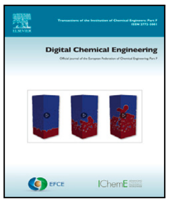
Contents lists available at ScienceDirect

Digital Chemical Engineering

journal homepage: www.elsevier.com/locate/dche



Original Article

Industrial data-driven machine learning soft sensing for optimal operation of etching tools

Feiyang Ou [a], Henrik Wang [a], Chao Zhang [c], Matthew Tom [a], Sthitie Bom [c], James F. Davis [a], Panagiotis D. Christofides [a,b,*]

[a] Department of Chemical and Biomolecular Engineering, University of California, Los Angeles, CA, 90095-1592, USA
[b] Department of Electrical and Computer Engineering, University of California, Los Angeles, CA 90095-1592, USA
[c] Seagate Technology, Minneapolis, MN 55435, USA

ARTICLE INFO

Keywords:
machine learning
Soft sensing
Optimal operation
Industrial tools
Oxide etching
Semiconductor manufacturing

ABSTRACT

Smart Manufacturing, or Industry 4.0, has gained significant attention in recent decades with the integration of Internet of Things (IoT) and Information Technologies (IT). As modern production methods continue to increase in complexity, there is a greater need to consider what variables can be physically measured. This advancement necessitates the use of physical sensors to comprehensively and directly gather measurable data on industrial processes; specifically, these sensors gather data that can be recontextualized into new process information. For example, artificial intelligence (AI) machine learning-based soft sensors can increase operational productivity and machine tool performance while still ensuring that critical product specifications are met. One industry that has a high volume of labor-intensive, time-consuming, and expensive processes is the semiconductor industry. AI machine learning methods can meet these challenges by taking in operational data and extracting process-specific information needed to meet the high product specifications of the industry. However, a key challenge is the availability of high quality data that covers the full operating range, including the day-to-day variance. This paper examines the applicability of soft sensing methods to the operational data of five industrial etching machines. Data is collected from readily accessible and cost-effective physical sensors installed on the tools that manage and control the operating conditions of the tool. The operational data are then used in an intelligent data aggregation approach that increases the scope and robustness for soft sensors in general by creating larger training datasets comprised of high value data with greater operational ranges and process variation. The generalized soft sensor can then be fine-tuned and validated for a particular machine. In this paper, we test the effects of data aggregation for high performing Feedforward Neural Network (FNN) models that are constructed in two ways: first as a classifier to estimate product PASS/FAIL outcomes and second as a regressor to quantitatively estimate oxide thickness. For PASS/FAIL classification, a data aggregation method is developed to enhance model predictive performance with larger training datasets. A statistical analysis method involving point-biserial correlation and the Mean Absolute Error (MAE) difference score is introduced to select the optimal candidate datasets for aggregation, further improving the effectiveness of data aggregation. For large datasets with high quality data that enable model training for more complex tasks, regression models that predict the oxide thickness of the product are also developed. Two types of models with different loss functions are tested to compare the effects of the Mean Squared Error (MSE) and Mean Absolute Percentage Error (MAPE) loss functions on model performance. Both the classification and regression models can be applied in industrial settings as they provide additional information regarding the process outcome. Individually, these models can reduce the number of metrology steps in semiconductor factories, and when developed further, can empower the development of advanced process control strategies.

1. Introduction

In the current decade, there has been a world-wide surge in demand for electronic products. This has in turn dramatically increased the manufacturing demand for related commodities such as microelectronics, hard drives, and integrated circuits (Nguyen et al., 2024). The innovations of modern-day electronics and their rising demand is in part owed to the rising density of transistors in semiconductor chips, which

* Corresponding author at: Department of Chemical and Biomolecular Engineering, University of California, Los Angeles, CA, 90095-1592, USA.
E-mail address: pdc@seas.ucla.edu (P.D. Christofides).

https://doi.org/10.1016/j.dche.2024.100195
Received 1 October 2024; Received in revised form 11 October 2024; Accepted 11 October 2024
Available online 22 October 2024

improves the computing performance of these chips (Burg and Ausubel, 2021). This increased demand has resulted in recurring shortages of electronics and has hurt the global economy, which now depends on the manufacturing of electronic devices. Thus, there is a growing need to pursue innovation in the manufacturing sectors (Nguyen et al., 2023). Smart manufacturing and Industry 4.0 concepts were proposed in the mid-2000s for developing smart plants and factories that utilize network communications, Information Technology (IT), Internet of Things (IoT) and big data (Christofides et al., 2007; Fuchs, 2018). Industry 4.0 aims to achieve resilient manufacturing processes that are characterized by high efficiency, high conformance, and high fidelity.

Smart Manufacturing, or Industry 4.0, necessitates holistic and continuous sensing, monitoring, and automation of processes that produce data in the form of quantifiable parameters. For instance, numerous advanced sensors are employed in manufacturing processes to monitor and understand the process parameters, and at the same time, they provide measured feedback to controllers for process automation. These sensors directly measure physical properties to obtain process information for monitoring and product quality assessment. For example, a quartz crystal microbalance is commonly used in the semiconductor industry to measure film thickness, providing coverage data for etching and deposition processes (Songkhla and Nakamoto, 2021). However, the operation of the measurement equipment is labor-intensive, time-intensive, and costly. In some scenarios, the capital and resource expenditures outweigh the value of the final product, one example of which is the treatment of wastewater (Wang et al., 2022). However, the costs of these measurement steps can be mitigated through advanced process monitoring (Perera et al., 2023). As industrial processes become increasingly complex, the direct measurement of key process parameters, which are often key performance indicators (KPIs), becomes more challenging. One process monitoring method is soft sensing, which detects critical process parameters by leveraging the wide range and scope of operational data that is generated in modern manufacturing processes (O'Donovan et al., 2015). Soft sensors use data from existing physical sensors and prior knowledge to develop data-driven algorithms that predict specific physical quantities and product quality, offering a more efficient, high-measuring frequency, and less labor-intensive alternative to traditional sensing approaches. Thus, they are particularly effective when there is a need to capture complex physical phenomena that are not easily measured or modeled or when there is a need for embedded fidelity that comes from years of operational experience. However, even when vast amounts of data are generated, the sensor performance will suffer if that data does not adequately cover the range, scope, and function of the operation relative to the sensor objectives.

In the context of modern complex manufacturing processes and the vast amounts of data they generate, there is a growing body of research that applies machine learning methods to develop soft sensors for detecting process and product properties. Recent literature reviews on deep learning methods in soft sensing (Sun and Ge, 2021) emphasize the significance of neural network approaches, including Convolutional Neural Networks (CNN) (Coleman et al., 2022), Recurrent Neural Networks (RNN) (Lee et al., 2021), and a combination of RNN and Feedforward Neural Networks (Hong et al., 2023). Seagate Technology also reported using the novel transformer network to predict the PASS/FAIL of an industrial etching process with time series data as the input (Zhang et al., 2021). Deep learning with neural networks have advantages in capturing complex nonlinear correlations between input and output parameters, and when large amounts of training data are available, they often outperform traditional machine learning methods (Sarker, 2021).

While many works have investigated which models are best suited for which deep-learning tasks, the question of if and how data aggregation can be used to supplement modeling tasks with small data volumes remains unanswered. Thus, this work proposes a deep-learning-based soft sensor developed using industrial data for detecting both binary properties (PASS/FAIL) and numerical properties (oxide thickness) for several industrial etching tools from Seagate Technology with high-dimensional input parameters. To address the problem where there is not enough data to train a model with high performance on a single tool, this paper proposes a novel data aggregation method where datasets from other tools are combined with the dataset of a single tool to improve model performance on that specific tool. The data aggregation approach aims to improve the performance of the trained soft sensor model by properly (in a sense to be made clear below) combining datasets to increase the amount and variety of training data. Specifically, this work introduces a statistical method to optimize dataset selection during the aggregation process to improve aggregation efficiency.

This work is organized as follows: Section 2.1 provides an overview of the industry data used, Section 2.2 describes the preprocessing operations applied to the datasets, Section 2.3 and Section 2.4 describe the development of the soft sensor models, Section 3 demonstrates and evaluates the performance of the trained soft sensor models, and Section 4 summarizes the findings of this work.

2. Data processing and modeling

This section describes the collection, processing, and contextualization of data from five industrial plasma etching tools, which are used to train two models: a classification model and a regression model. Then, we cover a cross-process data aggregation procedure for improving the classification model and the various loss functions used in training the regression model.

In the semiconductor fabrication industry, all products begin as a raw silicon wafer substrate. These substrates follow a set of procedures called the process flow, which describes each process step that the wafer must undergo. Once the wafer has gone through the entire process flow, it is a completed product. As the fabrication process is very repetitive, each wafer will be processed on the same tool multiple times at different process steps. This paper examines a toolset of five electrically-induced plasma etching tools. This toolset consists of five physically identical chemical etching reactors that possess up to two chambers; the reactor is referred to as the "tool", and the chamber is referred to as a "module". Each module can run a variety of process steps. The process data is gathered on a per-module basis, which means that each datum is for a specific tool-module combination. For example, an entry from T1-PM1 means that the wafer was processed in module PM1 of tool T1. A detailed diagram explaining the overall manufacturing process is shown in Fig. 1.

2.1. Industrial data generation

The process data used in this work was collected from four tools: T2, T4, T5, and T7. Each tool has up to two modules: PM1 and PM2. Specifically, process data was collected from T7-PM1, T7-PM2, T2-PM2, T5-PM2, and T4-PM1. Each data entry comes from a single run, which is defined as a process step that starts at $t = 0$ and ends at a preset process time, t_{end}. During the process, physical sensors track 33 numerical features, and their average is recorded down alongside two process-specific discrete features and a time stamp. Thus, there is no time-series data. Once the wafer has finished processing, it is referred to as the product. The dataset used in this work spans from February 1st, 2018 to December 31st, 2022. The numerical features are categorized into three classes of variables which are pressure and gas flow, electromagnetic, and other equipment statuses and properties. The discrete features are the process name ID and substrate family ID, which are both alphanumeric text entries. These 35 pieces of information are inputs for both the classification and regression model.

The information that both models aim to predict is whether the process was successfully completed, and this information is gathered at a different tool. After the etch step, the product wafer is processed at

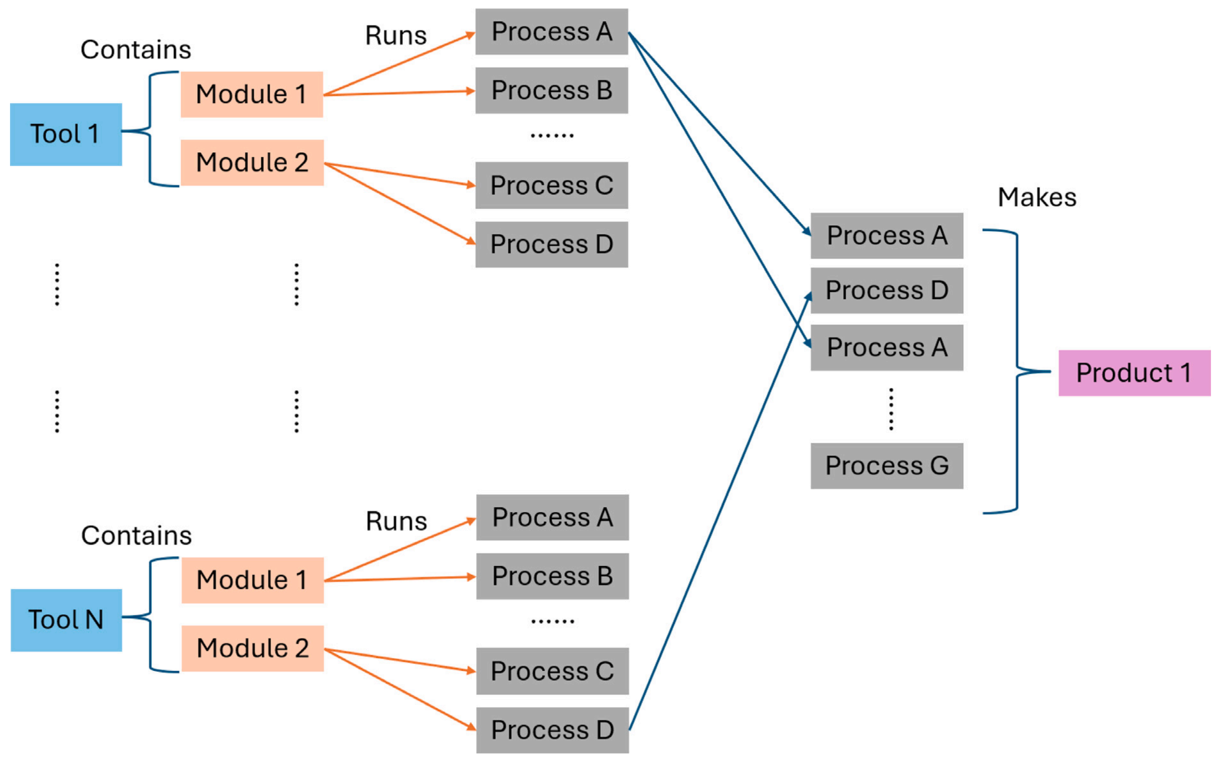


Fig. 1. The overall manufacturing system for an industrial etching equipment. Each tool is an etching reactor that has multiple modules and can run various processes. Wafers start as pure silicon substrates, and after a series of production processes, they become a finished product.

a metrology tool that measures how much substrate was etched away. As all of the processes examined in this paper are oxide etches, the metrology tool will measure the remaining oxide thickness. Depending on how much the measured oxide thickness deviates from the target oxide thickness, the run will be labeled as either a “PASS” or a “FAIL”. The classification model uses the binary PASS/FAIL measurement as its output, and the regression model uses the target oxide thickness as an additional input and the measured oxide thickness as its output.

2.2. Data preprocessing

Data preprocessing is a crucial step to effectively train any model. A properly preprocessed dataset allows the model to effectively learn the patterns of the data. For instance, Ranganathan (2021) highlights the necessity and importance of data preprocessing in several deep learning-based applications. Preprocessing steps often include removing or filling in invalid and abnormal data as is appropriate, encoding discrete variables, and normalizing features to avoid skewing caused by the absolute value of variables. These steps ensure that the model receives consistent and relevant data, which facilitate better learning and prediction ability; specifically for interpolating unseen data points within the applied training range. The input data preprocessing procedures in this work are demonstrated in Fig. 2, and the procedures for output data preprocessing are shown in Fig. 3.

From a practical manufacturing standpoint, physical sensors do not function properly at all times, and not all parameters can be measured in all processes. Thus, there are almost always missing physical measurements in real industrial data. The data analyzed in this work is not exempt from this phenomena. Some particular tool-module combinations are missing entire features, and other features are only collected within certain time ranges. To address these issues, any feature that has no measured value (which is recorded as N/A) is filled in with a numerical value of 0 to maintain consistency with the other data points. On the other hand, any runs (one row of data) that are missing either of the outputs, which are the PASS/FAIL criterion and the measured oxide thickness, are removed from the dataset. Without the true results, the input features are meaningless. By applying these two methods, all the invalid and abnormal data is pared from the datasets.

As with most machine learning models, which includes neural networks, all inputs must be numerical. Thus, it is necessary to encode any categorical or nonnumerical features if they are included in the training process. For this work, the label encoder from the scikit-learn package (Pedregosa et al., 2011) is used to encode both the substrate family ID and the process name ID from alphanumeric features into numerical features. To create a consistent and holistic encoder that is capable of handling all possible cases, the encoder is trained on data that comprise the concatenation of all datasets from all tool-modules. This step creates a complete map for the encoder, ensuring that each discrete feature is transformed into a unique number, which avoids conflicts during the training of the model. Additionally, for the binary PASS/FAIL output data, a 0/1 encoder is applied, which encodes PASS as 1 and FAIL as 0. These methods ensure that all the categorical features are transformed into numerical values so that the models can function effectively.

For neural networks, it is especially crucial to scale all numerical data to prevent the vanishing gradient and gradient explosion phenomena from occurring during the training process (Rehmer and Kroll, 2020). This is particularly important for input data features that vary significantly in absolute values. For instance, the dataset examined in this work has features in the range of both 10^{-4} and 10^{2}. A scaler normalizes the input features of the data, forcing each numerical feature to exist in a similar range and making the training process more stable and the optimizer task easier. For the classification task with a binary PASS/FAIL output, the numerical input features are scaled with a standard scaler, and the output features are already encoded as 1/0. For the regression task with numerical output features, separate MinMax scalers are applied for both the input and output data. The standard scaler is described in Eq. (1), and the MinMax scaler is described in Eq. (2). After the encoding process described earlier and the scaling processes described here are completed, all the

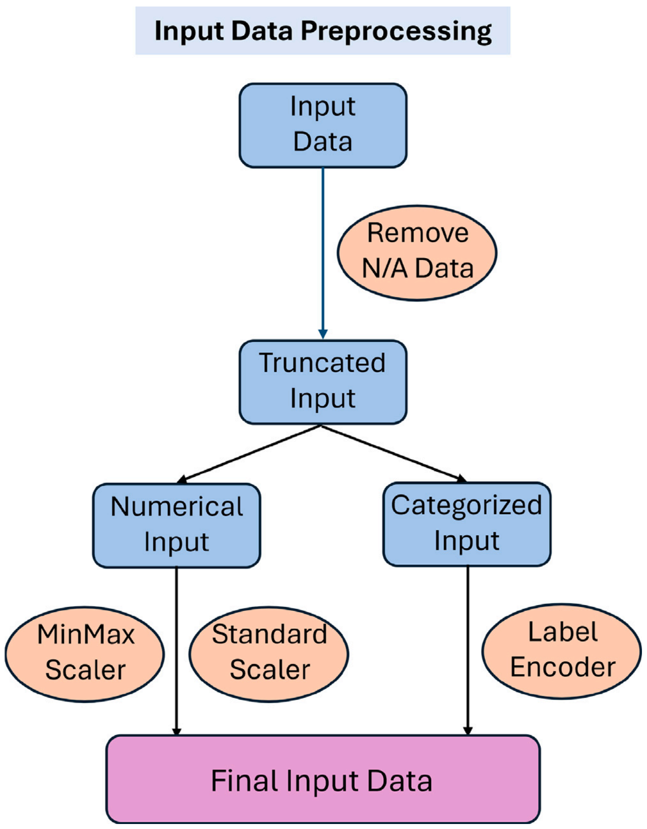


Fig. 2. Input data preprocessing **Step 1**: Eliminate or pad missing data. **Step 2**: Scale the numerical features, and encode the discrete features. **Step 3**: Combine numerical and discrete features into one complete input dataset.

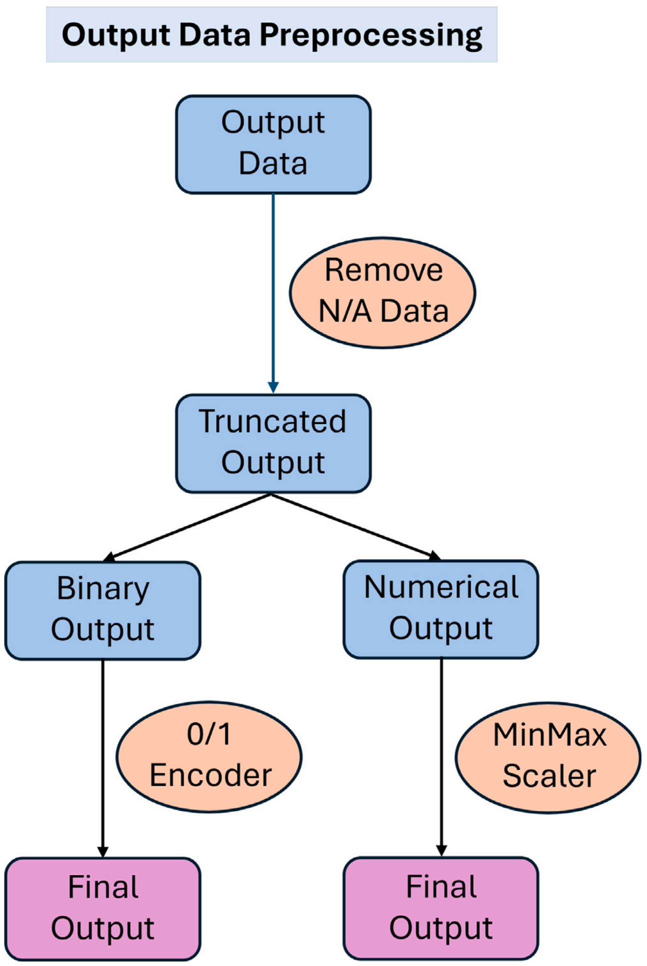


Fig. 3. Output data preprocessing **Step 1**: Eliminate missing data. **Step 2**: Encode binary output to FAIL(0) and PASS(1). **Step 2-2**: Scale numerical output with MinMax Scaler.

input features, including the scaled numerical features and encoded categorical features, are concatenated into a data vector that represents a single run.

$$Z = \frac{X - u}{s} \tag{1}$$

$$Z = \frac{X - Min(X)}{Max(X) - Min(X)} \tag{2}$$

where Z is the output vector of a scaled numerical feature, X is the input vector of the original feature, u is the average value of X, and s is the standard deviation of X. Note that the vectors used here consist of all the data for a particular feature (a data column). The standard scaler scales the original dataset around 0 for each feature in a similar range, while MinMax scaler scales the numerical data into a range of [0,1]. There is no fixed rule to determine the best scaling method. Rather, scaler selection is dependent on the optimal model performance, which will be more rigorously defined later on in Section 3. Specifically, both the standard scaler and the MinMax scaler are applied to the same dataset and used to train two independent models. Then, the scaler that yields the best performance is chosen for that task.

2.3. *Classification model*

The goal of a classification model is to accurately determine whether future, unknown wafers from a specific tool-module combination will pass or fail the following metrology step. The preprocessed input data vectors are the input variables to the model, and the output variable is a binary PASS/FAIL value. The dataset is first separated into two datasets by time: the modeling set and the test set. The modeling set comprises all the runs from February 1st, 2018 to December 31st, 2021, and it will be used to train the model. The test set comprises all the runs from January 1st, 2022 to December 31st, 2022, and it is used to evaluate model performance. These datasets are separated by time because it is necessary for the model to be generalizable across all times. From a practical point of view, the soft sensor model can only be considered successful if it can be effectively applied on unseen data and conditions from future manufacturing processes. To prevent overfitting, the modeling set is further separated into two more sets: the training set and the validation set. 80% of the modeling dataset is randomly chosen for the training set, and the remaining 20% is allocated to the validation set. The model is only trained and optimized on the training set, and the performance of each model is examined on both the training and validation sets to tune and optimize the generalization ability of the model. Stratified sampling is also used to ensure that the PASS/FAIL distribution is nearly identical between the training and validation datasets. With these specifications, when the model is trained on the training set and tested on the validation set, the candidate model is the model with the best performance on the validation set.

2.3.1. *Model training*

The Feedforward Neural Network (FNN) is applied to the classification model in this work. The general structure of the network is shown in Fig. 4.

In an FNN, each neuron in the hidden layers takes a weighted sum of inputs from the input layer or previous hidden layers, applies a non-linear activation function, and then passes the result to the next layer. The final output layer produces the binary PASS/FAIL classification

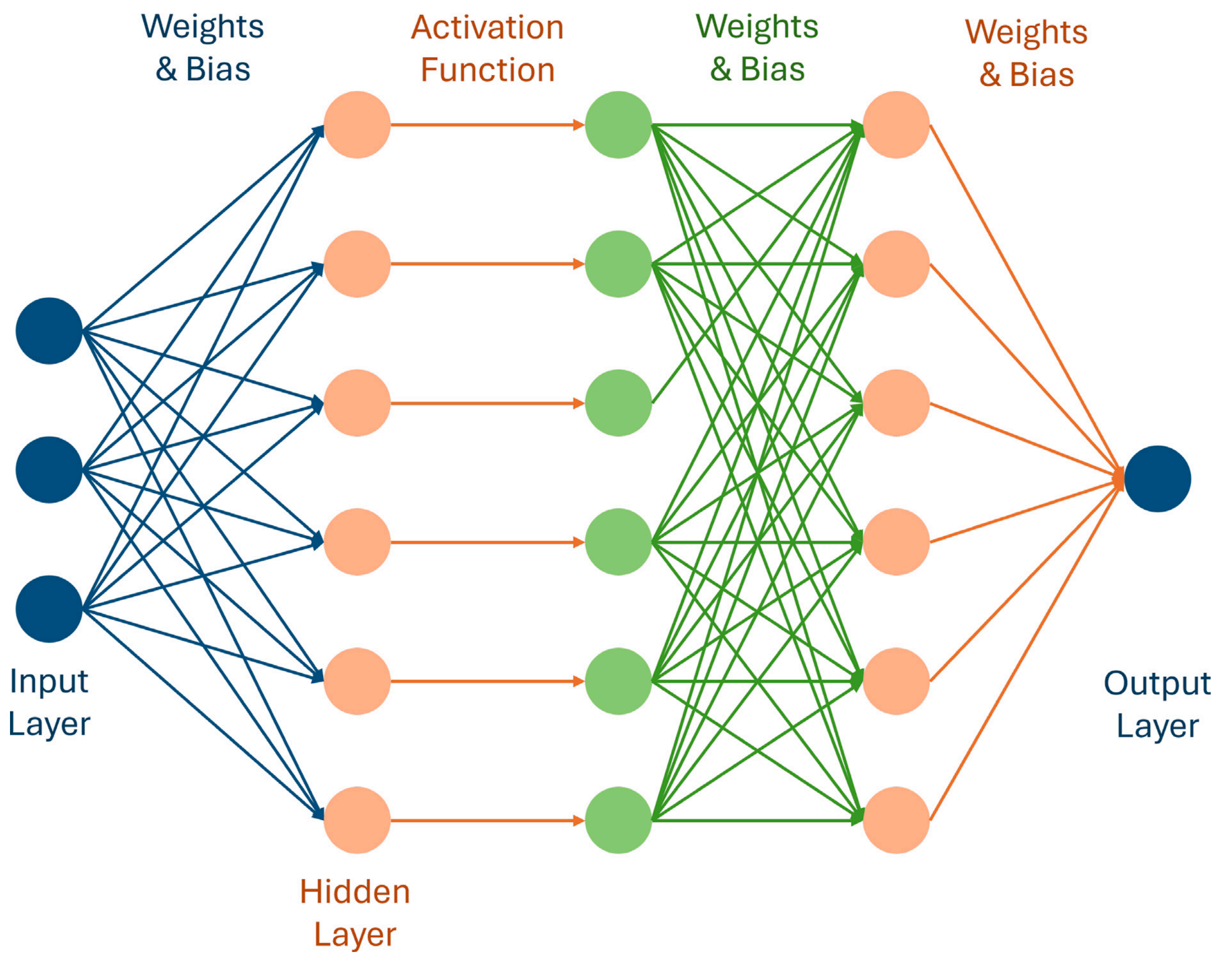


Fig. 4. The general structure of the Feedforward Neural Network (FNN) is shown here. The hidden layer neurons take a weighted sum from the input layer or the previous hidden layer, which is then activated by nonlinear functions. The output layer takes the weighed sum from the last hidden layer to output the result.

by using the sigmoid function to transfer the input value into the [0,1] range. The sigmoid function is also employed as the activation function in the hidden layers of classification models, due to its superior performance in classifier models (Zhang et al., 2021). The sigmoid function is described below in Eq. (3):

$$\sigma(x) = \frac{1}{1+e^{-x}} \tag{3}$$

The FNN has several tunable hyperparameters that need to be optimized to determine the best network structure. These hyperparameters include the number of hidden layers, the number of neurons in each hidden layer, the learning rate, and the L2 regularization coefficient. A grid search method is applied in this work to find the proper combination of hyperparameters to optimize the model performance on the validation set. The complete list of tuned hyperparameters and their candidate values are shown in Table 1. The models are trained on powerful graphical processing units (GPU), such as the Nvidia RTX A4000, Nvidia RTX 3060, and Nvidia RTX 4090, to facilitate the complete grid search of hyperparameters by exploiting the high computational capabilities of these GPUs. The selected hyperparameters are bolded in Table 1.

The training goal of a neural network is to minimize the loss function. For most binary classification tasks whose output values are processed by a sigmoid function, a cross-entropy loss is typically used, as shown in Eq. (4):

$$J = -\frac{1}{N}\sum_i [y_i \ln(y_i^p) + (1-y_i)\ln(1-y_i^p)] \tag{4}$$

Table 1
Classification FNN hyperparameters and tuning range.

Hyperparameters	Candidate values
Number of layers	[1,**2**,3]
Number of neurons	[32,**64**,128]
Learning rate	[0.005,**0.001**,0.0005,0.0001]
Dropout rate	[**0**,0.1,0.5]
L2 regularizer	[**0**,0.0001,0.0005,0.001]

where J is the loss value, y_i is the true value (0 or 1) of the data point i, y_i^p is the predicted value from the model within range [0,1], and N is the number of data points. However, it is important to note that the data is imbalanced because it comes from an industrial toolset that generally runs well with a low but still significant fail rate. All the fail rates for all five datasets are shown in Table 2, and most datasets exhibit a fail rate of approximately 2.5%. T5-PM2 is a notable exception with a significantly higher fail rate. This can be partially explained by its small dataset volume, which causes a few FAIL data points to have a large impact on the fail rate, but TM5-PM2's data clearly has more fails than the other tool-module combinations, marking it as different. Thus, by aggregating the TM5-PM2 dataset with other datasets and observing whether model performance increases or decreases, the effects of aggregating less related datasets together can be seen. For all other datasets, a model that always predicts "PASS" will not result in a high loss with the normal cross-entropy loss function, and that behavior will be favored during the training process.

However, such a model is not useful because it will have a 100% false positive rate. In other words, the model will not be able to detect

Table 2
Overall size and distribution of datasets.

Dataset	Total data points	PASS data points	FAIL data points	FAIL rate
T4-PM1	39 717	38 836	881	2.22%
T7-PM1	23 387	23 057	330	1.41%
T2-PM2	36 335	35 461	874	2.41%
T5-PM2	771	667	104	13.49%
T7-PM2	2722	2650	72	2.65%

any misprocessed wafers even though it has a very low training loss. To address this issue, a weighted cross entropy algorithm is proposed by multiplying the weight to data points by the output distribution (Zhang et al., 2021). The weighted loss function has the form:

$$J = -\frac{1}{N}\sum_i w_i[y_i \ln(y_i^p) + (1-y_i)\ln(1-y_i^p)] \tag{5}$$

where w_i is the weight and N is the total number of data points. The weights for binary classification are calculated as follows:

$$w_i = \begin{cases} \frac{N}{n_0}, & \text{if } y_i = 0 \\ \frac{N}{n_1}, & \text{if } y_i = 1 \end{cases} \tag{6}$$

where w_i is the ith weight, n_0 is the number of FAIL data points, and n_1 is the number of PASS data points. The weights applied in the loss function emphasizes minority instances to favor models with more comprehensive and balanced performances, thereby mitigating the potential bias that can arise from imbalanced datasets.

The Adam optimizer is applied throughout the entire model training process due to its excellent performance in handling various deep learning tasks (Kingma and Ba, 2017). Adam combines the advantages of other optimizers such as AdaGrad and RMSProp to enhance convergence speed and model performance. Model training is conducted over 1000 epochs, with the validation set loss continuously monitored after each epoch. The model with the lowest validation loss up to that epoch is then saved as the best model until the end of the training process. This approach ensures that the model with best generalization ability is retained, effectively reducing the risk of overfitting and improving the model's generalization capabilities. After the training process, the model is tested on future data of the test set to evaluate its performance.

To further reduce variability in the training and testing process, a cross-validation method is applied. This approach can greatly reduce model variance and improve the reliability of the model performance by averaging the performance of multiple models that are trained on different sections of the original data (King et al., 2021). First, the modeling dataset is randomly divided into five segments. Then, a segment is chosen as the validation set with the other four segments becoming the training set. By repeating this five times in total, once for each segment, five models are trained. Finally, each model is individually run on the test dataset, and the average of the five scores from the five models is used as the final test score for that dataset. This method enhances the robustness of the model evaluation by ensuring that the performance is consistent across different subsets of the data.

2.3.2. *Data aggregation*

Deep learning models, with their complex architectures and numerous parameters, can effectively capture intricate patterns within large datasets, leading to superior performance in most tasks (Sarker, 2021). As a result, deep learning networks have a distinct advantage over traditional machine learning methods, especially when working with training data at the industrial scale. However, industrial datasets often vary significantly in size between different tool-module combinations. For example, in this work, as shown in Table 2, T7-PM2 and T5-PM2 have dramatically smaller datasets with less than 3000 points compared to T4-PM1, T7-PM1, and T2-PM2, which have more than 20 000 points. This variation in dataset size implies that models trained using limited data from a particular tool-module combination for that specific combination will have considerably worse training results compared to tool-module combinations with larger datasets. Additionally, when the validation dataset is limited in size, it can lead to bias in the model, decreasing its ability to generalize because the validation set might no longer accurately represent the general data distribution. Moreover, even for the three tool-module combinations with large datasets, dataset aggregation will increase the variational and operational coverage of the training data and generally improve model performance. Consequently, maximizing high-quality training data volume is necessary to improve model performance. However, it is not enough to simply merge a large dataset and a small dataset, as that would merely result in a model that predicts the average behavior of the two aggregated datasets. Rather, multiple datasets from various tool-module combinations should be aggregated into a superset of varied data. When trained on this superset, the model performance improves as the larger volume of data better represents the overall process, increases the operational scope, and has more comprehensive information regarding process failures. A large amount of varied training data points that can only be obtained from aggregating data from multiple tool-module combinations refines the model, reduces overfitting, and enhances the robustness of the predictions.

To provide more data to train a model for each tool-module dataset, a data aggregation method is developed and tested that combines multiple datasets, significantly increasing the amount and variety of training data and improving model performance. During this process, the candidate datasets for aggregation must be selected carefully, making the analysis and selection process a critical data processing step. If the chosen tool-module dataset is very different from the current dataset, then the model will likely be misdirected by the new data and fail to retain the distribution of the original dataset. One method to guarantee optimal data aggregation is to exhaustively test all possible dataset combinations, but that is only feasible when there are only a few datasets. The number of possible dataset combinations increases exponentially with the number of datasets; n datasets have $2^n - 1$ unique combinations, making it impractical to exhaustively test all possible combinations when there are many tools and datasets.

To address this issue, this paper develops an indexing method to evaluate the similarities and differences between datasets. It was then used to select candidate datasets for aggregation, ones that are most similar to the dataset of interest. The indexing method is a point-biserial correlation analysis, which is a statistical method that measures the relationship between a continuous variable and a binary variable. In this work, the point-biserial correlation analysis is conducted on each numerical feature in the dataset with respect to the output binary variables, which provides information about the contribution of each feature to the binary outcome (PASS/FAIL). The analysis involves calculating the correlation coefficient as shown in Eq. (7).

$$r_{bis} = \frac{\bar{Y}_1 - \bar{Y}_0}{s_y}\sqrt{\frac{N_0 N_1}{N(N-1)}} \tag{7}$$

where r_{bis} is the correlation score between a specific feature and the output, $\bar{Y}_1$ is the average value of the feature for all PASS data points, $\bar{Y}_0$ is the average value of the feature for all FAIL data points, s_y is the standard deviation of the feature for all data points, N_0 is the number of FAIL data points, N_1 is the number of PASS data points, and N is the total number of data points. After the correlation coefficients are calculated for each feature, they are assembled into a characteristic vector. Each dataset has its own characteristic vector, and the difference score between any two datasets is calculated by finding the mean absolute error (MAE) between their characteristic vectors as shown in Eq. (8).

$$d = AVG(|\vec{c_1} - \vec{c_2}|) \tag{8}$$

where d is the difference score between the two datasets, AVG is an operation that takes the average value of all elements in a vector, $\vec{c_1}$ is the characteristic vector of first dataset, and $\vec{c_2}$ is the characteristic

Table 3
Regression FNN hyperparameters and tuning range.

Hyperparameters	Candidate values
Number of layers	[2,**3**]
First hidden layer neurons	[32,**6**,128]
Learning rate	[0.0001,**1E−5**,5E−6,2E−6]
Dropout rate	[**0**,0.1,0.25]
L2 regularizer	[**0**,0.0001,0.0005,0.001]

vector of second dataset. After the difference scores are calculated between the current analyzed dataset and all other datasets, the one with the smallest difference score is chosen as the candidate dataset for aggregation. The statistical analysis of the datasets themselves does not require any model training until the candidate datasets for aggregation are selected. This approach significantly reduces the number of models that need to be trained and tested, saving a substantial amount of time and computational resources.

2.4. Regression model

This work also explores machine-learning models that quantitatively predict the measured oxide thickness of processed wafers. This model, also called a regression model, is another FNN, though one with a different structure better suited for non-binary outputs. The preprocessed input data for the regression models are the same as that of the classification models, but with the addition of the target oxide thickness as an input feature. This feature is included as the target oxide thickness for each process must be known before the run starts. The model output is the measured oxide thickness, which spans a wide numerical range, from 0 to over 5000 Å, depending on the process and product.

Regression models generally require more data than classification models as the output of the latter is more complex. Thus, this work only focuses on training regression models with large datasets (Sarker, 2021). Specifically, T2-PM2, T4-PM1, and T7-PM1 are used to train regression models while T5-PM2 and T7-PM2 are not. The latter two datasets cannot support the training of a complex regression model because the limited data volume can induce problems such as severe overfitting and high variance results. Due to a lack of feasible datasets, data aggregation, which was explored for the classification task, is not conducted for the regression task because there are only three applicable datasets for regression. The available multi-dataset combinations are limited; specifically there are three two-set aggregations and one three-set aggregation. This small sample size means that any conclusions reached through the statistical analysis may be biased. Lastly, regression models do not require any data stratification like classification models because random selection and chronological selection can easily create artificial differences in fail rates between the training and validation sets due to the natural imbalance of the datasets. Because only datasets with substantial data volumes are selected, which effectively reduces the variance in the input features between different time segments, the training-validation split is organized solely by time. Specifically, the first 80% of the total training-validation dataset, sorted chronologically, is used for training, and the remaining 20% is reserved for validation. This approach aims to enhance model performance by selecting for the regression model that best predicts future data points.

2.4.1. Regression model training

The Feedforward Neural Network (FNN) for the oxide thickness regression task applies the ReLU (Rectified Linear Unit) activation function to provide nonlinearity for the model, which is described by the following equation:

$$\mathrm{ReLU}(x) = \max(0, x) \tag{9}$$

The ReLU function is chosen because of its promising performance on various training tasks, ability to avoid gradient vanishing problem, and increase in the speed of the training process (Xu et al., 2015; Waoo and Soni, 2021). In addition, due to how complex it is to train regression models and the need to save computational resources during the hyperparameter grid search, the neural network is designed such that each subsequent hidden layer has half the neurons of the previous layer. This approach helps manage model complexity, reduces the risk of overfitting, and ensures efficient feature extraction. Similar to the classification task case, the tunable hyperparameters and their range of interests are shown in Table 3, and the selected hyperparameters are bolded.

A small learning rate and many training epochs (100,000 in this work) are essential for training regression models due to the wide range of output values and the highly complex nonlinear correlations between the 37 input/output features. These conditions ensure that the training process reaches an optimum. Without the small learning rate, the model is sensitive to improper convergence, which causes the training process to jump around the optima or even diverge. By contrast, a small learning rate allows for more precise adjustments to the weights during backpropagation. The extensive number of training epochs ensures that the model has sufficient iterations to learn these complex relationships thoroughly, ultimately leading to better generalization and performance on future data.

Because the output has such a wide range of numerical values, two kinds of loss functions are tested in this work: the mean squared error (MSE) and the mean absolute percentage error (MAPE). The MSE loss function is given below:

$$J = \frac{1}{N}\sum_{i}(y_i^p - y_i)^2 \tag{10}$$

And the MAPE is described as follows:

$$J = \frac{1}{N}\sum_{i}\frac{|y_i^p - y_i|}{y_i + 1} \tag{11}$$

where J is the loss function value, N is the total number of data points, y_i^p is the predicted output value by the regression model, and y_i is the true output value. The plus one term in the MAPE equation prevents the equation from dividing by zero and also avoids ridiculously high loss values when the true output is close to zero. The loss functions are applied to the normalized data scaled by the MinMax Scaler, which keeps small data values that are close to 0 still close to 0 and scales large data values close to 1. As the original data's minimum value is retained, the data normalized by the MinMax scaler also retains the properties of the original dataset, which improves the ultimate performance of the model. When applied, these two loss functions intrinsically favor very different modeling patterns. MSE measures the average of the squares of the errors, treating all errors equally regardless of the magnitude of the true values. This results in data points with low true values having high percentage errors, as the model may focus more on minimizing absolute errors, which can disproportionately affect smaller values. Conversely, MSE can result in lower percentage errors for data points with high true values, as the absolute errors tend to be relatively smaller in proportion to the larger true values. This can be advantageous when precision for higher value predictions is critical. MAPE, on the other hand, measures the average absolute percentage error, ensuring that the model minimizes the percentage error across all data points. This results in a more uniform percentage error distribution, which is beneficial when the relative accuracy of predictions is more important. Each loss function has its benefits and drawbacks, which is why this work uses both to train the model. Using both MSE and MAPE allows for a holistic review of the data and the modeling process. This dual approach provides a more comprehensive understanding of the model's potential across the entire range of output values.

3. Results and analysis

3.1. Classification model performance

The performance of the classification models proposed in Section 2.3 are ideally evaluated within the context of a manufacturing environment. With the classifier model, there are four possible process outcomes: a pass is classified as a pass (true positive), a pass is classified as a fail (false negative), a fail is classified as a fail (true negative), and a fail is classified as a pass (false positive). Of these outcomes, the true positive and true negative outcomes are trivially good outcomes, as the classifier model is correct. The false negative, while not ideal, can be mitigated by manufacturing procedures. If all runs classified as fails are reevaluated at the metrology machine and manually measured to determine whether they truly failed, then the false negatives will be caught and correctly reclassified as passes. Thus, the main manufacturing concern is false positives, as there is no easy way to identify them; manually measuring all passes in addition to all fails would make the classifier model superfluous.

Generally speaking, most false positives will be eventually caught at future metrology steps or at the final metrology and reliability testing of the end product, which means that they will ultimately have little effect on product quality (Elsayed, 2012). The main impact of misidentifying a misprocessed wafer is that, as the misprocessed wafer moves through the process flow, it wastes resources and time. Thus, a high-performing classification model will have a low false positive rate as that minimizes wasted manufacturing resources, and any metrics used to analyze these models must be an indication of their false positive rates.

However, each model does not have a singular, representative false positive rate. Specifically, each model can be tuned to be more aggressive in flagging misprocessed wafers, lowering the false positive rate and increasing the false negative rate, or more conservative, which does the opposite. While confusion matrices are often used to evaluate classifier model performances, they are insufficient to evaluate the overall model performance, as a confusion matrix reflects the results of a single tuning approach. It cannot capture the model's performance across all possible tuning configurations. Other traditional criteria for evaluating classification model performance, such as overall accuracy, are also unsuitable because the binary outputs of all the datasets are highly imbalanced. As previously mentioned, a model that only predicts PASS may have a high accuracy but it will have a 100% false positive rate, rendering it meaningless in actual industrial applications. The Receiver Operating Characteristic (ROC) analysis offers a more robust evaluation method by examining the true positive rate (TPR) and false positive rate (FPR) of the model's predictions on test data at different thresholds. The output from the sigmoid function in the model's last layer is a continuous value in the range of [0,1]. Although a fixed threshold of 0.5 is traditionally used that classifies values higher than 0.5 as pass and those lower as fail, this threshold can be set to any value within the [0,1] range. With different thresholds, different TPR and FPR values are produced. For instance, setting the threshold to 0 (all pass) results in a TPR of 100% and an FPR of 100%. Conversely, setting the threshold to 1 (all fail) yields a TPR of 0% and an FPR of 0%. Thus, selecting an appropriate threshold involves balancing model sensitivity (TPR) and false acceptance rates (FPR).

In this context, the model's performance is evaluated using the Area Under the Curve (AUC) score. The AUC score is defined as the area under the ROC curve, which plots TPR on the y-axis and FPR on the x-axis across different thresholds. Ideally, a perfect model would achieve a TPR of 100% (max sensitivity) and an FPR of 0% (zero false alarms), resulting in an AUC score of 1. Conversely, a model that makes random guesses would have a TPR of 50% and an FPR of 50%, resulting in an AUC score of 0.5. The ROC-AUC score provides a comprehensive measure of model performance across all possible thresholds, making it particularly useful and widely applied for evaluating models on imbalanced datasets (Gonçalves et al., 2014).

Table 4
Single dataset training AUC score.

Dataset name	Average AUC score
T4-PM1	0.73
T7-PM1	0.73
T2-PM2	0.80
T5-PM2	0.48
T7-PM2	0.52

3.1.1. Single dataset model performances

The model is first trained on single tool-module datasets to evaluate if the training method is effective at making predictions that are significantly better than random guessing, which would have an AUC score of 0.5. For each dataset, five models are created via the cross-validation training process as described in Section 2.3.1. The average AUC score is defined as the mean value of the test scores for each trained model. Note that all ROC graphs are of the model whose performance best matches that of the average of the five models; this is the representative model. The scores for each tool-module combination are shown in Table 4.

The ROC-AUC plot of the representative model for all five datasets is shown in Fig. 5 The single dataset training results in Fig. 5 demonstrate that the amount of training data is a key factor in model performance. The two smaller datasets, T5-PM2 and T7-PM2, show unacceptable performance close to near-random guesses (AUC scores of around 0.5). In contrast, the three larger datasets have AUC scores significantly above 0.5, indicating successful model training and effective classification of the PASS/FAIL status of the product. For these three tool-module combinations, the models can achieve a FPR rate of about 35% to 40% when the TPR is around 80%. This means that, if 10 000 wafers are processed with 200 fails and 9800 passes (2% failrate), then 1960 wafers would be false negatives (20% of 9800 passes) and 70 wafers would be false positives (35% of 200 fails). This means that the overall rate of missed fails is less than 1% of the overall product throughput, which is considered high-performing. The single dataset cases suggest that data volume is crucial to model performance; with enough data, the models are able to effectively classify runs between PASS/FAIL. And with even more data, the AUC score can be further improved and the FPR reduced.

3.1.2. Multi dataset model performances

To validate the efficacy of data aggregation in enhancing model performance, the process is first investigated within each module (PM1, PM2). Specifically, this involves forming each unique superset between T4-PM1 and T7-PM1 for PM1 and each superset between T2-PM2, T5-PM2, and T7-PM2 for PM2. This results in three two-set supersets for PM2 and one two-set superset for PM1. PM2 also has the option to aggregate all three datasets. The optimal AUC score for a given tool-module combination is defined as the best AUC score among all the possible supersets that contain that tool-module combination, and the optimal AUC score for each tool-module combination is illustrated in Fig. 6. This analysis aims to determine whether combining datasets from different tools within the same module can lead to improved predictive performance, as measured by the AUC score.

From the same-module data aggregation results shown in Fig. 6, the AUC score for the T7-PM2 model trained on only the T7-PM2 dataset is just above 0.5, which is an unacceptable performance. However, for the best-case two- or three-set supersets, the AUC score improves past 0.6. The results for T5-PM2 are even better. In the case of T2-PM2, it is clear that the base T2-PM2 dataset contributes the most to the model and that the model receives little benefit when it is aggregated with the other smaller datasets. However, the contribution that the T2-PM2 dataset makes in the orange and green bars of the T7-PM2 and T5-PM2 models demonstrates substantial performance improvement for Tools 05 and 07, which have smaller datasets. Generally, data

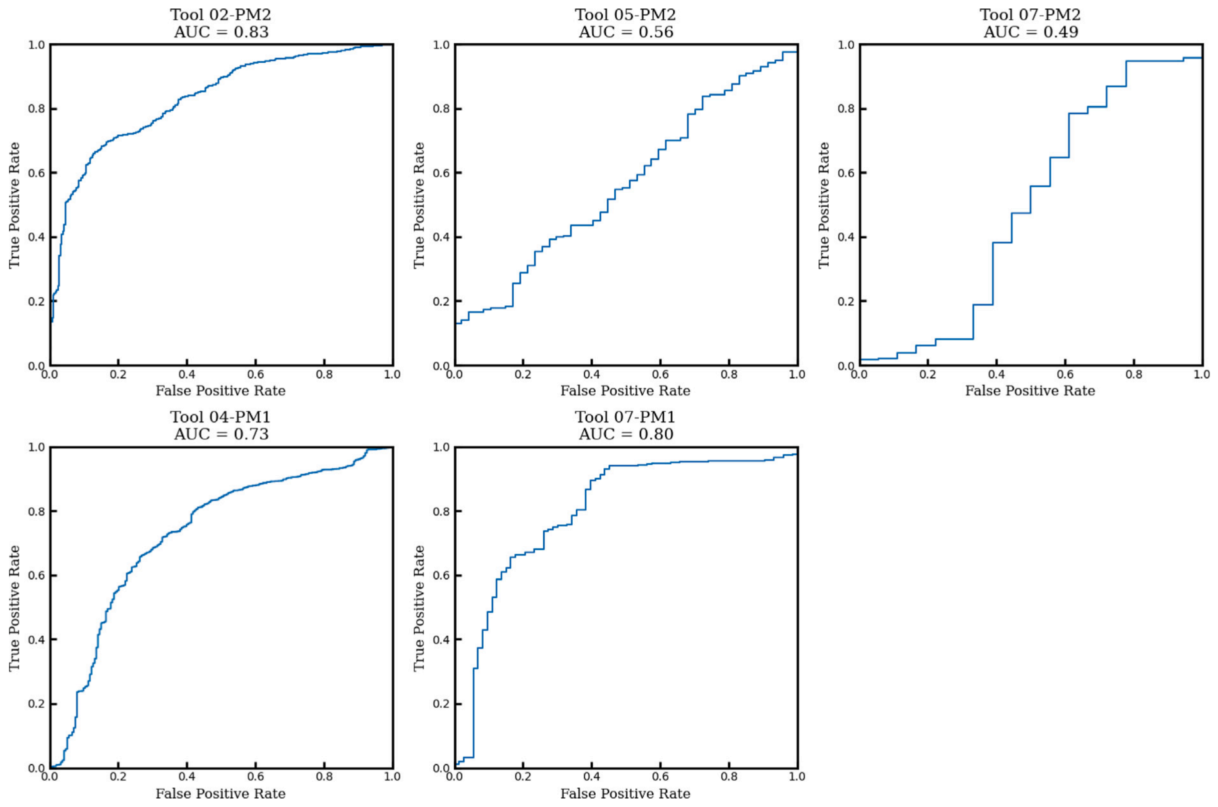


Fig. 5. ROC plot for all five datasets. The performance of the models trained on datasets T5-PM2 and T7-PM2 is similar to random guesses, while the models based on the other three datasets have superior performances.

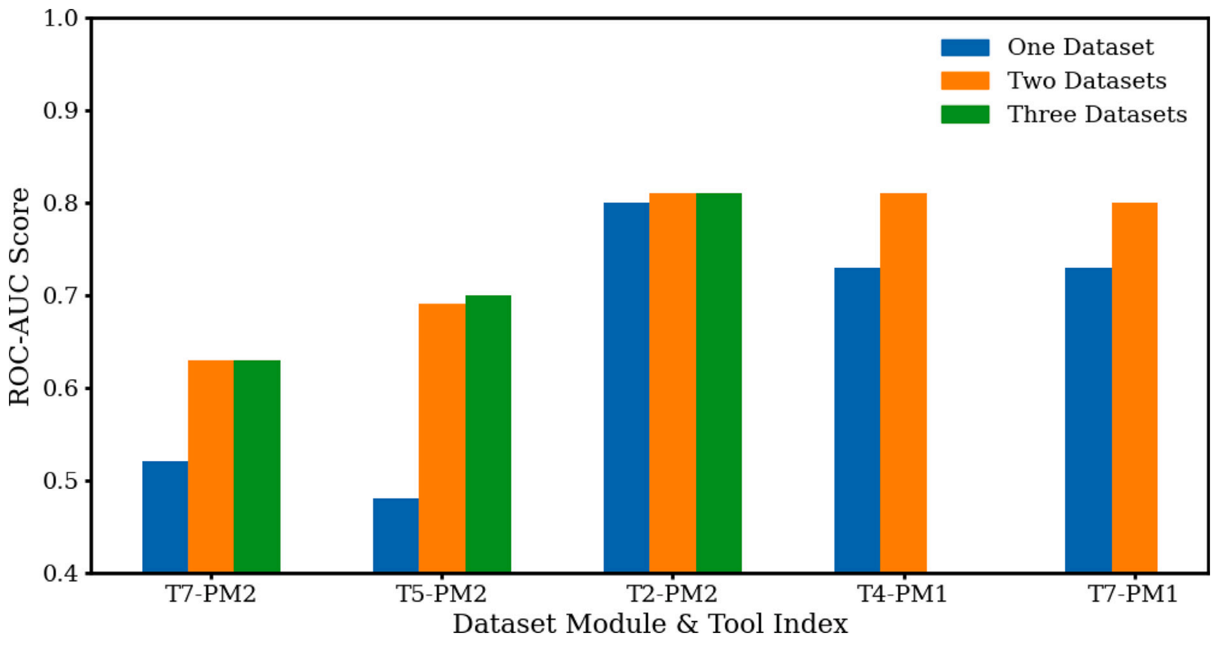


Fig. 6. Best possible AUC score for all five tool-module combinations. Model performances improve substantially as data increases for all five cases. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

aggregation substantially improves model performance for tool-module combinations with smaller datasets, such as T5-PM2. Furthermore, the three-set aggregation in PM2 allows the models to achieve a FPR of around 50% with a TPR of about 80%, as shown in the ROC plot in Fig. 7. For the base model trained on only one dataset, an 80% TPR would correspond to an 80% FPR, which would result in an overall missed fail rate of 1.6% given a 2% fail rate. However, the model trained with multi-dataset aggregation would have a missed fail rate

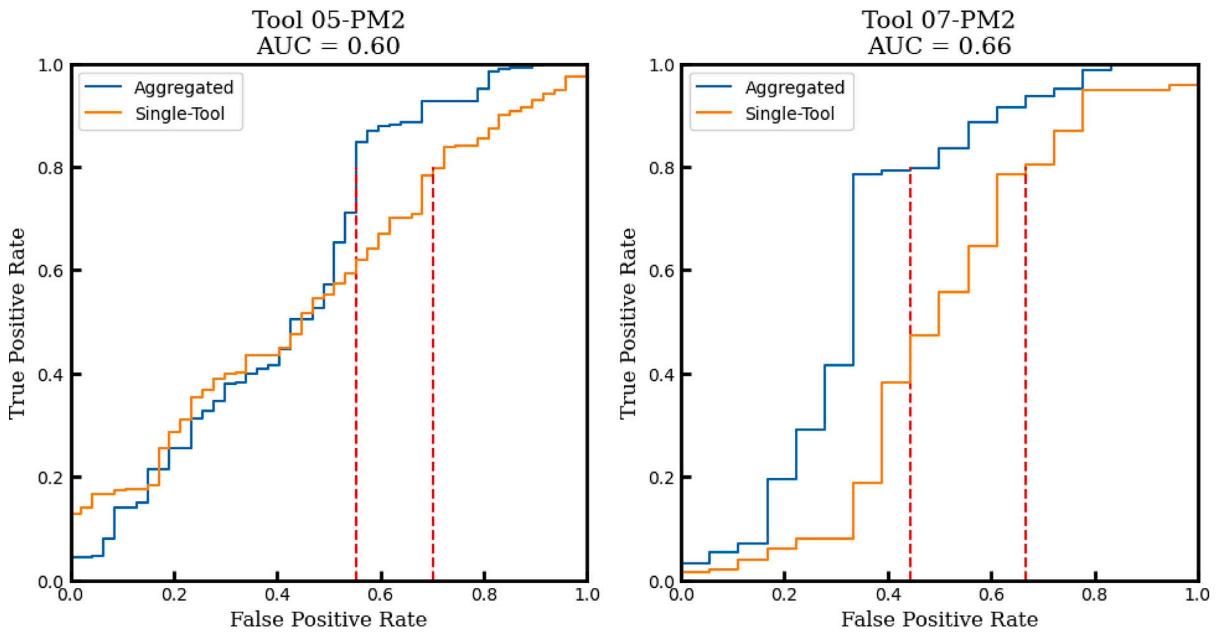


Fig. 7. ROC plots of the representative models of T5-PM2 and T7-PM1, which were trained by aggregating all available datasets in the module. The performances are noticeably better than that of random guessing and single-set models. The FPR values for an 80% TPR value have also significantly improved.

Table 5
Difference score between each pair of datasets.

	T4-PM1	T7-PM1	T2-PM2	T5-PM2	T7-PM2
T4-PM1	N/A	0.019	0.017	0.083	0.037
T7-PM1	0.019	N/A	0.011	0.083	0.032
T2-PM2	0.017	0.011	N/A	0.080	0.029
T5-PM2	0.083	0.083	0.080	N/A	0.101
T7-PM2	0.037	0.032	0.029	0.101	N/A

of 1%, a significant improvement. Additionally, even the tool-module combinations that have the largest datasets improve with data aggregation. Compared to their single-set results in Fig. 6, the AUC scores of the aggregated models reach over 0.8 for both T2-PM2 and T7-PM1, as shown in Fig. 6. These results underscore the effectiveness of data aggregation in bolstering model accuracy, especially for tool-module combinations with limited datasets.

3.1.3. *Candidate dataset selection method*

While the previous section shows that data aggregation improves model performance, it does not explain how it should be conducted. Additionally, there is no easy way to evaluate a model's performance without actually developing and testing the model. Even in Fig. 6, the displayed datasets had varying levels of improvement. Thus, to minimize the presence of false positives, it is necessary to examine how to optimally aggregate datasets together. To that end, five datasets (T4-PM1, T7-PM1, T2-PM2, T5-PM2, T7-PM2) are examined to assess the effectiveness of the indexing method explained in Section 2.3.2 that is used to select ideal datasets to aggregate with the base dataset, also called candidate datasets. The difference scores for each dataset pair as calculated with Eq. (8) are shown in Table 5.

The candidate dataset selection criterion for a given dataset is to choose the paired dataset with the smallest difference score. For example, to improve the T4-PM1 model, the T4-PM1 dataset should be aggregated with the T2-PM2 dataset, as the T4-PM1/T2-PM2 pair has the smallest difference score (0.017) compared to the T4-PM1/T7-PM1 (0.019), T4-PM1/T5-PM2 (0.083), and T4-PM1/T7-PM2 (0.037) pairs. For the same reason, the candidate datasets for T5-PM2 and T7-PM2 are both T2-PM2; the T5-PM2/T2-PM2 (0.080) pair is the smallest amongst all the T5-PM2 pairings, and the same holds for the T7-PM2/T2-PM2 pair. Note that the candidate dataset relationship is not necessarily true in reverse. While the candidate dataset for T5-PM2 may be T2-PM2, the candidate dataset for T2-PM2 is not necessarily T5-PM2. From Table 5, it can be seen that the candidate dataset for T2-PM2 is actually T7-PM1, with a difference score of 0.011. Additionally, to aggregate three datasets, or when choosing the second candidate dataset, the difference score has to be recalculated between the current two-set superset and all the other datasets. For example, to determine the candidate dataset for the T4-PM1/T2-PM2 superset (SS1), the difference scores for the SS1/T7-PM1, SS1/T5-PM2, and SS1/T7-PM2 pairs must be recalculated.

To further assess the effectiveness of this data aggregation metric, four datasets are examined in a case study: all three PM2 datasets and T4-PM1. It is still possible to exhaustively test the 15 possible unique supersets formed from these four datasets, which will allow us to evaluate both the idea that data aggregation generally improves model performance and the effectiveness of the candidate dataset selection method. First, the best possible AUC score for each tool-module combination is found by exhaustively creating a model for every possible superset and then selecting the superset that yields the highest AUC score for that tool-module. The best possible AUC score at each superset size is shown in Fig. 8. Then, the best possible AUC score is compared to the AUC score obtained by aggregating candidate datasets, and this is shown in Fig. 9 for two-set supersets and Fig. 10 for three-set supersets.

Fig. 8 validates the idea that data aggregation generally improves model performance because the best possible AUC scores for all four tool-module combinations increase as more datasets are aggregated. Additionally, in Fig. 9 for two-set aggregation and Fig. 10 for three-set aggregation, the AUC scores of the proposed data aggregation strategy based on statistical analysis methods aligns with the best possible score, with T7-PM2 being an exception in both cases. The exception of T7-PM2 can be explained by examining the difference score between its historical data and its future data. In Table 6, the difference score between the historical data used to train models and the future data used to test models is displayed for five tool-module combinations. Notably, the difference score for T7-PM2 is the largest of the examined datasets, which implies that there is significant variability between the modeling set and testing set that may impact data aggregation effectiveness as the data aggregation strategy is purely based on the

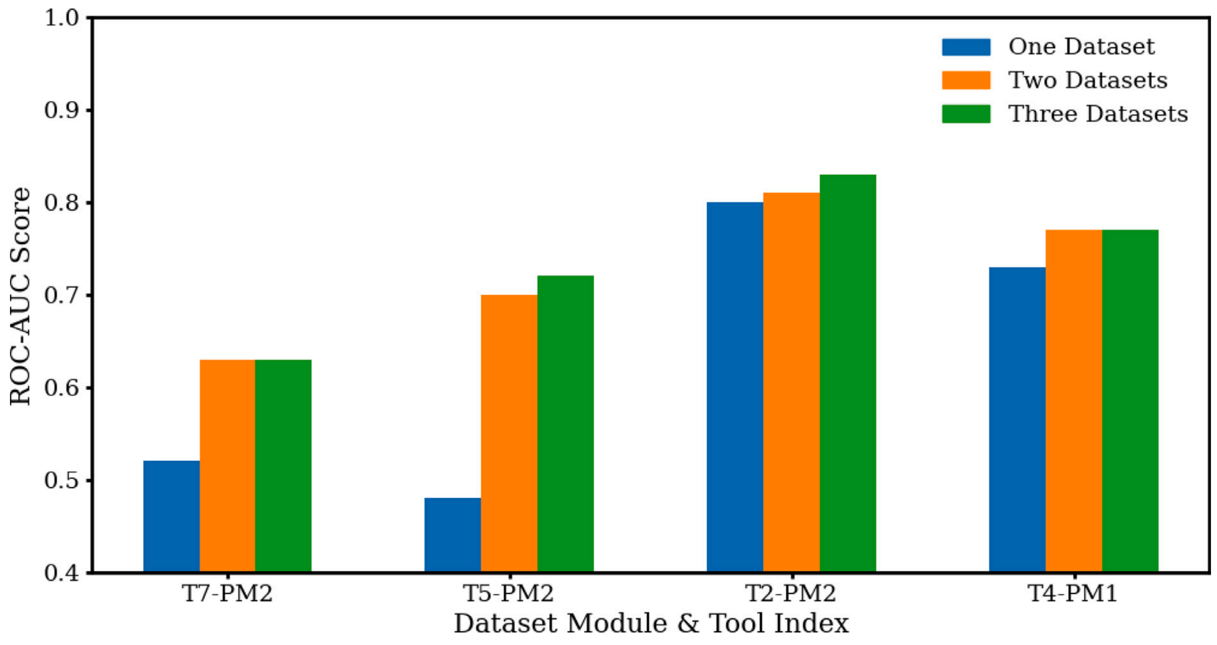


Fig. 8. Best possible AUC score as a function of how many datasets are aggregated among the four datasets. Model performance improves significantly for all tool-modules as data aggregation increases.

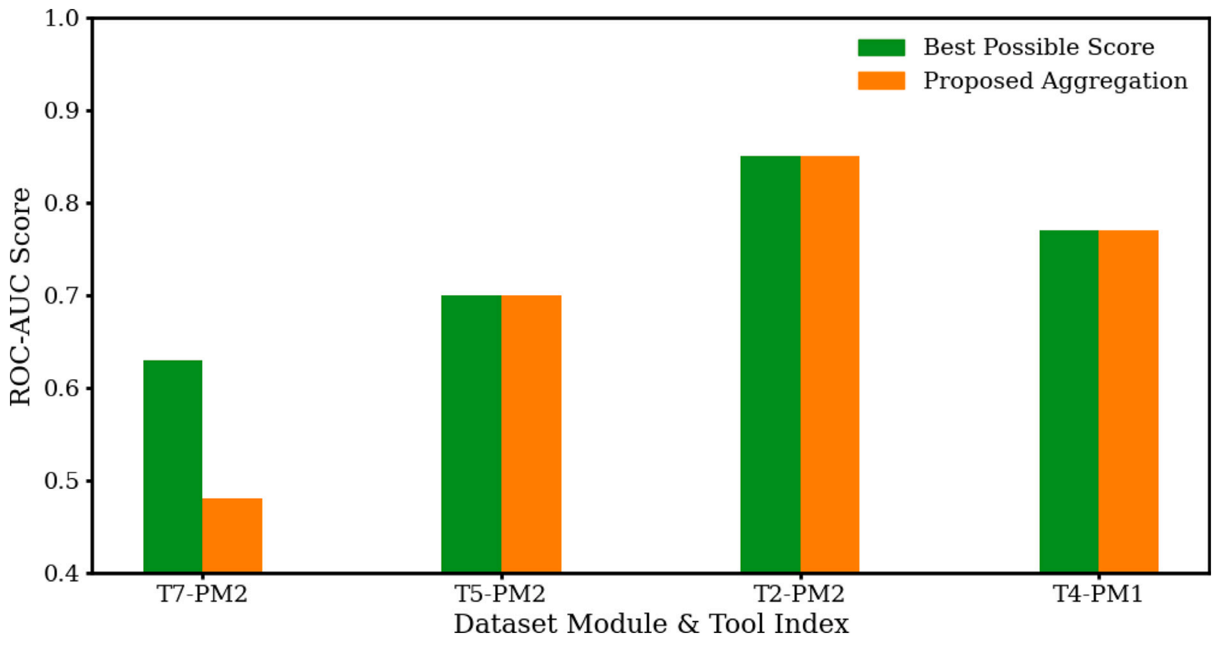


Fig. 9. Comparison between the best AUC score for a **two-set** superset and the AUC score for the superset proposed by the candidate dataset selection method. Proposed AUC scores conform with the best-case scores for most tool-modules.

historical dataset. This indicates that the proposed statistical analysis methods are effective for this case study and can be considered as a preliminary solution for selecting candidate datasets for aggregation.

One of the most important applications of the candidate dataset selection method is to address scenarios where numerous tool-module combinations are involved. In these situations, it is impractical to exhaustively test all possible combinations of data aggregation to determine the ideal superset for each tool-module combination. To further test the capabilities of the candidate dataset selection method, the five datasets in Table 5 are reexamined. For this analysis, not all possible superset combinations are tested. Instead, only the supersets proposed by the candidate dataset selection method are examined. The resulting AUC scores are then evaluated to validate the effectiveness of the proposed aggregation method that was previously demonstrated to work well for the four dataset case study. This approach aims to show that the aggregation strategy will continue to be feasible even as the

Table 6
Historical and future data difference score.

Dataset name	Difference score
T4-PM1	0.089
T7-PM1	0.052
T2-PM2	0.082
T5-PM2	0.090
T7-PM2	0.093

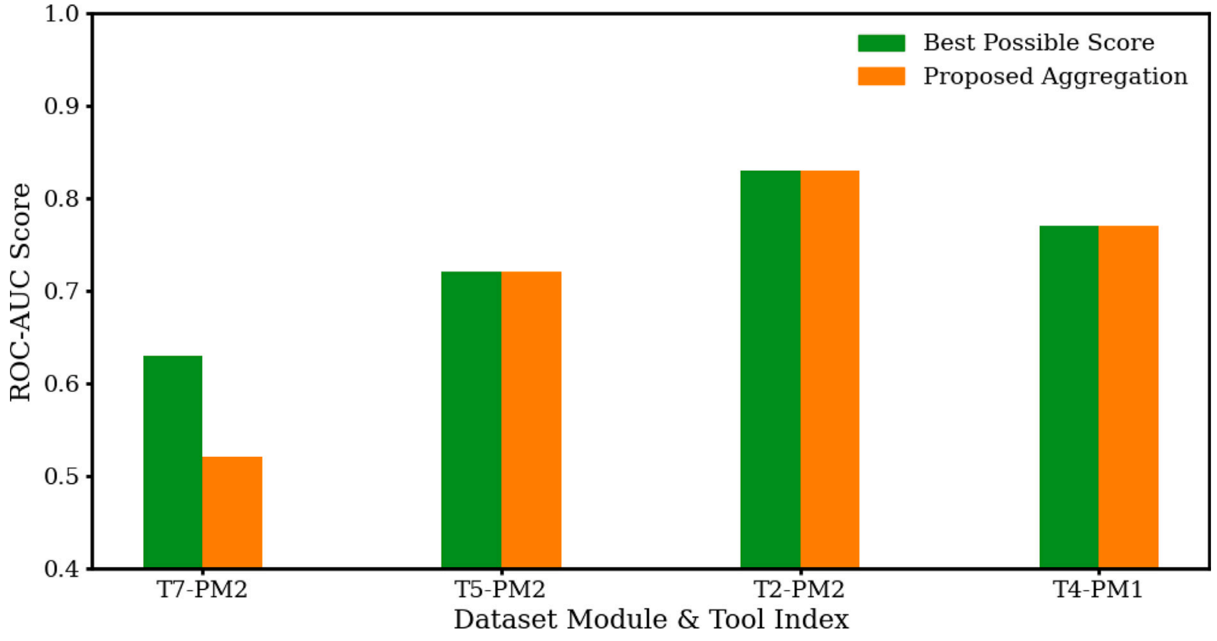


Fig. 10. Comparison between the best AUC score for a **three-set** superset and the AUC score for the superset proposed by the candidate dataset selection method. Proposed AUC scores conform with the best-case scores for most tool-modules.

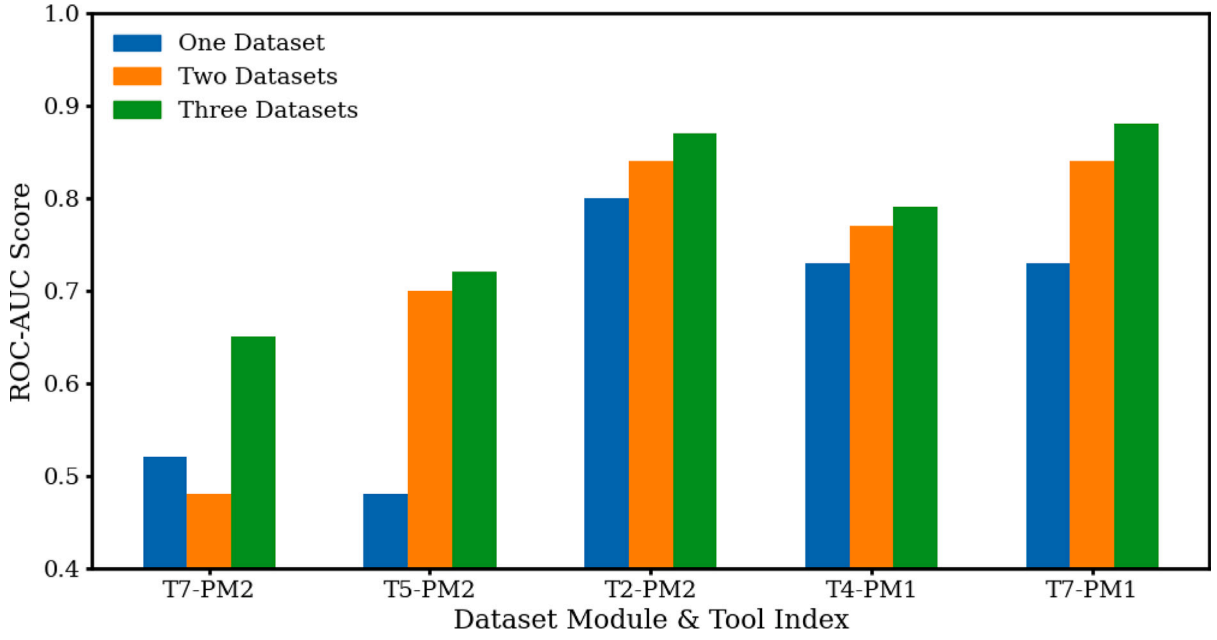


Fig. 11. AUC scores for the models trained on the supersets proposed by the candidate dataset selection method. The AUC scores for all the tool-module combinations improve as more datasets are aggregated into the modeling set.

number of datasets increases and prove that statistical analysis methods can identify the ideal candidate dataset without training numerous models. Among the five datasets, the candidate dataset for T2-PM2 is T7-PM1 as this pair has the lowest difference score (0.011), and the remaining datasets have the same candidate datasets as in the previous case study. The aggregation AUC scores are shown in Fig. 11, and a specific example is shown in Fig. 12. Fig. 11 illustrates a noticeable improvement in the performance for all the tool-module models. Specifically, T2-PM2 and T7-PM1 achieve AUC scores of 0.87 and 0.88, respectively, which is considered high performing. As shown in Fig. 12, the ROC plots for T2-PM2 and T7-PM1 indicate that aggregating data has lowered the FPR from between 35%–40% to 20% at 80% TPR, representing a significant improvement.

Fig. 11 also demonstrates the importance of data volume and how data aggregation can supplement datasets with small volumes. This is especially in the case of T7-PM2, which previously underperformed in the four dataset case study. Table 2 states that it has a small volume of data of 2722 compared to T4-PM1, T7-PM1, and T2-PM2, which have dataset sizes in the tens of thousands. For the cases where only one or two datasets are used, the T7-PM2 model performs poorly, with an average ROC-AUC score of 0.5, which is the same as randomly guessing. However, by aggregating three datasets, it achieved an AUC score close to 0.7 and an FPR of about 40% with a TPR of around 80%. This demonstrates the effectiveness of the data aggregation strategy based on statistical analysis methods at enhancing model performance, particularly in scenarios where there are so

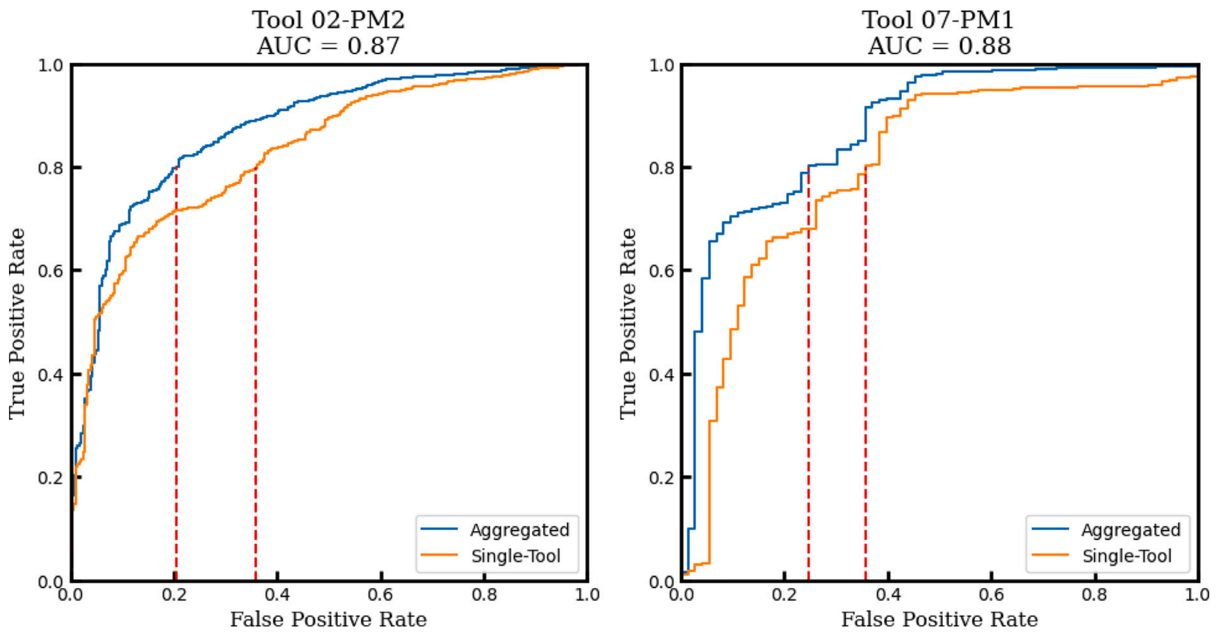


Fig. 12. ROC plots for model performance on T2-PM2 and T7-PM1, which are trained by aggregating three datasets. The performances are noticeably better than the models trained on single datasets. The FPR values at 80% TPR are improved from good (around 40%) to perfect (around 20%).

many datasets that it is impractical to exhaustively test all possible combinations.

While data aggregation is certainly powerful, it also has its own limitations. The two tool-module combinations with the smallest datasets, T5-PM2 and T7-PM2, cannot attain a FPR of 20% at a TPR of 80% like T2-PM2, a tool-module with a large dataset. This is because the original dataset is simply too limited. Although data aggregation can greatly increase the amount of training data by combining similar datasets, the newly added data will never perfectly follow the distribution of the original data, or in other words, data aggregation dilutes the “identity” of the original dataset. This forms a complex balance between data aggregation increasing the data volume, which improves model performance, and data aggregation diluting the identity of the original dataset, which decreases model performance. Additionally, the improvement effect from increased data volume experiences diminishing returns as seen in Fig. 11; T7-PM1’s performance experiences a sizable increase when moving from one to two datasets, but its performance only experiences a minor boost when moving from two to three datasets. Thus, the diminishing returns of increased data volume and the cost of identity dilution implies that there exists an optimum where further data aggregation would lead to decreased rather than increased model performance.

This optimum will also differ for each individual classification problem and dataset. For example, complex problems naturally require larger datasets in comparison to simpler problems, which reduces the diminishing returns effect of increased data volume. Additionally, the dilution extent of the original dataset’s identity is dependent on the potential candidate datasets; if the datasets to be aggregated are very similar, then the dilution effect will be weak whereas if the datasets to be aggregated are very different, the dilution effect will be strong. Nonetheless, the improved AUC scores and reduced FPRs highlight the efficacy of data aggregation at limiting the number of missed misprocessed wafers. A key part to data aggregation is the candidate dataset method and its ability to search for ideal datasets that optimize model accuracy and efficiency when dealing with extensive datasets as randomly selecting datasets will result in minor improvements if not decreases in performance. It should be mentioned that the point-biserial analysis and difference scores are not guaranteed to nominate the ideal candidate dataset. Nevertheless, the results of this work demonstrate that the proposed data aggregation method can still significantly improve the performance of industrial classification models, validating the practicality and effectiveness of the approach in real-world applications.

3.2. Regression model performance

The regression model is evaluated with the median percentage error metric. Percentage error is chosen as it is commonly used in industrial settings for numerical data that spans a wide range from 0 to over 5000 Å, and it is calculated with the following equation:

$$J_{percentage} = \frac{|y_{pred} - y_{true}|}{y_{true}} \tag{12}$$

where $J_{percentage}$ is the percentage error, y_{pred} is the predicted oxide thickness, and y_{true} is the measured oxide thickness. y_{true} spans from 0 to 5000 Å, and some values are close to 0 Å, e.g. 0.001 Å. The percentage error for these data points with small y_{true} values will be abnormally large despite their small absolute error. To minimize the influence of these outliers, the median, rather than the mean, is used to represent the overall percentage error as it is a more robust and representative measure of model performance. The R^2 criterion is not applied here because of the imbalanced data distribution; the data points are dense around a low target region (from 0 to 200) and sparse at a high target region (from 200 to over 5000). A model that mostly focuses on the dense, low target region can still receive a good R^2 score even if it performs badly on the sparse, high target region. Thus, the median percentage error is a better criterion for evaluating the overall model performance on all ranges of data. As previously stated, the regression task is more complex than the classification task, so only the T2-PM2, T4-PM1, and T7-PM1 datasets are analyzed. Furthermore, since the training, validation, and test datasets for the regression task are all sorted by time rather than randomly selected as was the case for the classification task, the median percentage error of all three datasets is an informative metric and will be shown.

Since the target oxide thickness is always known, a benchmark model can be constructed. Specifically, the benchmark model is defined as a model that always predicts the measured oxide thickness to be the same as the target oxide thickness; it always guesses that the product will pass metrology. Then, by calculating the median percentage error of this benchmark model on the validation set, it can act as a

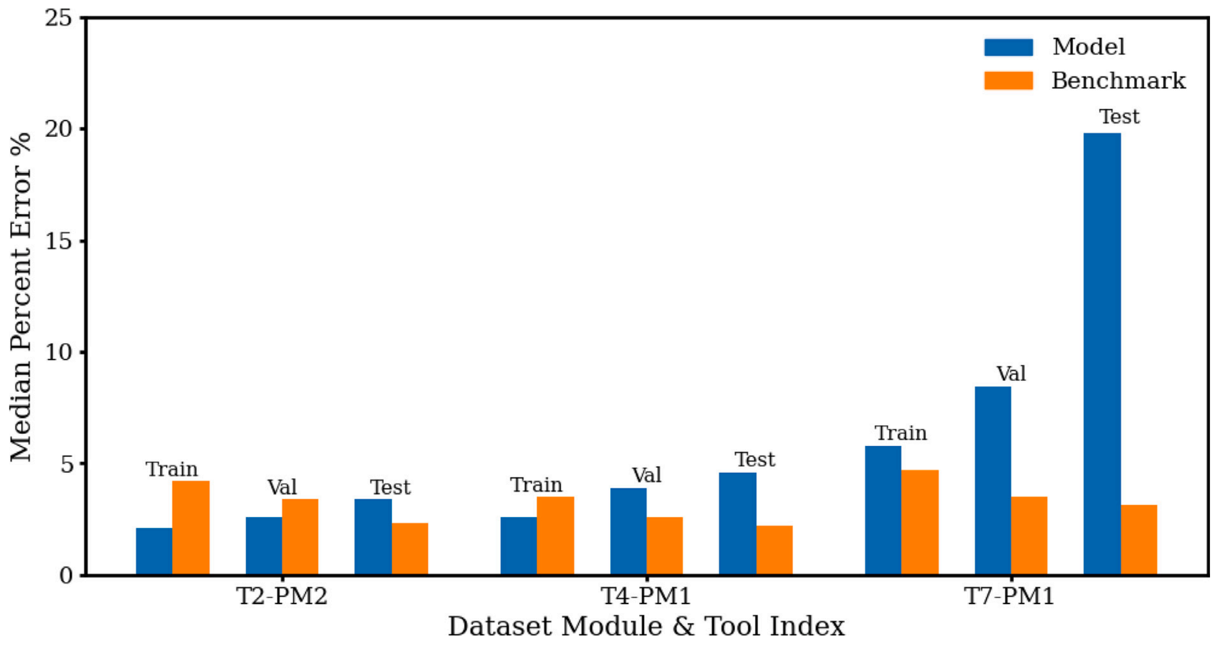


Fig. 13. The median percentage error after training with the **MAPE** loss function, compared to the benchmark model. Each tool-module labeled on the x-axis has three groups of bars, which represent the training, validation and test sets, respectively. Within each groups of bars, the blue bar is the median percentage error of the trained regression model, and the orange bar is the median percentage error of the benchmark model. Fig. 14 to Fig. 18 have the same formatting. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

baseline for performance evaluation. The minimum requirement for an acceptable regression model is to outperform the benchmark model. By comparing the performance of various regression models against this benchmark, we can better assess the regression model's accuracy at predicting the measured oxide thickness, especially in an industrial context where reliability is crucial.

3.2.1. MAPE training results

The results for all the models trained with the Mean Absolute Percentage Error (MAPE) loss function for the three specified datasets split by the training, validation and test datasets are shown is shown in Fig. 13. Fig. 13 shows that, although the trained regression model outperforms the benchmark model on the training and validation datasets for T2-PM2, the regression model still underperforms in comparison to the benchmark model on the test set. For T4-PM1, the trained regression model even fails to outperform the benchmark model on the validation set. Moreover, for T7-PM1, the trained model does not outperform the benchmark model on any of the dataset. Overall, it is evident that any regression models trained with the MAPE loss function cannot surpass the benchmark model when evaluated over the entire dataset.

Even when the data is divided into runs with a target oxide thickness value lower than 200 Å (low target) and runs with a target oxide thickness higher than 200 Å (high target) as shown in Figs. 14 and 15, the previous conclusion remains consistent. The regression model trained with the MAPE loss function does not outperform the benchmark model on either the low target or the high target test sets. The designation of "low" and "high" targets is arbitrarily determined by the real-world behavior of the tool-module combinations. Specifically, the runs with a target below 200 Å and those above 200 Å have very different behaviors. By splitting up the data and analyzing it in detail, it gives a more detailed and comprehensive sense of the overall model performance by examining the model performance in different applications.

3.2.2. MSE training results

The results for all the models trained with the Mean Square Error (MSE) loss function for the three specified datasets split by the training,

Table 7
Percentage of runs with high targets in each dataset.

Tool-Module	Training	Validation	Testing
T2-PM2	20%	17%	38%
T4-PM1	21%	39%	38%
T7-PM1	8%	18%	11%

validation and test datasets are shown in Fig. 16. Fig. 16 shows that the median percentage error of the regression model trained with the MSE loss function is worse than that of the regression model trained with the MAPE loss function. This is expected as the MAPE loss better aligns with the evaluation criterion. The regression models trained with the MSE loss function perform worse than the benchmark model for all three tool-module combinations across the training, validation, and test sets. This trend is also evident in the model's performance on low target data points, as shown in Fig. 17. Due to the intrinsic properties of the MSE loss function, which treats absolute errors on the high and low target data points the same, the median percentage error on runs with low targets is even higher than the median percentage error on all runs. Conversely, this means that the model performs exceptionally well on runs with high targets, as shown in Fig. 18. When only examining the runs with high targets, the median percentage error of the trained regression models is significantly lower than those of the benchmark model across the training, validation, and test datasets.

The results shown in Figs. 16–18 prove that the MSE loss function can achieve exceptional performance in terms of median percentage error for runs with high targets. Nevertheless, if the runs with high targets only constitute a negligible portion of the overall dataset, then the good performance of the regression model in this category is not particularly noteworthy. In this work, however, the high target data points contribute significantly to the overall dataset, as shown in Table 7. For instance, over one-third of the data points in the test sets of T2-PM2 and T4-PM1 have high target oxide thicknesses.

Given this substantial representation, although the model trained with the MSE loss function cannot be confidently used in all cases, it holds a distinct advantage in predicting the measured oxide thickness when the target oxide thickness is high. Since the target value is always

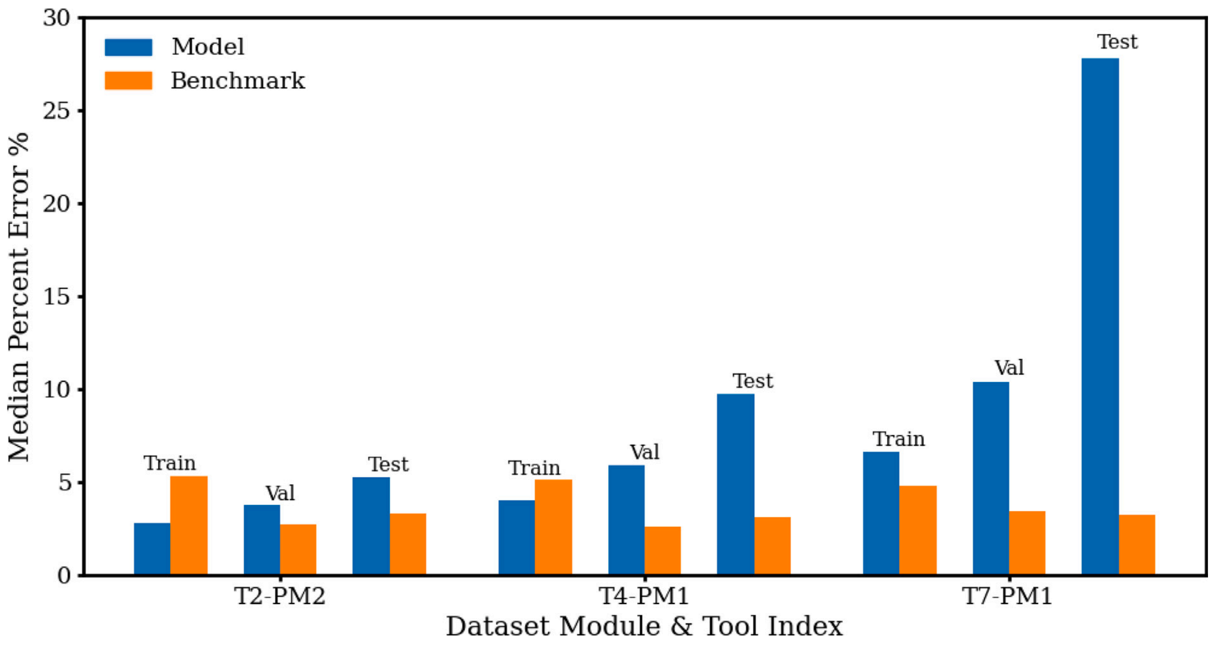


Fig. 14. Comparison of the median percentage error between various regression models trained with the **MAPE** loss function for the training, validation and test sets and the benchmark model on runs with target oxide thicknesses of **less than** 200 Å.

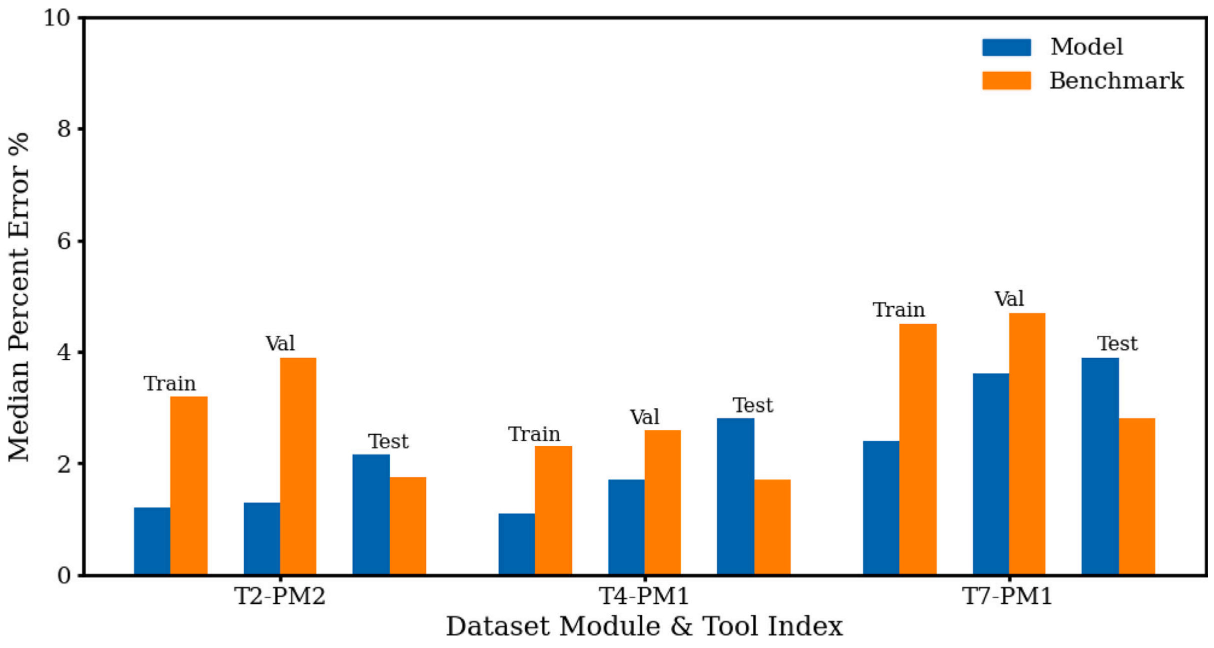


Fig. 15. Comparison of the median percentage error between various regression models trained with the **MAPE** loss function for the training, validation and test sets and the benchmark model on runs with target oxide thickess of **over** 200 Å.

known before processing, this characteristic can be strategically utilized to employ the model in areas where it shows strong performances.

The poor performance of the regression models for both the MSE and MAPE loss functions, compared to that of the benchmark model, can primarily be attributed to the exceptionally high pass rate, which reaches 98% in some datasets. In industry, a common pass criterion is for the result to be within a specified threshold of the target value. This high pass rate indicates that most runs are successfully etched and that their measured oxide thickness is very close to the target thickness, resulting in the benchmark model having a very low median percentage error in most cases. However, this means that it is very difficult to train a regression model that outperforms the benchmark model across all target values. This stands in stark contrast to the results of the classification model, where the favorable ROC-AUC curves in Fig. 12 show that the trained models can and do outperform a benchmark model that randomly guesses the outcome.

The reason for the different results stems from the difference in complexity between the tasks. The available process data contains enough

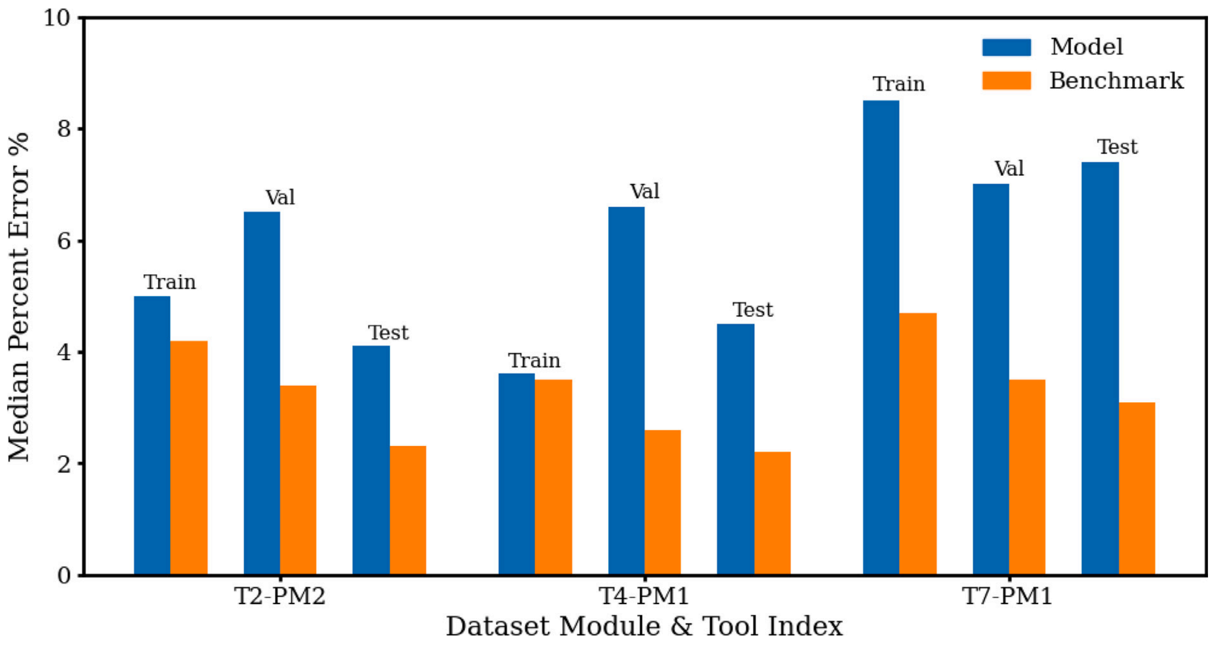


Fig. 16. The median percentage error after training with the **MSE** loss function, compared to the benchmark model. For all tool-module combinations and all datasets, the trained regression model performance is worse than that of the benchmark model.

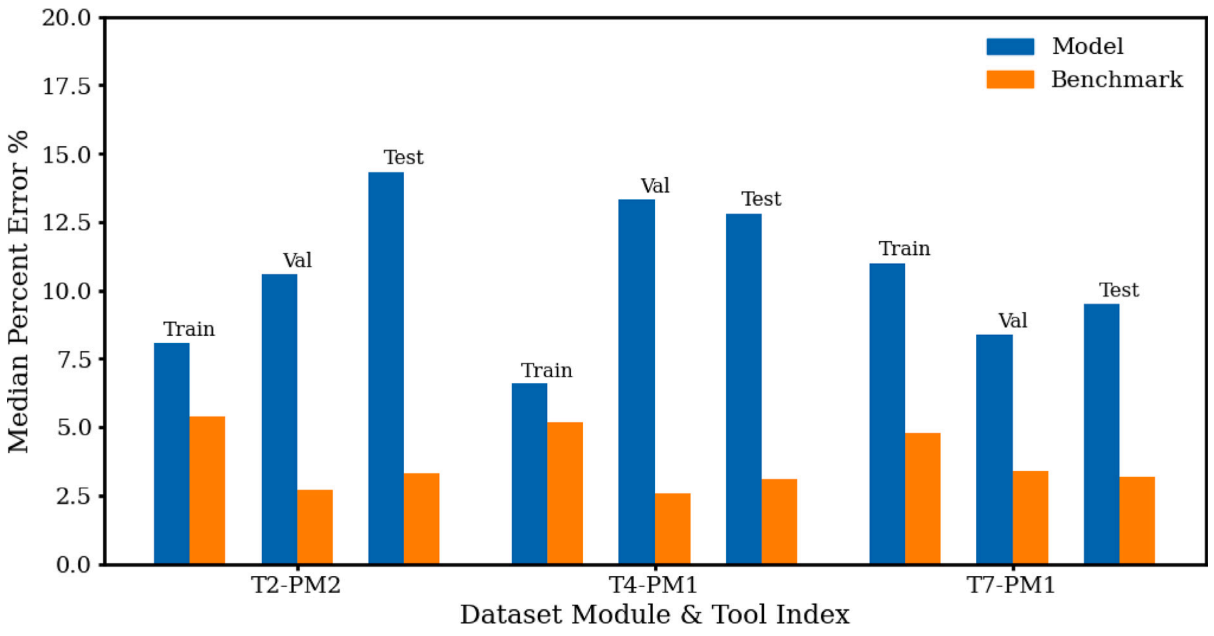


Fig. 17. Comparison of the median percentage error between various regression models trained with the **MSE** loss function for the training, validation and test sets and the benchmark model on runs with **low** target oxide thicknesses. The performance of the regression model here is even worse than in Fig. 16.

information to train a high-performing classification model that simply allocates the results into two classes. However, the regression model must predict the exact oxide thickness value from a wide range of possible values. As the regression task is many times more complex than the classification task, a high-performing regression model requires process data that contains deep insight into the process itself. Thus, the reason why the regression models do not outperform the benchmark model is because the process data does not contain enough information regarding the process. To improve the performance of the regression models, more complex process data, such as time-series data, must be incorporated into the training process. In conclusion, machine learning-based soft sensors are powerful at predicting the physical properties of the process, but only if the model is trained on sufficiently insightful process data collected from physical sensors.

4. Conclusion

This paper describes machine learning-based soft sensors trained on process data from five industrial etching reactors for two tasks: PASS/FAIL classification and oxide thickness regression. A data aggregation method is proposed to improve the model performance for the predictive classification task. In addition, for the regression task of predicting measured oxide thickness, both MSE and MAPE losses are tested for model training. The results presented in this paper validate the hypothesis that data aggregation and statistical analyses can

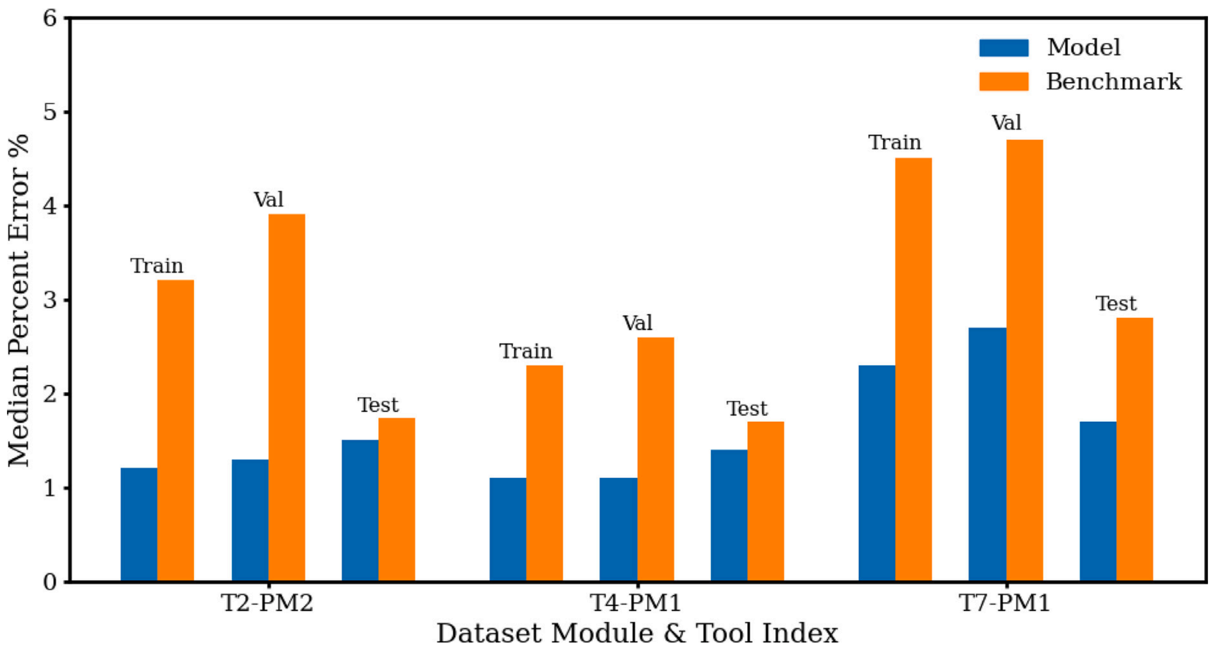


Fig. 18. Comparison of the median percentage error between various regression models trained with the **MSE** loss function for the training, validation and test sets and the benchmark model on runs with **high** target oxide thicknesses. The performance of the regression model is better than that of the benchmark model on all examined runs for all three training, validation and test datasets.

significantly enhance the binary PASS/FAIL prediction accuracy and robustness of feed forward neural network models on industrial etching reactors. This is done by aggregating datasets using point-biserial correlations and difference scores as metrics to choose candidate datasets for aggregation, particularly for tools with initially small data volumes. The two datasets with the smaller number of data points improved from a score barely above random guessing ($AUC \approx 0.5$) when trained on a single dataset to a significantly better performance ($AUC \approx 0.65$) with a 40% false positive rate (FPR) at a 80% true positive rate (TPR) when trained on an aggregation of three datasets out of five possible datasets. The other datasets with relatively larger datasets still benefited from data aggregation; their model performance improved from $AUC \approx 0.8$ to $AUC \approx 0.9$, and the FPR improved from around 35% at 80% TPR to 20%. The statistical analyses accurately chose the best candidate dataset for aggregation, aligning with the results shown by exhaustively testing every possible dataset combination. These results confidently show the effectiveness of data aggregation and using statistical methods as an aggregation strategy. Additionally, for regression models that predict oxide thicknesses, while the MAPE loss function was more effective for overall percentage error minimization, the trained model could not outperform the benchmark model that always predicted the measured oxide thickness to be the target thickness. Although the model trained with the MSE loss function did not yield satisfactory predictions across all data points, it exhibited exceptional performance for runs with high target oxide thickness with target values of over 200 Å. This finding is useful for specific processes with high target values, as these target values are typically known in industrial settings. Consequently, the strategic use of MSE for processes with high target values can enhance regression model performance in these specific scenarios.

CRediT authorship contribution statement

Feiyang Ou: Writing – original draft, Software, Methodology, Investigation, Formal analysis, Data curation, Conceptualization. **Henrik Wang:** Writing – original draft, Software, Methodology, Investigation, Formal analysis, Data curation, Conceptualization. **Chao Zhang:** Writing – review & editing, Supervision, Investigation. **Matthew Tom:** Software, Formal analysis, Data curation, Conceptualization. **Sthitie Bom:** Writing – review & editing, Supervision, Investigation. **James F. Davis:** Writing – review & editing, Investigation, Conceptualization. **Panagiotis D. Christofides:** Writing – review & editing, Project administration, Conceptualization.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

Financial support from the National Science Foundation and the Department of Energy, USA is gratefully acknowledged. Industrial data and support from Seagate Technology, USA is gratefully acknowledged. This work used computational and storage services associated with the Hoffman2 Shared Cluster provided by UCLA Office of Advanced Research Computing's Research Technology Group.

References

Burg, D., Ausubel, J.H., 2021. Moore's law revisited through intel chip density. PLoS One 16 (8), e0256245.
Christofides, P.D., Davis, J.F., El-Farra, N.H., Clark, D., Harris, K.R., Gipson, J.N., 2007. Smart plant operations: Vision, progress and challenges. AIChE J. 53 (11), 2734–2741.
Coleman, S., Kerr, D., Zhang, Y., 2022. Image sensing and processing with convolutional neural networks. Sensors 22 (10), 3612.
Elsayed, E.A., 2012. Overview of reliability testing. IEEE Trans. Reliab. 61 (2), 282–291.
Fuchs, C., 2018. Industry 4.0: the digital german ideology. Triplec: Commun. Cap. Crit. 16 (1), 280–289.
Gonçalves, L., Subtil, A., Oliveira, M.R., de Zea Bermudez, P., 2014. ROC curve estimation: An overview. REVSTAT-Statist. J. 12 (1), 1–20.
Hong, S., An, N., Cho, H., Lim, J., Han, I.-S., Moon, I., Kim, J., 2023. A dynamic soft sensor based on hybrid neural networks to improve early off-spec detection. Eng. Comput. 39 (4), 3011–3021.
King, R.D., Orhobor, O.I., Taylor, C.C., 2021. Cross-validation is safe to use. Nat. Mach. Intell. 3 (4), 276–276.
Kingma, D.P., Ba, J., 2017. Adam: A method for stochastic optimization. arXiv:1412.6980. URL: https://arxiv.org/abs/1412.6980.

Lee, M., Bae, J., Kim, S.B., 2021. Uncertainty-aware soft sensor using Bayesian recurrent neural networks. Adv. Eng. Inform. 50, 101434.

Nguyen, T.Q., Hoang, T., Zhang, L., Dobre, O.A., Duong, T.Q., 2024. A survey on smart optimisation techniques for 6G-oriented integrated circuits design. Mob. Netw. Appl. 1–18.

Nguyen, P., Ivanov, D., Sgarbossa, F., 2023. A digital twin–based approach to reinforce supply chain resilience: Simulation of semiconductor shortages. In: Proceeding of IFIP International Conference on Advances in Production Management Systems. Springer, pp. 563–576.

O'Donovan, P., Leahy, K., Bruton, K., O'Sullivan, D.T., 2015. Big data in manufacturing: a systematic mapping study. J. Big Data 2, 1–22.

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., Duchesnay, E., 2011. Scikit-learn: Machine learning in python. J. Mach. Learn. Res. 12, 2825–2830.

Perera, Y.S., Ratnaweera, D., Dasanayaka, C.H., Abeykoon, C., 2023. The role of artificial intelligence-driven soft sensors in advanced sustainable process industries: A critical review. Eng. Appl. Artif. Intell. (ISSN: 0952-1976) 121, 105988.

Ranganathan, G., 2021. A study to find facts behind preprocessing on deep learning algorithms. J. Innov. Image Process. 3 (1), 66–74.

Rehmer, A., Kroll, A., 2020. On the vanishing and exploding gradient problem in gated recurrent units. IFAC-PapersOnLine (ISSN: 2405-8963) 53 (2), 1243–1248, 21st IFAC World Congress.

Sarker, I.H., 2021. Machine learning: Algorithms, real-world applications and research directions. SN Comput. Sci. 2 (3), 160.

Songkhla, S.N., Nakamoto, T., 2021. Overview of quartz crystal microbalance behavior analysis and measurement. Chemosensors 9 (12), 350.

Sun, Q., Ge, Z., 2021. A survey on deep learning for data-driven soft sensors. IEEE Trans. Ind. Inform. 17 (9), 5853–5866.

Wang, G., Jia, Q.-S., Zhou, M., Bi, J., Qiao, J., Abusorrah, A., 2022. Artificial neural networks for water quality soft-sensing in wastewater treatment: a review. Artif. Intell. Rev. 55 (1), 565–587.

Waoo, A.A., Soni, B.K., 2021. Performance analysis of sigmoid and relu activation functions in deep neural network. In: Intelligent Systems: Proceedings of SCIS 2021. Springer, pp. 39–52.

Xu, B., Wang, N., Chen, T., Li, M., 2015. Empirical evaluation of rectified activations in convolutional network. arXiv preprint arXiv:1505.00853.

Zhang, C., Yella, J., Huang, Y., Qian, X., Petrov, S., Rzhetsky, A., Bom, S., 2021. Soft sensing transformer: hundreds of sensors are worth a single word. In: Proceedings of IEEE International Conference on Big Data. pp. 1999–2008.