141, 2020.

[16] C. Zhang and Y. Ma, *Ensemble Machine Learning: Methods and Applications*. New York, NY, USA: Springer, 2012.

[17] N. Lopes and B. Ribeiro, "An efficient gradient-based learning algorithm applied to neural networks," *Neural, Parallel and Scientific Computations*, vol. 11, no. 3, pp. 253-272, 2003.

[18] M. Jana, "Exploring Machine Learning Models: A Comprehensive Comparison of Logistic Regression, Decision Trees, Svm, Random Forest, and XGBoost," medium.com, para. 2, Sep. 11, 2023. [Online]. Available: https://medium.com/@malli.learnings/exploring-machine-learning-models-a-comprehensive-comparison-of-logistic-regression-decision-38cc12287055. [Accessed Nov. 12, 2024].

[19] Jerome H. Friedman, "Greedy function approximation: A gradient boosting machine," *The Annals of Statistics*, vol. 29, no. 5, pp. 1189-1232, 2001.

[20] Z. Zhang, Y. Zhao, A. Canes, D. Steinberg, and O. Lyashevska, "Predictive analytics with gradient boosting in clinical medicine," *Annals of Translational Medicine*, vol. 7, no. 7, pp. 152-152, 2019.

[21] V. Kumar, N. Kedam, K. V. Sharma, D. J. Mehta, and T. Caloiero, "Advanced machine learning techniques to improve hydrological prediction: A comparative analysis of streamflow prediction models," *Water*, vol. 15, no. 14, pp. 2572, 2023.