S & M 4249

# Machine-learning-based Prediction of Initial Values from Two-dimensional Features in Areal Density Measurement

Sitthichai Pomthong and Rutchanee Gullayanon*

Department of Robotics and AI Engineering, School of Engineering, King Mongkut's Institute
of Technology Ladkrabang, 1, Soi Chalong Krung 1, Ladkrabang, Bangkok 10520, Thailand

(Received June 30, 2025; accepted August 20, 2025)

***Keywords:*** areal density, two-dimensional features, machine learning, hard disk drive, extreme gradient boosting, random forest regressor, light gradient boosting

Since calculating the density of a hard disk drive is crucial for determining its storage capacity, the measurement and computation processes require a considerable amount of time. The key variables affecting density are bits per inch (*BPI*) and tracks per inch (*TPI*), where these initial values are currently defined by a fixed constant derived from known passive data. In this research, we propose using machine learning from upstream data to appropriately set the initial density values for each head on the basis of its specific characteristics. Additionally, we explore enhancing efficiency by incorporating particular data into the model. The most challenging part of this research is predicting the final state after applying the share margin process, which alters the end capability of each read head depending on the characteristics of neighboring heads. To address this, we employed a learning method that leverages data from the neighboring heads to predict the final state. The results demonstrate a significant reduction in the workload of each head's read/write process, improving their performance, and effectively increasing the areal density of hard disk drives compared with traditional methods.

## 1. Introduction

The primary factor in hard disk drive (HDD) data recording is the storage density on the hard disk to achieve the desired hard disk capacity. Therefore, numerous research efforts have been made to increase the HDD density. For instance, Tarnopolsky conducted research on magnets that increased hard disk capacity.(1) Similarly, Granz *et al*. studied density indicators in magnetic data recording systems.(2) This included using machine learning to enhance the density efficiency in data storage on hard disk recording plates. For example, Qin *et al*.(3) and Liu and Paffenroth(4) used machine learning to increase storage density, leading to areas on the hard disk with the highest density. These research studies were aimed at maximizing the efficiency to achieve the highest storage density. Nevertheless, in other industry research efforts, the use machine learning to find initial values was attempted. For instance, Sun and Grotstollen presented novel methods for determining switching angles in selective-harmonics-eliminated

*Corresponding author: e-mail: rutchanee.gu@kmitl.ac.th

https://doi.org/10.18494/SAM5829

ISSN 0914-4935 © MYU K.K.
https://myukk.org/

pulse-width modulation inverters.[5] A set of nonlinear equations defines these switching angles. To solve these equations, the authors employed a predictive algorithm that calculates initial values as first-order approximations of the exact solutions. Furthermore, Fuller *et al*. explored the concept of digital twins, virtual replicas of physical processes used for simulation and optimization.[6] Digital twins can predict initial values and optimize factory operations by simulating different scenarios. Since the primary objective of this research is to optimize the process by applying machine learning techniques, this aligns with the approaches proposed by several other researchers. For instance, Wu *et al*. proposed an integrated machine learning framework to predict rolling force in H-shaped steel during hot-rolling production, leveraging real-time sensor data.[7] Their model combines multiple sensor inputs and process parameters to reduce reliance on direct force measurements, streamlining the production workflow. They achieved high prediction accuracy by applying various regression techniques and feature extraction methods while significantly decreasing the number of physical measurements required. This contributed to process optimization and measurement cost reduction, aligning with the growing smart manufacturing trend. In terms of process optimization in an HDD factory, Kanjanapruthipong *et al.* proposed a method to reduce the testing time in HDD manufacturing by predicting the second flying height (FH2) using a deep neural network optimized with particle swarm optimization, based on FH1 input data.[8] This is an excellent example of applying AI to streamline calibration processes.

In this research, we aim to enhance efficiency in calculating density after all components are assembled as hard disks. This is because the original process of calculating density starts from the constant values of bits per inch (*BPI*) and tracks per inch (*TPI*). Finally, with various measurement processes, the end density differs significantly from the starting point. We intend to use machine learning from upstream data, which can indicate the characteristics of the reader head well, in determining the initial values of both *BPI* and *TPI* densities suitable for that reader head. This method will help reduce the workload on the read head and increase efficiency so that the original process can squeeze out the maximum density. Finally, substituting the initial value measurement process with an efficient machine learning method will aid in reducing time and improving the quality of the results.

## 2. Experimental Procedure

In this section, we will explain the traditional areal density calibration process to illustrate the profiles of *BPI* and *TPI*, which initially rely on constant values throughout the process. The prediction process depends on previous data, but since this is an early-stage process, there is limited data correlated with the target profiles. This led to the consideration of upstream data, which includes measurements of head characteristics before they are assembled into the drive, as crucial input for predicting the initial values of *BPI* and *TPI*.

### 2.1 Areal density measurement

Figures 1 and 2 illustrate the average *BPI* and *TPI* values across 1500 drives, showing the variations at each step of the areal density calibration process. In the images, green indicates

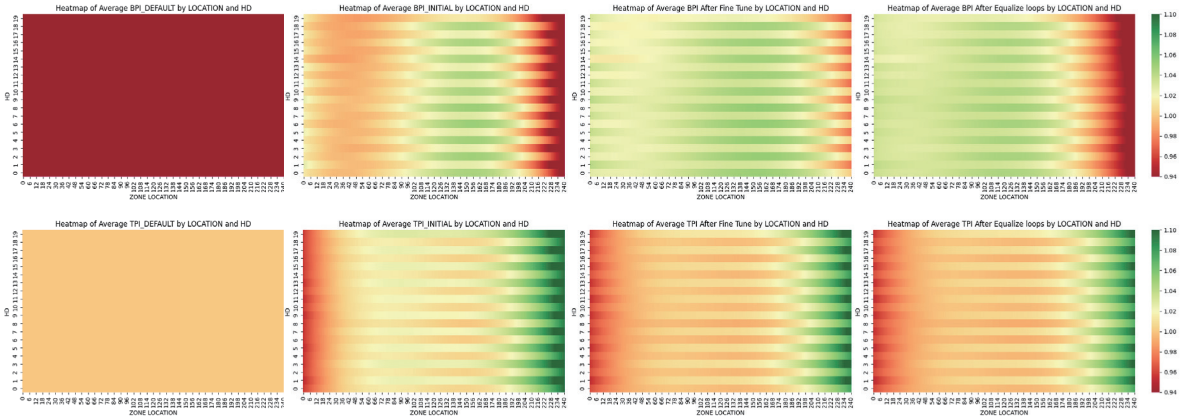
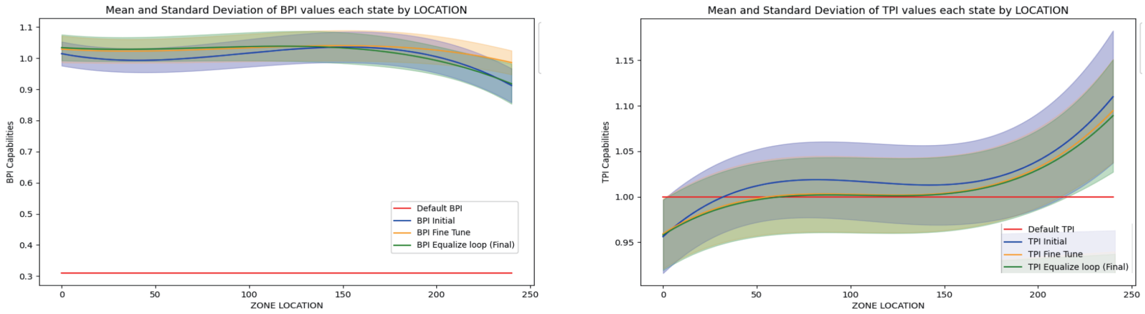


Fig. 1. (Color online) Variations in *BPI* (top row) and *TPI* (bottom row) capability density across each state during the areal density calibration process, represented by the average values from 1500 drives.



(a) (b)

Fig. 2. (Color online) Average values and deviations of (a) *BPI* and (b) *TPI* of each state: default (red), initial (blue), fine-tuned (yellow), and final (green).

areas of high capability, whereas red represents low-capability areas. The *x*-axis corresponds to the zone location, and the *y*-axis corresponds to the head position. The process at each step can be explained as follows: the first step begins with setting default values using constant values for both *BPI* and *TPI*, which *BPI* configured at a low capability to ensure that accurate density measurement can be conducted. Figure 2(a) illustrates the gap between the *BPI* default value and the final state. Meanwhile, *TPI* is set at a median value that allows the capacity to reach its target after the first step of default values, followed by a tuning process for measurement preparation. The next step involves measuring values to find suitable initial values for both *BPI* and *TPI*. Once suitable initial values for *BPI* and *TPI* are obtained, subsequent changes in these parameters necessitate retuning of the head to ensure optimal performance. Upon completion, the second measurement process begins. This involves stress measurement to find the best *BPI* and *TPI* values for all heads. Subsequently, a detailed tuning process is carried out using the margin sharing principle, which relies on the number of iterations in the repetition process. The final step involves checking whether the final *BPI* and *TPI* yield the desired hard disk capacity. In Fig. 1, the profile graphs of *BPI* and *TPI* illustrate the variance from the initial point to the final point of the process. In this research, we intend to predict the initial values derived from the first measurement to replace the traditional method of initial value measurement.

### 2.2 Hypothesis

Our research is based on the hypothesis that replacing fixed initial values of *BPI* and *TPI* with more appropriate values, either predicted or calculated, can yield improved outcomes. To test this hypothesis, an experiment was conducted in which HDDs with known areal density and finalized *BPI* and *TPI* values were reinitialized using these values instead of conventional fixed constants (Fig. 1). The objective was to assess the impact of appropriately determined initial parameters from the outset. Experimental results revealed that employing the finalized *BPI*/*TPI* values as initial settings reduced the bit error rate (*BER*), increased the areal density, and enhanced the track margin capability, as shown in Table 1.

The experimental results indicate that initializing both *BPI* and *TPI* values contributes to greater efficiency in determining the areal density of HDDs and reducing the time spent on equalizer loop processes. These findings have led to further research on predicting head-specific initial values for both *BPI* and *TPI* independently at the beginning of the process.

### 2.3 Key input features from upstream data

The initialization is an early process of the entire operation, and the exploration for additional data to be utilized in predictive modeling is initiated. Therefore, we initiated the synchronization of a head's preliminary data prior to the assembly process, as these data represent the head's specific characteristics (Fig. 3). The key parameters from upstream data consist of the width of the writer (wrt_wdt), the width of the reader (rd_wdt), and the ability to read data on the testing

Table 1
Read/write performance metrics for the initialization with traditional constants and hypothesized final *BPI*/*TPI* values.

| Measurement parametric | Initial *BPI* and *TPI* with fixed values | Initial *BPI* and *TPI* with final values | Improvement (%) |
|---|---|---|---|
| Bit error rate | 2.071 | 2.076 | 0.24 |
| Areal density | 1.031 | 1.033 | 0.21 |
| Track margin | 0.561 | 0.662 | 17.86 |
| Track density | 0.997 | 0.998 | 0.12 |
| Bit density | 1.035 | 1.036 | 0.08 |
| Loop equalizer | 3.750 | 3.000 | 20.00 |

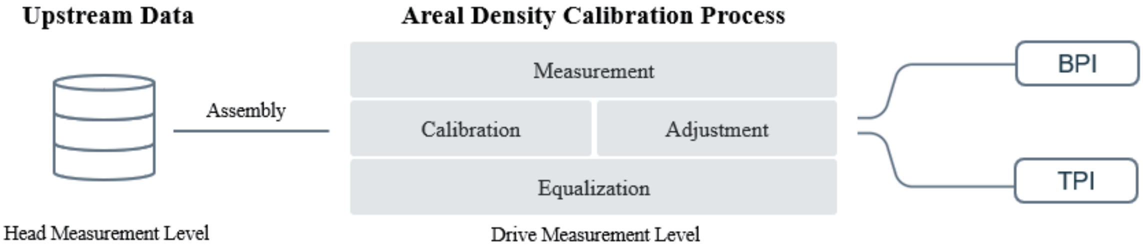


Fig. 3. Connecting head characteristics to areal density calibration process.

media represented by *BER* (rd_ber) correction and electrical current required to overwrite data on a testing media (ovw). This information has a direct relationship with *BPI* and *TPI*, as shown in Fig. 3.

### 2.4 Introducing coarse measurement

Upstream data on the specific characteristics of the head cannot fully capture the changes in behavior that occur after the head is assembled into a drive, such as interactions with the media or the installation of the head onto the drive. Conducting brief measurements after the drive assembly will provide critical data that reflects the real performance of the head in its new environment. This process takes significantly less time than the standard method, with measurements focused on reading the *BER* to roughly estimate the *BPI* and *TPI* of the drive. The correlation between the coarse measurement and the target *BPI* and *TPI* is robust, indicating a high predictive power compared with other key process input variables (KPIVs), as demonstrated in Fig. 4 in columns bpi_short and tpi_short.

### 2.5 Introducing a new predictive flow to supersede preliminary measurement

The results of the measurement flow shown in Fig. 1 reveal that the initial values of both *BPI* and *TPI* start with fixed constants. The head prerequisite to be tuned under these specified values ensures the head performs adequately during the initial measurement phase. After obtaining the appropriate *BPI* and *TPI* values, the subsequent measurement steps must be returned. If the initial values are predicted appropriately for each head, as shown in Fig. 5, this process can be shortened, and specific repetitive tasks can be minimized. When using the prediction of an initial value to replace a fixed constant, there are two possible target points for prediction, as shown in Fig. 6. The first is to predict the initial state, which we believe is highly

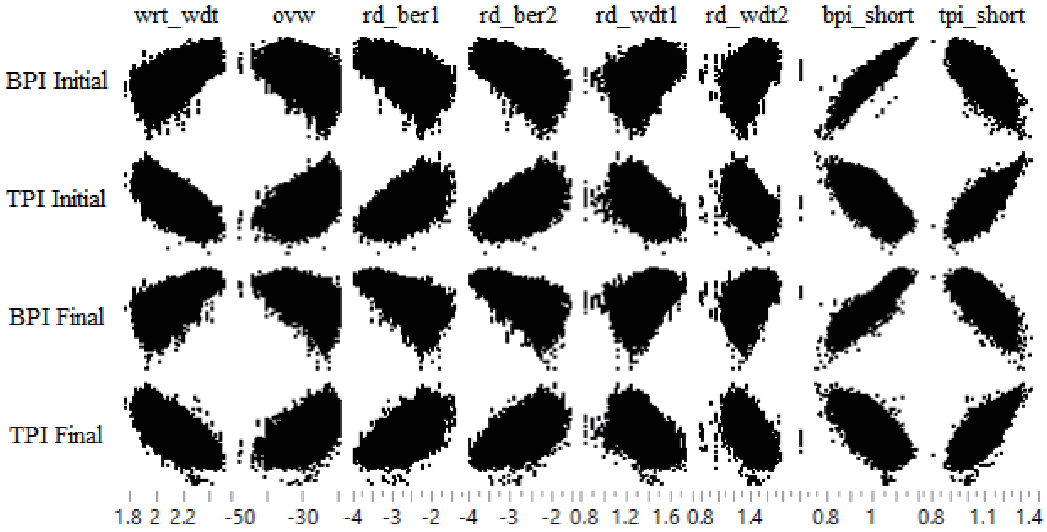


Fig. 4. Correlations of all KPIVs.

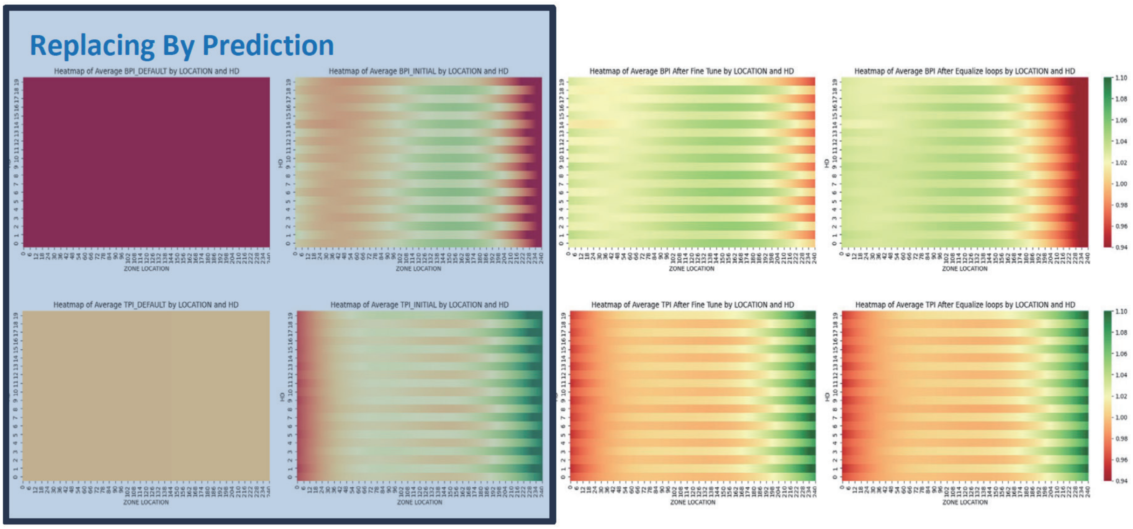


Fig. 5. (Color online) Position where prediction is used instead of initial measurement of *BPI*/*TPI*.

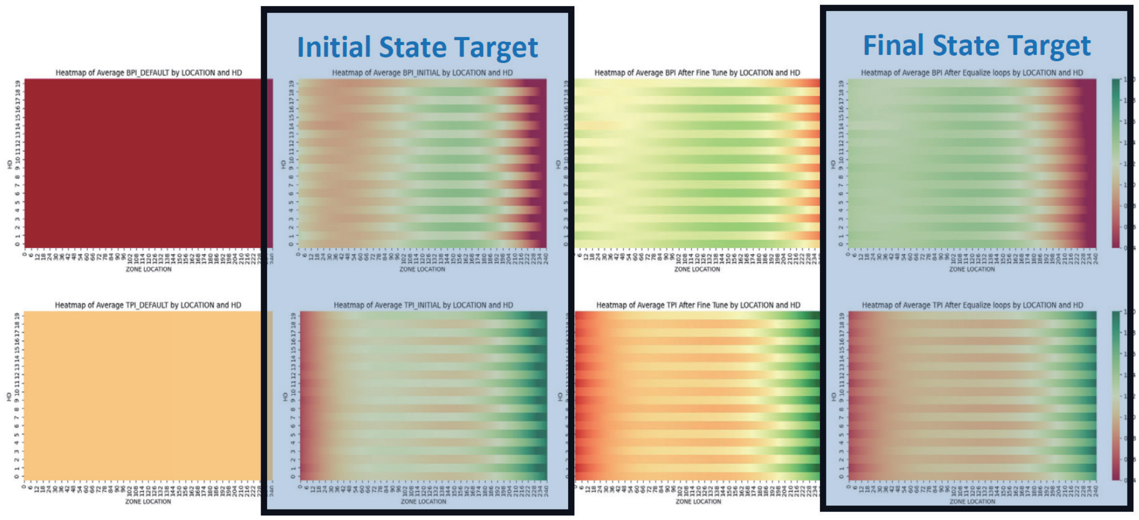


Fig. 6. (Color online) Prediction targets include the Initial State and Final State positions.

feasible because this stage is still the beginning of the measurement process and no margin-sharing procedure has yet occurred. However, the second target also deserves consideration, as it aims to reduce the number of measurement steps involved in the loop equalizer. If the predicted state closely matches the final hypothesized state, this approach is expected to decrease the number of intermediate measurement steps by more than 40%.

## 3. Methodology

In this section, we will discuss the new flow of *BPI* and *TPI* prediction, replacing traditional preliminary measurement and group of inputs used for prediction, the model employed for prediction, and the methods for measuring the hard disk's statistical and physical performance.

### 3.1 Essential input features for enhanced prediction

As seen in Table 2, the prediction process divides the input data for learning into three parts. The first part is the upstream data, which provides specific head characteristics. The second part is the short measurement data, which effectively describes the characteristics of the head and the media. The third part is the location zone and head position data. Considering Fig. 1, in the initial state, the density shows a pattern along the zones and heads as different positions on the media yield varying performances, which can improve the prediction accuracy.

### 3.2 Feature arrangement for predictive models

The initial experiment targeted the prediction of the initial state, as most KPIVs exhibit a direct relationship with the *BPI* and *TPI* values of each head. The share margin process has not yet occurred at this stage, making it suitable for a straightforward predictive approach. As a result, a direct prediction method requiring minimal complexity in data preparation was adopted for modeling the initial state. This approach is illustrated in Fig. 7, which presents a one-to-one

Table 2
Relationships among key feature groups associated with *BPI* and *TPI* targets at both the initial and final states.

| Group of features | Features | Correlation of *BPI* initial state | Correlation of *TPI* initial state | Correlation of *BPI* final state | Correlation of *TPI* final state |
|---|---|---|---|---|---|
| Upstream data | Writer width | 0.76 | −0.85 | 0.72 | −0.74 |
| | Reader1 width | 0.45 | −0.52 | 0.44 | −0.51 |
| | Reader2 width | 0.45 | −0.53 | 0.43 | −0.50 |
| | Overwrite | −0.58 | 0.61 | −0.57 | 0.57 |
| | Reader1 *BER* | −0.69 | 0.66 | −0.66 | 0.62 |
| | Reader2 *BER* | −0.70 | 0.67 | −0.67 | 0.64 |
| Coarse measure | *BPI* short | 0.96 | −0.66 | 0.90 | −0.67 |
| | *TPI* short | −0.75 | 0.89 | −0.72 | 0.89 |
| Drive physical | Head and zone | 0.18 | 0.15 | 0.17 | 0.15 |

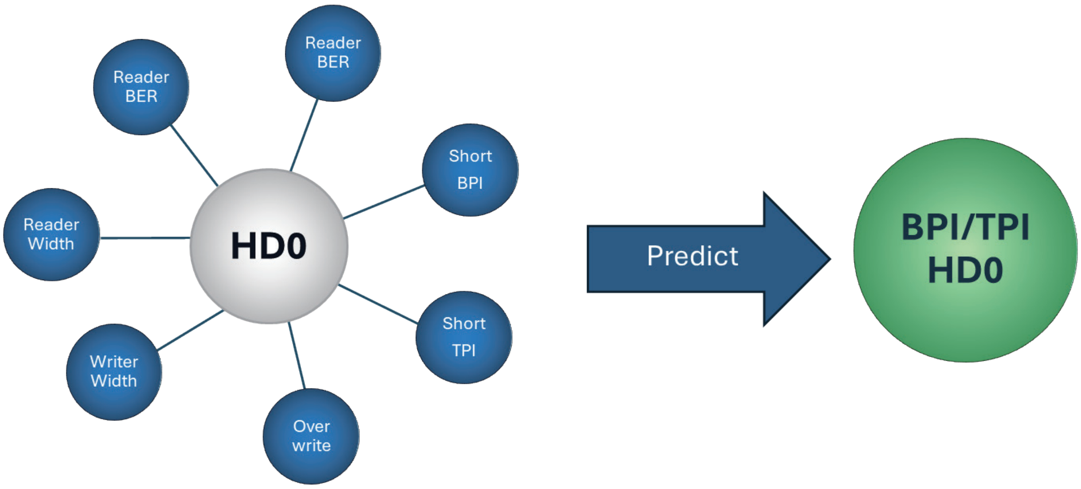


Fig. 7. (Color online) One-dimensional prediction by each head.

prediction example where the KPIVs of Head 0 are used to predict its initial *BPI* and *TPI* values. This simplified setup serves as a foundational case prior to extending the prediction to other heads, and is defined by Eq. (1).

$$y_i = \varphi(x_i + \beta) \tag{1}$$

Here, $y_i$ is the vector of the output *BPI* and *TPI* by head $i$ at the first and final states, $x_i$ is the vector of all features from Table 2 of each head *I*, $\beta$ is a constant setting based on the location head and zone, and $\varphi$ is the predictive model.

Despite other considerations, predicting the final state is the primary objective of this research. However, one-dimensional (head-to-head) prediction is insufficient for final state estimation owing to the margin compensation that occurs before reaching the final state. During the areal density calibration process, low-performing heads receive shared margins from high-performing head counterparts, thereby altering the head-specific outcomes. As a result, increasing the dimensionality of the feature set becomes essential. By incorporating neighboring head information into the features, the accuracy of final state prediction can be enhanced.

The two-dimensional prediction approach incorporates data from two dimensions, that is, the performance of the target head itself and the performance of other heads in the same drive, as illustrated in Fig. 8 and the accompanying equation:

$$y_i = \varphi(\lambda^T x_i + \beta)\,, \tag{2}$$

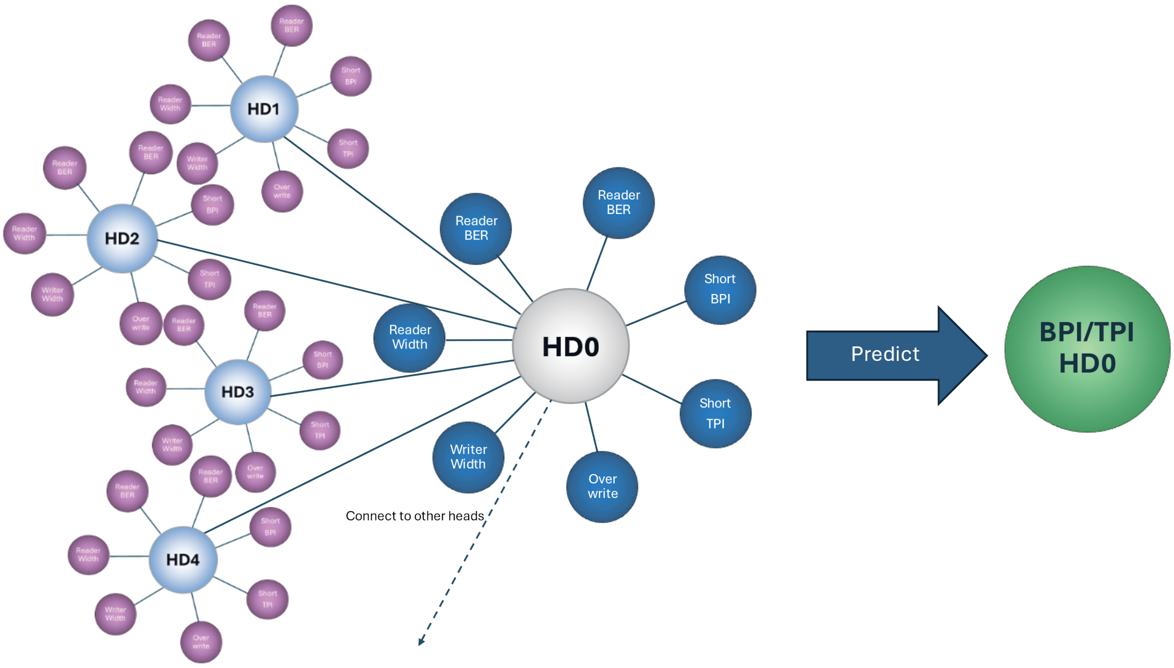


Fig. 8. (Color online) Two-dimensional prediction by each head.

where $y_i$ is the vector of the output *BPI* and *TPI* by head *i* at the first and final states, $\lambda$ is the vector of the head position, $x_i$ is the vector of all features from Table 2 of each head *I*, $\beta$ is the constant setting based on location head and zone, and $\varphi$ is the predictive model.

### 3.3 Predictive modeling techniques

In this study, we employed three ensemble-based regression algorithms consisting of the Extreme Gradient Boosting Regressor (XGBoost Regressor), Random Forest Regressor (RF Regressor), and Light Gradient Boosting Machine Regressor (LightGBM Regressor) to model and predict target variables based on structured input features.

(1) XGBoost Regressor(9,10) is a scalable and regularized gradient boosting framework that builds additive decision trees sequentially to minimize prediction error. XGBoost incorporates shrinkage, column subsampling, and sparsity-aware learning to enhance performance and prevent overfitting. In this study, XGBoost was implemented using the following default hyperparameters: learning_rate = 0.3, max_depth = 6, n_estimators = 100, subsample = 1.0, colsample_bytree = 1.0, and objective = 'reg:squarederror'. No manual tuning was applied, as the aim was to establish baseline performance.

(2) RF Regressor(11,12) is an ensemble learning method that constructs multiple decision trees using random subsets of both the data and feature space. The final prediction is computed as the average of individual tree outputs, which enhances robustness and reduces overfitting in regression tasks. This technique is particularly effective for modeling complex, nonlinear relationships within the data. The RF model was configured using default parameters: n_estimators = 100, max_depth = None, min_samples_split = 2, min_samples_leaf = 1, max_features = 'auto', and criterion = 'squared_error'. These settings were sufficient for initial benchmarking without further optimization.

(3) LightGBM Regressor(13,14) is a gradient boosting framework that constructs decision trees using a histogram-based algorithm and leaf-wise tree growth strategy. By discretizing continuous features into bins and prioritizing splits with the highest gain, LightGBM achieves faster training and lower memory usage than traditional boosting methods. LightGBM's ability to handle large data sets, support parallel learning, and natively process categorical features makes it particularly effective for high-dimensional regression tasks. Default hyperparameters were used for LightGBM, including learning_rate = 0.1, max_depth = −1, num_leaves = 31, n_estimators = 100, subsample = 1.0, colsample_bytree = 1.0, and objective = 'regression'. No manual tuning or feature engineering was applied at this stage.

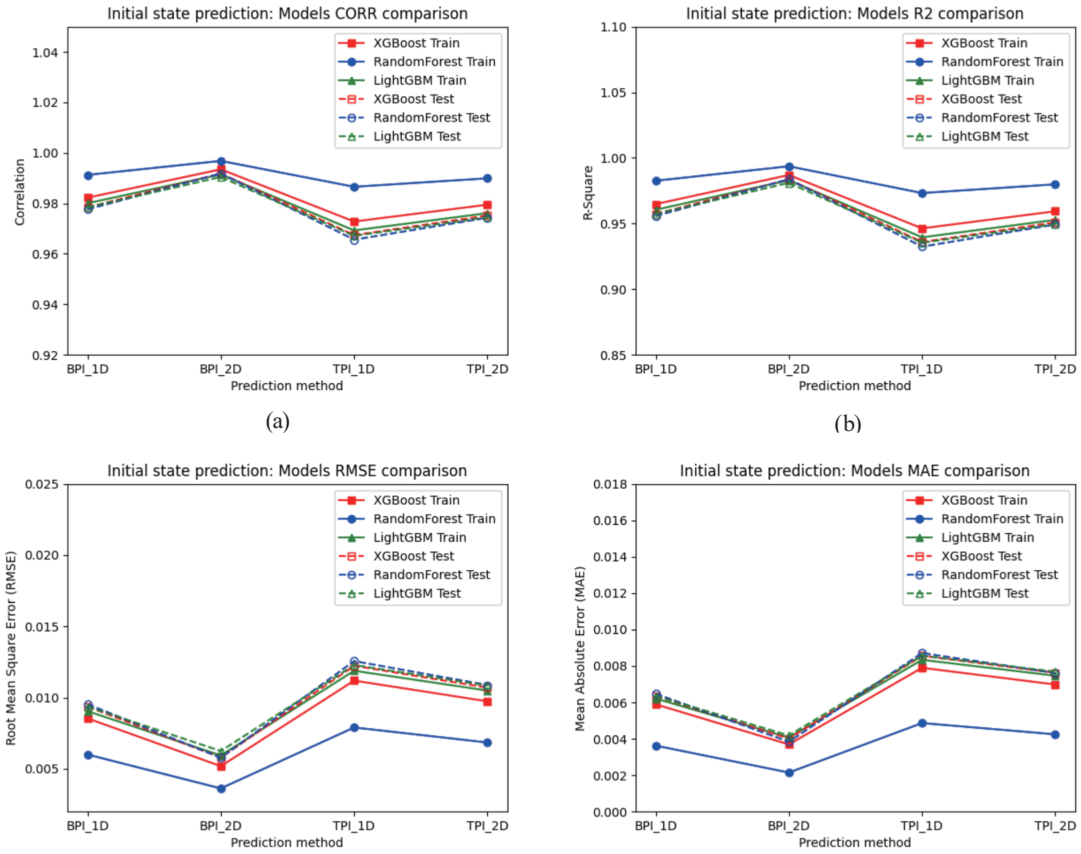
A dataset comprising 130000 data points was utilized for model development. The data were randomly partitioned into a training set (70%) and a testing set (30%). Default hyperparameters were employed across all models to maintain consistency and avoid bias introduced by model-specific tuning. This decision was made to ensure that the observed performance differences primarily reflect the impact of input dimensionality, specifically the comparison between one-dimensional and two-dimensional prediction strategies, rather than the effects of hyperparameter optimization. By controlling model complexity, the experiment was aimed at isolating and evaluating the predictive contribution of the input structure itself.

## 4. Experiment Results

The experiment results were categorized into two sections: the first involved predictions of the initial state, and the second is the final state predictions using one-dimensional and two-dimensional approaches. All projections were carried out using three regression models: XGBoost Regressor, RF Regressor, and LightGBM Regressor. Model performance was evaluated using standard metrics such as *R*-squared,[15] Pearson correlation coefficient, root mean square error (*RMSE*), and mean absolute error (*MAE*). Predictive effectiveness was assessed in a real process environment by comparing prediction outcomes between the initial and final state targets.

### 4.1 Initial state prediction

Experimental findings indicate that the one-dimensional prediction of the initial state (Fig. 9) yielded favorable results across all models. For the validation group, the correlation coefficient reached approximately 97.8% for *BPI* (Table 3) and 96.7% for *TPI* (Table 4), with *RMSE* and *MAE* both around 1%. Although the RF Regressor achieved the highest performance on the



(a) (b)

(c) (d)

Fig. 9. (Color online) (a) Correlation, (b) *R*-square, (c) *RMSE*, and (d) *MAE* of actual *BPI*/*TPI* of initial state relative to the predicted BPI/TPI on both training and validation sets of each model.

Table 3
Standard metrics of *BPI* prediction result for each model's 1D and 2D methods.

| Metrics | Target | Group | XGboost | | Random Forest | | LightGBM | |
|---|---|---|---|---|---|---|---|---|
| | | | BPI_1D | BPI_2D | BPI_1D | BPI_2D | BPI_1D | BPI_2D |
| Corr | Initial | Train | 0.982 | 0.993 | 0.991 | 0.997 | 0.980 | 0.991 |
| | | Test | 0.978 | 0.992 | 0.978 | 0.992 | 0.978 | 0.990 |
| | Final | Train | 0.953 | 0.972 | 0.977 | 0.985 | 0.945 | 0.962 |
| | | Test | 0.945 | 0.965 | 0.939 | 0.962 | 0.942 | 0.960 |
| $R^2$ | Initial | Train | 0.965 | 0.987 | 0.983 | 0.994 | 0.961 | 0.983 |
| | | Test | 0.957 | 0.984 | 0.956 | 0.984 | 0.957 | 0.981 |
| | Final | Train | 0.909 | 0.944 | 0.954 | 0.971 | 0.894 | 0.926 |
| | | Test | 0.893 | 0.930 | 0.882 | 0.926 | 0.888 | 0.921 |
| *RMSE* | Initial | Train | 0.009 | 0.005 | 0.006 | 0.004 | 0.009 | 0.006 |
| | | Test | 0.009 | 0.006 | 0.010 | 0.006 | 0.009 | 0.006 |
| | Final | Train | 0.013 | 0.010 | 0.010 | 0.008 | 0.014 | 0.012 |
| | | Test | 0.014 | 0.012 | 0.015 | 0.012 | 0.015 | 0.012 |
| *MAE* | Initial | Train | 0.006 | 0.004 | 0.004 | 0.002 | 0.006 | 0.004 |
| | | Test | 0.006 | 0.004 | 0.006 | 0.004 | 0.006 | 0.004 |
| | Final | Train | 0.010 | 0.008 | 0.006 | 0.005 | 0.011 | 0.009 |
| | | Test | 0.011 | 0.009 | 0.012 | 0.009 | 0.011 | 0.010 |

Table 4
Standard metrics of *TPI* prediction result for each model's 1D and 2D methods.

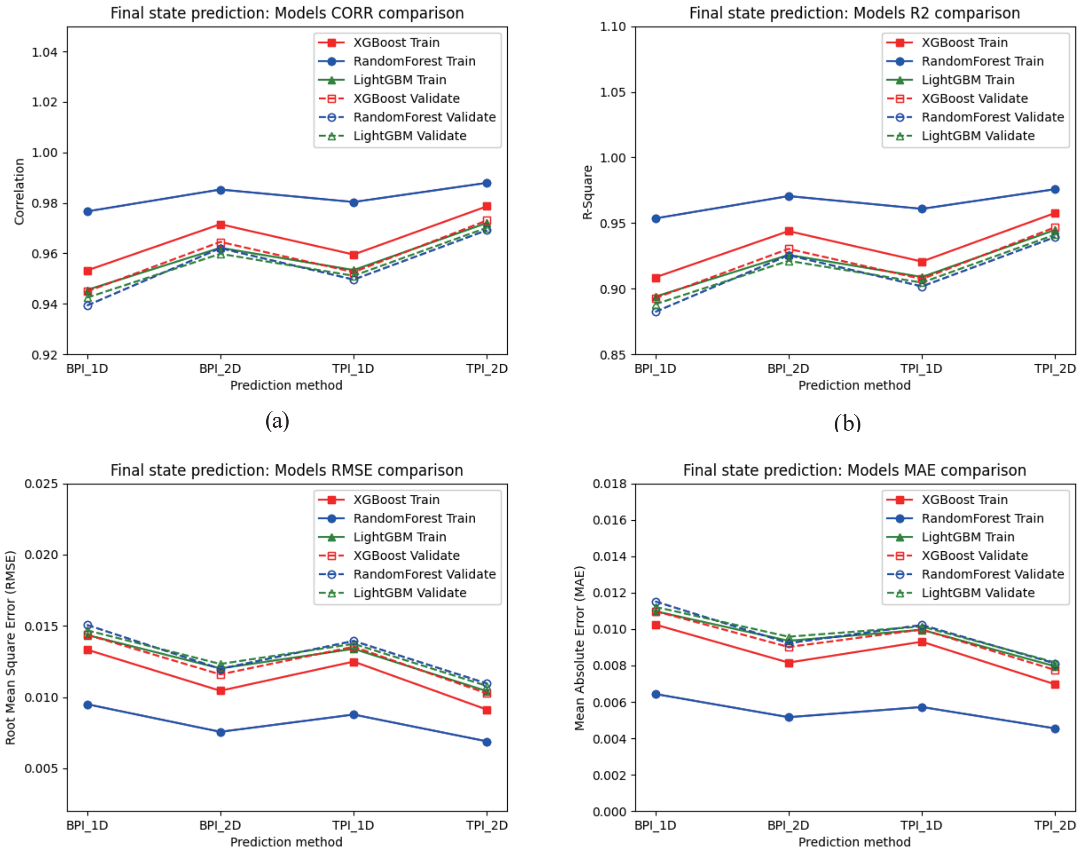
| Metric | Target | Group | XGboost | | Random Forest | | LightGBM | |
|---|---|---|---|---|---|---|---|---|
| | | | TPI_1D | TPI_2D | TPI_1D | TPI_2D | TPI_1D | TPI_2D |
| Corr | Initial | Train | 0.973 | 0.979 | 0.987 | 0.990 | 0.969 | 0.976 |
| | | Test | 0.967 | 0.975 | 0.966 | 0.974 | 0.967 | 0.974 |
| | Final | Train | 0.960 | 0.979 | 0.980 | 0.988 | 0.953 | 0.972 |
| | | Test | 0.953 | 0.973 | 0.950 | 0.969 | 0.951 | 0.970 |
| $R^2$ | Initial | Train | 0.946 | 0.959 | 0.973 | 0.980 | 0.940 | 0.953 |
| | | Test | 0.936 | 0.951 | 0.932 | 0.949 | 0.935 | 0.950 |
| | Final | Train | 0.921 | 0.958 | 0.961 | 0.976 | 0.909 | 0.945 |
| | | Test | 0.907 | 0.947 | 0.902 | 0.939 | 0.905 | 0.941 |
| *RMSE* | Initial | Train | 0.011 | 0.010 | 0.008 | 0.007 | 0.012 | 0.010 |
| | | Test | 0.012 | 0.011 | 0.013 | 0.011 | 0.012 | 0.011 |
| | Final | Train | 0.012 | 0.009 | 0.009 | 0.007 | 0.013 | 0.010 |
| | | Test | 0.014 | 0.010 | 0.014 | 0.011 | 0.014 | 0.011 |
| *MAE* | Initial | Train | 0.008 | 0.007 | 0.005 | 0.004 | 0.008 | 0.007 |
| | | Test | 0.009 | 0.008 | 0.009 | 0.008 | 0.009 | 0.008 |
| | Final | Train | 0.009 | 0.007 | 0.006 | 0.005 | 0.010 | 0.008 |
| | | Test | 0.010 | 0.008 | 0.010 | 0.008 | 0.010 | 0.008 |

training set, it exhibited signs of overfitting, as evidenced by a significant drop in correlation within the validation set. In contrast, the XGBoost Regressor demonstrated consistent and reliable performance across training and validation sets, making it the most robust model for the initial state prediction. Furthermore, incorporating two-dimensional KPIVs led to performance improvements in all the models, with the XGBoost Regressor remaining the most reasonable choice. This enhancement resulted in an approximately 1% increase in correlation and a reduction in *RMSE* and *MAE* to below 1%.

However, for the initial state, the one-dimensional prediction proved sufficient, as all regression models yielded comparable performance on the validation set. This suggests that any regression model can provide a reasonably accurate prediction.  Notably, XGBoost provided the most reasonable results for predicting the initial state, demonstrating superior stability and consistency in its estimations compared with other methods.

## 4.2 Final state prediction

The experimental results for final state prediction (Fig. 10) indicate that one-dimensional modeling generally performed worse than initial state prediction. This discrepancy is primarily attributed to the in-process margin sharing mechanism, which introduces variability among heads. Specifically, the correlation coefficient for *BPI* decreased from 97.8% in the initial state to 94.2% in the final state (Table 3), while *TPI* dropped from 96.7% to 95.1% (Table 4). Additionally, the *RMSE* increased by 0.5% for *BPI* (Table 3) and 0.2% for *TPI* (Table 4) under one-dimensional prediction. Among all the models, the XGBoost Regressor yielded the best performance on the training and validation sets.



(a)

(b)

(c)

(d)

Fig. 10. (Color online) (a) Correlation, (b) *R*-square, (c) *RMSE*, and (d) *MAE* of the actual *BPI*/*TPI* of the final state relative to the predicted *BPI*/*TPI* on both the training and validation sets of each model.

In contrast, the two-dimensional prediction approach significantly improved the model accuracy. Correlation coefficients for both *BPI* and *TPI* increased by approximately 2%, while *RMSE* values decreased by 0.3% for each metric. As illustrated in the *XY* plot in Fig. 11, the two-dimensional prediction using the XGBoost Regressor produced the most accurate and stable results for final state prediction.

### 4.3 Evaluation results of using predicted *BPI*/*TPI* as initial values in production

An evaluation was conducted using the XGBoost Regressor to predict *BPI* and *TPI* by a two-dimensional method for both the initial and final states. The experiment was performed on a set of 50 drives, and the results were compared against the hypothesis outlined in Sect. 2.2. A summary of the evaluation outcomes is presented in Table 5. The prediction of the initial state

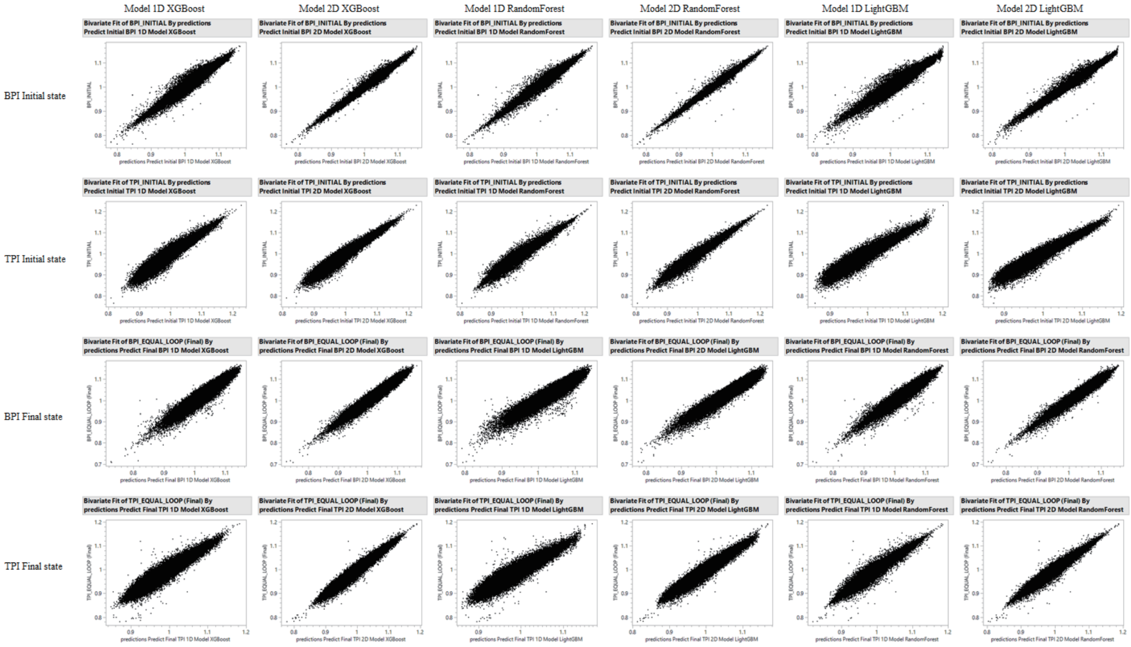


Fig. 11. The *XY* plots illustrate the relationship between the predicted and target values at both the initial and final states for each model.

Table 5
Measurement metrics across experimental and predictive scenarios by 2D techniques.

| Measurement metric | Initial *BPI*/*TPI* with fixed values (Traditional) | Initial *BPI*/*TPI* with final values (Hypothesis) | Initial *BPI*/*TPI* with predicted initial state (Evaluation) | Initial *BPI*/*TPI* with predicted final state (Evaluation) |
|---|---|---|---|---|
| Bit error rate | 2.072 | 2.076 | 2.070 | 2.073 |
| Areal density | 1.031 | 1.034 | 1.032 | 1.032 |
| Track margin | 0.561 | 0.662 | 0.573 | 0.635 |
| Track density | 0.997 | 0.998 | 0.996 | 0.998 |
| Bit density | 1.036 | 1.036 | 1.036 | 1.035 |
| Loop equalizer | 3.750 | 3.000 | 3.750 | 3.250 |

yielded results comparable to those of the conventional approach; thus, implementing this prediction method may primarily reduce measurement and computation time, with no significant improvement in performance. In contrast, the prediction of the final state demonstrated superior outcomes compared with the traditional method, most notably, an apparent increase in track margin and a reduction in equalizer loop iterations by approximately 13%. However, the overall results still did not surpass those anticipated in the hypothesis outlined in Sect. 2.2.

## 5. Conclusions

In this study, we demonstrated that leveraging upstream data and short-process information effectively enables the prediction of appropriate initial *BPI*/*TPI* values. The application of two-dimensional techniques substantially enhances prediction accuracy for initial and final states. Among the models evaluated, the XGBoost Regressor proved to be the most suitable for this task, notably reducing outliers in the test set with correlations of 99.2% and 97.5% for *BPI* and *TPI* at the initial state, and 96.5 and 97.3% for *BPI* and *TPI* at the final state, respectively. When evaluated in a production environment, initial state predictions using this approach reduced the time of the areal density calibration process by approximately 12%. In the case of final state prediction, the time was reduced by up to 35% owing to fewer equalizer loop iterations.

## Acknowledgments

This research was made possible through funding from the School of Engineering at King Mongkut's Institute of Technology Ladkrabang, focusing on industrial applications.

## References

1 G. J. Tarnopolsky: IEEE Trans. Magn. **40** (2004) 301. https://doi.org/10.1109/TMAG.2003.821169

2 S. D. Granz, T. Ngo, T. Rausch, R. Brockie, R. Wood, G. Bertero, and E. C. Gage: IEEE Trans. Magn. **53** (2017) 3100104. https://doi.org/10.1109/TMAG.2016.2607518

3 Y. Qin, P. Bellam, R. Galbraith, W. Hanson, N. Ravindran, I. Oboukhov, and J.-G. Zhu: IEEE Trans. Magn. **58** (2022) 3101407. https://doi.org/10.1109/TMAG.2021.3131139

4 Y. Liu and R. C. Paffenroth: Proc. 2021 IEEE 20th Int. Conf. Machine Learning and Applications (IEEE, 2021) 1660–1667. https://doi.org/10.1109/ICMLA52953.2021.00265

5 J. Sun and H. Grotstollen: Proc. 1992 Int. Conf. Industrial Electronics, Control, Instrumentation, and Automation (IEEE, 1992) 259–264. https://doi.org/10.1109/IECON.1992.254623

6 A. Fuller, Z. Fan, C. Day, and C. Barlow: IEEE Access **8** (2020) 108952. https://doi.org/10.1109/ACCESS.2020.2998358

7 L. Wu, D.Y. Zang, K. C. Wang, S. J. Qiu, and J. Z. Pan: Sens. Mater. **37** (2025) 2287. https://doi.org/10.18494/SAM5569

8 W. Kanjanapruthipong, P. Prasitmeeboon, and P. Konghuayrob: Sens. Mater. **36** (2024) 1377. https://doi.org/10.18494/SAM4825

9 T. Chen and C. Guestrin: Proc. 22nd ACM SIGKDD Int. Conf. Knowledge Discovery and Data Mining (2016) 785–794. https://doi.org/10.1145/2939672.2939785

10 A. Hafid, M. Ebrahim, M. Rahouti, and D. Oliveira: Proc. 2024 IEEE Int. Performance, Computing, and Communications Conf. (IEEE, 2024) 1–6. https://doi.org/10.1109/IPCCC59868.2024.10850357

11 T. K. Ho: IEEE Trans. Pattern Anal. Mach. Intell. **20** (1998) 832. https://doi.org/10.1109/34.709601

12 L. Breiman: Mach. Learn. **45** (2001) 5. https://doi.org/10.1023/A:1010933404324

13 F. Alzamzami, M. Hoda, and A. E. Saddik: IEEE Access **8** (2020) 101840. https://doi.org/10.1109/ACCESS.2020.2997330

14 G. Ke, Q. Meng, T. Finley, T. Wang, W. Chen, W. Ma, Q. Ye, and T. Y. Liu: Proc. 31st Int. Conf. Neural Information Processing System (2017) 3149–3157. https://dl.acm.org/doi/10.5555/3294996.3295074

15 J. Benesty, J. Chen, Y. Huang, and I. Cohen: Noise Reduction in Speech Processing (Springer, Berlin, Heidelberg, 2009) 1–4. https://doi.org/10.1007/978-3-642-00296-0_5