ECAI 2024
U. Endriss et al. (Eds.)

doi:10.3233/FAIA241042

Equipment Condition-Integrated Predictive Modeling for Optimized Scheduling of Ion Implantation in Semiconductor Manufacturing

Andreas Laber[a,b,*], Martin Gebser[b] and Konstantin Schekotihin[b]

[a]Infineon Technologies Austria AG
[b]University of Klagenfurt
ORCID (Andreas Laber): https://orcid.org/0000-0002-2837-8270, ORCID (Martin Gebser): https://orcid.org/0000-0002-8010-4752, ORCID (Konstantin Schekotihin): https://orcid.org/0000-0002-0286-0958

Abstract. In view of the high total cost of semiconductor manufacturing assets, respective equipment needs to be as productive as possible. To avoid needless idling and unnecessary downtime, scheduling and maintenance strategies are important in practice. This paper presents a novel approach to reduce the substantial setup costs inherent to ion implantation by deriving scheduling constraints based on current equipment conditions. Consequently, a supervised learning pipeline is established that utilizes built-in sensors and process target data to accurately predict setup costs. The derived constraints are integrated into scheduling, thereby enhancing its efficiency through dynamic dispatching adaptations. The application of our method is projected to significantly improve equipment availability by avoiding more than 100 hours of potential downtime annually.

1 Introduction

Given the prevalent winner-takes-most market situation and the capital-intensive nature of the semiconductor industry, companies constantly seek ways to improve their time-to-revenue and cost position. The use of Artificial Intelligence (AI) can be a valuable tool for achieving both. Semiconductor manufacturing is prone to be impacted the most in absolute terms by AI according to Goeke et al. [11], because respective equipment comes not only with high acquisition costs, but there are also significant costs associated with the idling and maintenance of assets. Sophisticated scheduling methods aim to reduce standby times, thereby improving the equipment's throughput. However, these schedules frequently encounter disruptions due to unforeseen circumstances, such as the introduction of new preemptive tasks or sudden unavailability of equipment. Such challenges cannot be addressed by modern schedulers directly. Therefore, in this paper, we focus on incorporating Predictive Maintenance (PdM) approaches [37, 3] into scheduling to reduce the substantial setup costs inherent to ion implantation equipment.

Data is extracted from the semiconductor equipment in real-time by the Advanced Process Control (APC) system. APC, introduced over two decades ago, enhances process stability and product quality by monitoring equipment-internal sensors. Additional systems are built on top of APC: Fault Detection and Classification (FDC) and Run-to-Run (R2R). FDC systems detect abnormal system behavior during processing by employing rule-based approaches, whereas R2R is capable of adapting process parameters dynamically based on measurements and predictions to compensate for process variations. Respective process target specifications are listed in recipes, with the content varying based on the equipment platform employed. Switching from one recipe to another incurs setup times, which can serve as valuable input for scheduling.

Therefore, to increase the Overall Equipment Effectiveness (OEE) of implantation equipment, we propose to integrate equipment condition-derived constraints into the scheduling process. This leads us to the following key contributions:

Datasets We collect logistic and sensor information from APC for implanters from two distinct manufacturing facilities and provide the corresponding setup costs as labels. The resulting dataset comprises an expert-selected subset of 110 features to represent the current equipment condition and three distinct labels. We publish an anonymized version of this data to support future research.[1]

Predictive Modeling We conduct a comparative study on the performance of state-of-the-art models and share the related source code for predicting setup costs together with the dataset.

The predicted setup costs are incorporated as additional constraints into our pre-existing scheduling solution, allowing for enhanced and automated decision-making. This novel application of Machine Learning (ML) improves the efficiency of implantation processes, which play a pivotal role in semiconductor device fabrication.

The structure of this paper is as follows: We begin with a discussion of contemporary ion beam implantation techniques and establish the research goals in Sections 2 and 3. Subsequently, Section 4 provides a review of existing PdM strategies based on APC, FDC, and R2R, with a special focus on studies involving ion implantation equipment. Section 5 outlines the process of data extraction and dataset preparation, and provides details about the proposed ML pipeline. Faced challenges and limitations arising from the required integration into the productive scheduling system are emphasized in Section 6. Results of the experimental study on APC data are presented in Section 7. The paper concludes with Section 8, where we summarize our findings and discuss future work.

* Corresponding Author. Email: andreas.laber@infineon.com

[1] https://zenodo.org/records/11084332

2 Ion Beam Tuning

2.1 Process Description

There are different types of implantation equipment available: medium current, high current, and high energy [10]. Ion implantation requires doping with desired species, e.g., boron, phosphorus or arsenic to control the electrical properties of the substrate, usually a silicon wafer. These charged dopants are accelerated in an electric field and directed onto wafers. A schematic of a cutting-edge medium current ion implanter is shown in Figure 1. The doping gas, e.g., boron trifluoride BF_3, is injected in an ion source, wherein the gas molecules are ionized and a plasma is generated. Typically, electric currents of a few Ampere (A) are flowing within this plasma. To avoid breakdown of the plasma, ion currents of only a few mA are extracted from the ion source, with voltages up to 80 kilovolts (kV). The ions are then accelerated to their final energy, e.g., 200 kiloelectronvolt (keV) acceleration energy results in final boron ion velocities of 2000 km/s. The charged ions are deflected by the magnetic field of a mass separator magnet. Ions deviating from the desired mass are deflected differently and subsequently intercepted within the analyzer magnet. Electrostatic lenses and magnets are distributed throughout the system to maintain the focus of the ion beam. Mechanical wafer movement and electrostatic scanner plates can direct the ions to different positions on the wafer. Typically, the goal is to implant a uniform dose in the range from $1e^{11}$ to $1e^{16}$ at/cm^2 throughout the wafer. The unit process target is achieved by applying the intricate settings specified in a recipe to the equipment. For implantation equipment, this comprises the definition of targets for dopant gas flows, voltages, currents, magnet field strengths and the positioning of the extractor magnet along the x, y and z axes. Continual monitoring of feedback values against target values is necessary to identify and correct any offsets. This is a challenging process, given the interdependent relationships between the components.

2.2 Tuning Procedure

Ion implantation is a complex process that involves a series of components, each meticulously adjusted in a recurring sequence, to produce a desired ion beam output. The recommended tuning order prescribes to firstly adjust the components of the ion source, i.e., dopant gas flow, filament current, arc voltage and the analyzer magnet field, to ensure beam quality. Subsequently, we need to find appropriate settings for a distinct set of parameters, i.e., source magnet field, extraction suppression voltage, extractor electrode position and focus voltage, that influence the properties of the beam. If specifications cannot be fulfilled with the chosen parameter set, the process is repeated. This reiterative setup procedure is called ion beam tuning.

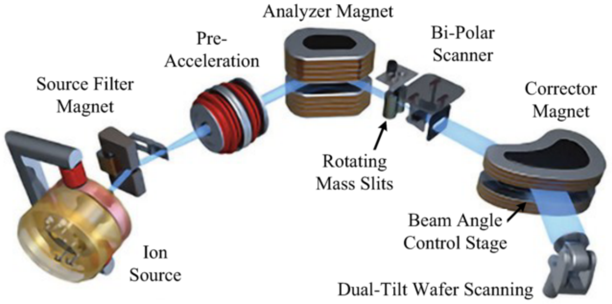

Figure 1. Schematic of medium current ion implantation beamline components of VIISta 900XP [2].

There is an increasing chance between preventive maintenance cycles that the ion beam tuning is not feasible, given the degrading equipment condition. Simultaneously, the processed recipes have a significant impact on the tuning success rate: There is no tuning required for a follow-up lot with the same recipe, apart from routinely checking that the required uniformity is still achieved. Switching between recipes with the same species and only minor adjustments to the energy, the tuning process takes two to four minutes on average. On the other hand, changing the species and switching to a distant energy level is time-consuming to tune. If a tuning attempt fails multiple times, the setup is aborted, and either the equipment is maintained or a less demanding recipe is scheduled, both resulting in OEE loss.

Manual ion beam tuning is time-intensive and its success rate depends on the person's experience performing it. Alternatively, there is a built-in feature for automatic beam tuning based on the proprietary algorithm initially developed for classic VIISta medium current implanters. The product family of VIISta 900 evolved through continuous improvements spanning more than two decades [31, 33, 12]. Currently, Applied Materials offers the VIISta 900XP and VIISta 900 3D [2]. According to Viviani and Falco [39], the state vector, a set of variables that represents the current state of a system, for ion beam control consists of source gas density inside the arc chamber, species, filament current, arc current, source magnet field, suppression current, extraction current, beam current and the positioning of the extractor electrode along the x, y and z axes. Such a vector exists for each combination of energy, species and beam current. To maintain the system in a stable and efficient condition, optimization is performed on a second set of parameters. For instance, the output beam current, which is determined by the arc current, is one such parameter. Other parameters in this group, like gas pressure and source magnet current, are also optimized to ensure that the system remains stable. As another example, Cucchetti et al. [9] state that by manipulating six parameters, the transmission through the beamline can be tuned to fit custom recipe specifications. These parameters include the x, y and z positions of the extraction electrode, the extraction suppression voltage, the source magnet field and the focus voltage.

3 Research Objectives

In the context of manufacturing, a recipe specifies parameters to meet the requirements of a process step. For ion implantation, the most critical parameters are energy, species, and dose. Whenever recipes are changed, a setup is necessary, i.e., ion beam tuning. This tuning is instrumental in minimizing deviations from recipe target specifications under varying equipment conditions.

We develop two distinct predictive models that forecast the results of the tuning process in terms of its success and duration. The former being of utmost importance since it enables proactive scheduling adaptations to enhance equipment uptime. By incorporating respective success predictions as (soft-)constraints, scheduling systems can proactively address potential tuning issues. This is achieved by avoiding the dispatch of lots to equipment that is currently not in the condition to tune for the associated recipes efficiently. Dynamic updates to the setup cost matrix with more precise tuning duration estimates through regression analysis assist in identifying more efficient setup sequences. To enable these improvements, we address the following *Research Questions* (RQ):

RQ-1. How can we perform data engineering on large manufacturing datasets?

RQ-2. To what extent can we leverage physical principles of implantation to improve feature engineering for ML purposes?

RQ-3. Which ML model is most effective in predicting ion beam tuning on an equipment/recipe basis, based on relevant metrics?

RQ-4. How can we integrate predictions into scheduling to optimize performance by reducing poor tuning attempts?

The workflow detailed in Figure 2 aims to improve OEE by enhancing the heuristic-based scheduling methodology. It integrates constraints from predicted transition costs between recipes, using sensor data from APC. Outputs are stored in a database and utilized by the scheduling system, reducing downtime from failed setup attempts.

3.1 Cost-Benefit Analysis

To evaluate the performance of our classification model, we use Precision and Recall. To achieve the economically optimal balance, it is important to consider the impact of each error type. *False Positives* can reduce scheduling flexibility by preventing the assignment of lots with misclassified recipes to manufacturing equipment. To mitigate this negative effect, the scheduling system bypasses recipe transition restrictions when a dispatch list contains only a few processable lots. Larger work centers are less affected, as they offer more potential destinations for a given lot. In contrast, *False Negatives* lead to potentially avoidable equipment downtime, presenting a substantially more severe disadvantage. Therefore, our main optimization potential lies within minimizing the frequency of wrongfully predicted tuning failures. Ultimately, the negative effects of inaccurate predictions are only quantifiable on a case-by-case basis via simulations or by retrospectively analyzing work center performance.

We collected data on the equipment's operational status over time. Thereby distinguishing between periods of production and downtime due to setups, and joining this data with its tuning history. This culminated in categorized downtime durations, from which we derived time lost due to setup issues. This analysis indicates that removing unsuccessful tuning completely would result in an equipment uptime increase of approximately 2 percentage points. As the Recall of our classification model decreases, the uptime improvement potential lessens alongside. When also accounting for overruled predictions due to pre-existing scheduling heuristics, uptime still rises by more than 1 percentage point. This translates to an annual uptime extension by roughly 100 hours, which allows for additional layer starts without requiring extra capital expenditure.

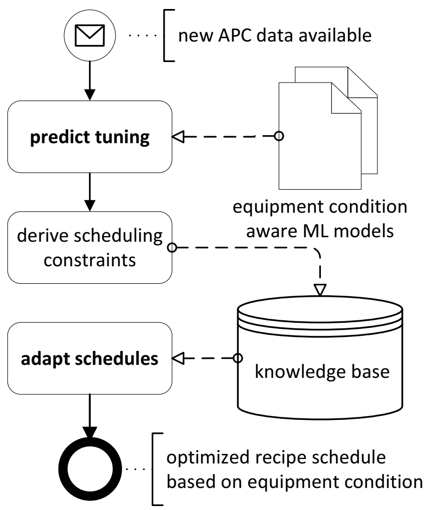

Figure 2. Whenever we get new APC data, pre-trained models predict tuning outcome. Schedules are adapted based on derived constraints.

4 Related Work

AI's potential to elevate performance across domains is evidenced for scheduling [26], PdM [17], FDC [19, 20] and R2R [41, 43, 42]. Additional use cases estimate the Remaining Useful Life (RUL) of an equipment component [15, 14] or search for recipe parameters and equipment configurations that further improve process efficiency [23]. In the studies of Yugma et al. [44] and Stehli et al. [34], the importance of integrating scheduling and APC in semiconductor manufacturing is discussed, emphasizing that these two aspects are mutually dependent and that their synergy can enhance performance.

When focusing on research performed on implantation equipment, we found the following to be relevant: Lin and Horng [27] performed FDC on ion implantation equipment amidst recipe-induced complexity. Susto et al. [36] estimated the RUL of tungsten filaments, a vital component of the ion source. The same task was analyzed by Susto and Beghi [35] while researching feature extraction techniques on time-series data. Kurakula and Trujillo [21] detected ion source faults preemptively by data mining respective sensor behavior. They found that spikes in filament current indicate an imminent breakdown of the ion source. Lang et al. [25] developed a scalable anomaly detection method and tested it on ion implantation and plasma etch recipes. Moreover, Lang et al. [24] modeled the relationship between the implant duration and dose profile to allow for dosing uniformity optimizations. Yang et al. [40] modified dispatching rules to favor recipes of the same job family for reducing setup times due to beam tuning.

We framed the classification problem of predicting ion beam tuning success, but only presented preliminary results based on log files in Laber et al. [22]. We expand upon our previous research in several ways. First, we recognize that setup times significantly impact the uptime of implanters of all types: medium current, high current, and high energy. Therefore, we investigate scalable methods that can be easily rolled out to similar equipment types, due to the same underlying data structure within APC. Second, we go beyond predicting the success or failure of tuning by also examining the duration of tuning attempts. This additional layer of analysis allows us to further optimize the OEE of implantation equipment. Lastly, our models' outputs are not just theoretical predictions, but they are incorporated directly into the scheduling system, dynamically updating the setup cost matrix and adding optimization constraints in the form of forbidden recipe transitions. This practical application of our research results in tangible improvements in equipment uptime.

5 Implementation

5.1 Dataset

Tuning is required, whenever the recipe is changed. The sensor data recorded from the equipment prior to this process can serve as input for machine learning algorithms, which are essential for developing a robust scheduling approach. The equipment tracks internal traces of over 860 unique status variables, primarily derived from feedback values of built-in components and sensors. These traces yield statistics such as minimum, maximum, average, and standard deviation for each processed wafer, multiplying the available parameters by at least fourfold. Out of these, we consider a derived dataset of a semiconductor manufacturer with a subset of 769 parameters, carefully curated to monitor the quality of the equipment's manufacturing process. However, over time a substantial volume of data accumulates. To address this challenge efficiently, we leverage automated data pipelines that extract, transform, and load (ETL) data. These pipelines are built on Apache Spark, a powerful framework for large-scale data processing. The resulting dataset consists of several types of data, including strings for equipment or recipe names, timestamps for the start and end of setup and production processes, and primarily numerical data for sensor readings, such as gauge and vacuum pressure, or the voltage and current of components. Figures 3 and 4 illustrate on an exemplary basis APC data for one equipment (denoted as m_1) over time and colored per recipe. Additionally, the dataset includes binary labels indicating whether the ion beam tuning was successful or not, as well as the amount of time it took to complete the tuning. The tuning process of interest is initiated prior to the processing of the first wafer of the upcoming lot. Other cases of tuning are removed from the dataset, e.g., when there is tuning within a lot to re-check the uniformity, as these can not be avoided through dispatching adaptations. Thus, APC data closest to our decision point during inference is obtained from the most recently processed lot. Consequently, we need to match the sensor data collected while processing this lot to the upcoming tuning attempt. In this process, we also transform our data to be in the (X, y)-shape needed for supervised learning, X containing the features of our observations and y resembling the corresponding tuning labels. Though the label information is in the same table, it still needs to be joined to the correct database records. There is no primary key to join our sensor data with label information, thus we define the join criteria based on equipment and time columns. After these transforms the data is persisted into a data lake table. With the continuous operation of production equipment, there is more data available, which is added to this table for re-training purposes.

5.2 Machine Learning Pipeline

Feature Selection. To identify the most relevant features for the VIISta 900XP equipment platform, implant domain experts screened the available APC parameters and classified them based on their direct and indirect impact on the tuning procedure. This process resulted in the sub-selection of 110 features.

Feature Engineering. We introduce process target values as separate columns, specifically, dose, species, and energy. Adding the same information for the upcoming recipe, enables us to calculate

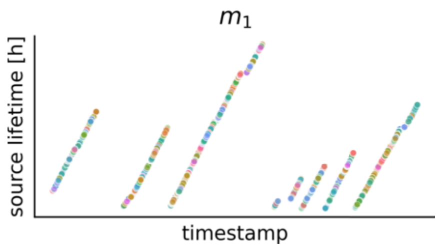

Figure 3. Ion source lifetime is *component-specific*, and thus is not influenced by different recipes. We observe a classical saw-tooth curve, with an increasing counter until it is reset upon replacement.

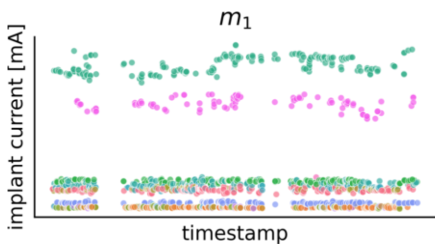

Figure 4. Implant current is *recipe-dependent*, in contrast to Figure 3.

absolute offset values to indicate the distance between recipes to the model. On the same note, we introduce a baseline heuristic for tuning duration estimation. The Chebyshev distance (D_{cheby}) captures the larger distance between the current and upcoming minimum/maximum scaled values for either dose or energy. This yields $D_{cheby} \in [0, 1]$, which we scale by a constant factor to align with the upper limit of allowed tuning time. The respective value defaults to a high value for species changes, as these require major equipment-internal reconfiguration. Longer periods of idling are known to cause instabilities, therefore we also calculate the elapsed time between two lots. During inference, we accordingly measure the time between the last observed $runstart$ in our database and the current system time. Furthermore, we transform the categorical species value into the corresponding atomic mass unit, e.g., representing boron as a scalar value of 11 instead of B. The lifetime of the ion source is reduced when predominantly running boron recipes. However, this can be mitigated by running any other species. To indicate this aspect to the model, we increment a counter for every boron recipe run and decrement the same for every other species. When the ion source is replaced, we reset the counter. After ensuring the dataset is sorted by time, we drop date columns, alongside the recipe name as we already extracted all relevant information. Finally, to convert our mixed-type dataset to a strictly numerical one, we encode the remaining categorical information, such as the equipment name column. To prevent identification of the underlying process details, a pseudonymized and scaled version of this dataset accompanies this paper.

Preprocessing. To ensure the optimal performance of our model in a productive setting, we impose certain restrictions. The model is trained only on data that is available at the point of decision-making. To maintain the integrity of the dataset as a time series, we avoid shuffling while splitting into train and test sets, and use TimeSeriesSplit() for cross-validation. The train set ($\mathcal{T}$) includes $38,848$ samples, while the test set ($\mathcal{E}$) comprises $9,713$ samples. After splitting, the data is scaled and centered on $\mathcal{T}$ via RobustScaler(). Analogously, median-value imputation substitutes missing data. The distribution of tuning success varies per equipment, typically there is a $85:15$ success/failure ratio present. This ratio shifts in favor of tuning success if only the first tuning attempt is retained, thereby dropping consecutive setup failures. To mitigate imbalance and enhance decision boundaries, hard-to-classify instances in $\mathcal{T}$ are pruned using TomekLinks undersampling [38]. Comparable techniques, such as SMOTE [5], did not perform well on our dataset.

Modeling. Several state-of-the-art ML models are evaluated:

- Scikit-learn's (1.4.0) Random Forests *rf* [4], k-Nearest Neighbors *knn*, Support Vector Machine *svm* based on stochastic gradient descent, Gaussian Naive Bayes *gnb* and histogram-based gradient boosting machines $hist_{gbm}$ [30];
- Boosted tree variants: LightGBM (4.2.0) $light_{gbm}$ [18], XGBoost (2.0.3) *xgb* and its random forest implementation xgb_{rf} [6];
- Standalone Multi-Layer Perceptrons *mlp* and Long-Term Short Memory *lstm* [13] network via Tensorflow/Keras (2.14.0) [8];
- Gated Adaptive Network for Deep Automated Learning of Features *gandalf* [16] via PyTorch (2.2.2).

In the interest of maintaining a fair comparison, we assessed the ML models in their default configurations, apart from appropriately indicating the class imbalance. As for the models of the deep learning (DL) domain: For *mlp* we used three dense layers with 64 nodes each and intermediate dropout layers removing roughly 0.25 nodes during training. Our *lstm* model architecture consists of an LSTM

Table 1. We provide tuning outcome predictions on recipe transition level: probability values for tuning failure and tuning duration in seconds.

From	To	Failure	Dur. [s]
Recipe 1	Recipe 2	0.09	241
Recipe 1	Recipe 3	0.75	
Recipe 2	Recipe 4	0.15	600

layer followed by batch normalization, dense, and dropout layers. During our experimentation phase, we trialed providing various sequence lengths to the *lstm*. However, we achieved the best performance by training the model on single observations only, indicating that we fail to extract meaningful information out of past sequences. Notably, *gandalf* did not require a custom network topology definition, as it ingests data into a customizable feature abstraction layer, and feeds the additional information into a multi-layer perceptron to output suitable predictions.

Hyperparameter Tuning. In our study, we leverage the Optuna framework [1] to mitigate overfitting through regularization for better generalization to new data. We therefore define the parameter search space and Optuna identifies an optimal combination of dataset-specific hyperparameters for our models based on cross-validation scores. Whenever applicable, we focused on tuning a comparable set of hyperparameters across our models. Model-dependent hyperparameters with a regularization effect include boosting type, learning rate, lambda, dropout, feature sampling, and tree depth.

Evaluation. For binary classification on imbalanced datasets, Average Precision and the Matthews Correlation Coefficient are suitable metrics if classes are equally important [32, 7]. In our case there is more cost associated with the minority class, thus we chose a variant of the popular F_1-score to reflect that Recall is more important than Precision. For the regression task, we optimize for Mean Absolute Error and Root Mean Squared Error. To ensure generalizability, we employed cross-validation on dataset $\mathcal{T}$. Additionally, the model was evaluated on a test dataset $\mathcal{E}$, simulating real-world production conditions. This evaluation on $\mathcal{E}$ provided insights into the model's expected performance during deployment. For improved model interpretability and effective communication with domain experts, we employ Shapley value analysis via the SHAP Python library [28]. This analysis allows us to understand the relative contribution of each feature to the model's predictions, facilitating clear explanations to experts and building trust in the model's capabilities.

5.3 Scheduling Integration

As new APC data about the equipment condition becomes available, the process flow of the proposed productive solution is triggered. The dispatch list continuously contains lots that require implantation as next step in their processing sequence. Trained models predict the ion beam setup in two distinct ways. The respective output is persisted as a lookup table of recipe transitions, as depicted in Table 1.

Our scheduling model, along with its subsequent workflows, governs all operational decisions. These decisions require identifying the subsequent lots for processing, specifying their locations on designated equipment, and determining whether it is efficient to adjust doping gas, dose, or energy levels at this juncture. To optimize operations, we impose empirically derived limitations on gas runtimes and adhere to rule-based gas gap restrictions. For instance, after hours of boron implantation, we prioritize recipes with other dopants.

Our scheduling solution interfaces with relevant data sources to extract the information necessary to derive constraints and provide updates to the setup cost matrix. Scheduling constraints encompass machine capacity limitations, equipment dedications, heterogeneous processing times and modeling of lot arrival times. We add to this set of constraints by incorporating our failed tuning predictions as forbidden recipe transitions. The setup cost matrix yields expected tuning durations for switching recipes. As a static baseline, we consider D_{cheby} (see Section 5.2) for equipment-internal reconfiguration, which is also used to favor lot sequences with minimal deviations in dose or energy. Changing species necessitates switching the doping gas, which typically requires significantly more time. Our equipment-condition-aware tuning duration predictions provide more precise updates for recipe transition costs.

After defining the objective function and decision variables for lot allocation within the implantation work center [29, p. 123–125], we calculate the optimal solution using a commercial high-performance optimization solver. The resulting schedule ensures efficient resource utilization, adheres to critical operational requirements, and allows for customizable model objectives, including minimizing makespan, balancing completion times, optimizing equipment usage, reducing setups, minimizing transport costs, ensuring timebound adherence, and avoiding forbidden changes. As a result of the integration of our predictions the OEE of the implantation work center is significantly enhanced, as the equipment spends less time on poor setup attempts.

6 Challenges and Limitations

In our study, we dedicated considerable effort to data preparation to enable efficient training and inference of ML models. Given the large volume of APC data, we implemented a sampling strategy for sensor data from each production lot, creating a more manageable data representation. However, we acknowledge that this method may have excluded signals unique to discarded wafers. We assessed performance using two methods: (1) based on the average feature values of all wafers within a lot, and (2) relying solely on data from the last wafer in each lot. Our analysis found no significant statistical difference between these methods, leading us to choose the latter for its simpler data handling. Labeling required careful tracking of tuning attempts in APC, necessitating a thorough analysis of log files. We merged sensor data from the current lot with the next tuning attempt, a task complicated by disruptions in sequences between processed lots and future tuning attempts, often caused by repair events. If not properly managed, these disruptions could lead to labeling errors. We engaged in extensive discussions with domain experts in the implantation field for feature selection and engineering. While the direct link between the implantation procedure and the corresponding sensor data was well understood, we needed to explore the implications of associating sensor data from the currently processed lot with the upcoming tuning attempt. During the modeling phase, we faced several challenges that required us to revisit the data preparation phase. For instance, we discovered the need to exclude occasional uniformity tunings within a lot, as they could not be avoided by dispatching adaptations and distorted performance metrics. Aligning the inference process with the scheduling integration imposed restrictions, which could be ignored when treating the dataset purely academically. For example, we could only rely on features for prediction that were guaranteed to be available at the inference time of a model. Moreover, since inference takes place asynchronously with our scheduling runs, it was vital to maintain consistent predictions for recipe transitions over time. The constraints on data transfer within our scheduling environment further limited our modeling flexibility, necessitating a reduction in the cardinality of our interface table. It is important to highlight that the volatility of the underlying tuning failure frequency significantly influences our performance metrics. This impact is discernible both at the equipment level and over time.

7 Experimental Results

7.1 *Predicting Tuning Success*

We construct a single model to reduce the influence of volatile tuning failure rates on evaluation metrics. On a monthly basis, we observe a mean $setup\_result$ of $\bar{r} = 0.13 \pm 0.33$. A $setup\_result$ of 0 indicates tuning success, while 1 signals tuning failure.

Apart from adjustments to account for class imbalance, we evaluate different ML models with their default settings. As displayed in Figure 5, two models in their default configuration cross the $F_2 = 0.7$ line on dataset $\mathcal{E}$. Ranked by F_2 this yields: (1) $light_{gbm}$, (2) $hist_{gbm}$, (3) *mlp* and (4) *xgb*. The top pane in Table 2 shows the model performance with default settings, while the bottom pane showcases regularized versions, fine-tuned to mitigate overfitting. For a comprehensive overview of classification metrics, we display F_2-score (F_2), Precision (P), and Recall (R) as recorded on $\mathcal{E}$. High F_2-scores during cross-validation on $\mathcal{T}$ (F_2') qualifies the respective model for hyperparameter. We also include the mean $\bar{F}_2'$ on the test partition, and the difference $\Delta F_2'$ between test and train partitions during cross-validation to indicate the overfitting tendency. Due to the computational expense of training DL models and the employment of overfitting prevention techniques, such as dropout and early stopping, cross-validation is often deemed unpractical. Consequently, we do not provide respective cross-validation metrics. As *rf* expands nodes until all leaves are pure by default, it overfits and is not able to adjust for class imbalance if the maximum depth of a tree is unrestricted. For consistency, we did not resolve this issue by setting additional parameters. Among the DL models, (1) *mlp* achieves the most promising results, (2) *lstm* underperforms, despite its sequence learning capabilities, suggesting minimal temporal dependencies in the data and (3) *gandalf*, designed for tabular data, also yields unsatisfactory results. With $hist_{gbm}$ ranking second-best on the test set twice, and achieving the highest $\bar{F}_2'$ with only slight overfitting in regularized form, we investigate additional aspects of this model.

Confusion Matrix. Table 3 depicts the respective test set confusion matrix. The goal is to predict the largest possible amount of unsuccessful tuning ($\sim 20\%$) correctly, while at the same time keeping false positives low. Although a significant proportion of tunable recipes (21.40%) were misclassified, the benefits gained from potentially avoiding 86% of tuning failures outweigh the negative impact on scheduling flexibility.

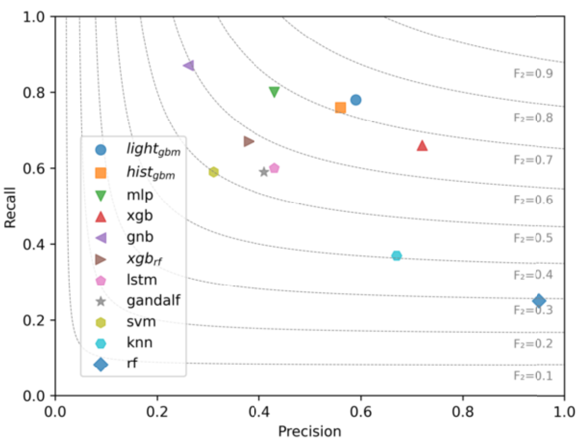

Figure 5. Precision-Recall diagram with F_2 contour lines on $\mathcal{E}$.

Feature Importance and Contribution. To obtain a comprehensive understanding of what $hist_{gbm}$ learned, we utilize SHapley Additive exPlanations (SHAP) [28]. The beeswarm plots, as depicted in Figure 6, illustrate two key aspects: (1) The model's assigned feature importance in descending order, and (2) the correlation between a feature's value and its contribution to the model's output. For instance, the most important feature, *sensor_19*, contains information about the standard deviation of the processing speed from wafer to wafer, which signals instabilities from potentially various components. Our engineered features have a high impact ranking according to SHAP, indicating the model's understanding of the upcoming recipe's significance. Based on the performed expert interviews, we found that examining SHAP values provides a valuable approach for experts to derive insights on enhancing the success ratio of tuning by adapting maintenance protocols. Based on the identified improvement potential and the physical understanding of dependencies, they can focus on specific components. We showcase two examples:

- The beam angle should ideally be perpendicular ($90°$) to the substrate. If there is a deviation ($< 3°$), the respective beam angle parameter is altered to better position the wafer chuck. However, this is only necessary if the beam is still off after being aligned, which indicates issues in upstream components.
- Glitches (short-lived electrical overshoots) are caused by conductive material deposition, and can be addressed temporarily through adjustments, such as reducing field strength and placing the extraction electrode farther from the source. For a more permanent solution, component cleaning or replacement may be necessary.

Analysis shows that SHAP values indicate that the model learned the correct correlations for both above cases, exhibiting a clearly distinguishable color distribution, with higher values contributing towards unsuccessful tuning. Additionally, SHAP waterfall plots aid in troubleshooting recurring issues for specific recipe transitions.

Table 2. Classification metrics: tuning success prediction on $\mathcal{T}$ and $\mathcal{E}$.

Model	regularized	F_2	P	R	$\bar{F}_2'$	$\Delta F_2'$
$light_{gbm}$		**0.74**	0.59	0.78	0.54	-0.20
$hist_{gbm}$		0.71	0.56	0.76	0.54	-0.30
mlp		0.68	0.43	0.80	-	-
xgb		0.67	0.72	0.66	0.50	-0.50
gnb		0.60	0.26	0.87	0.33	-0.05
xgb_{rf}		0.58	0.38	0.67	**0.58**	-0.10
lstm		0.56	0.43	0.60	-	-
gandalf		0.54	0.41	0.59	-	-
svm		0.49	0.31	0.59	0.37	-0.40
knn		0.40	0.67	0.37	0.32	-0.10
rf		0.29	0.95	0.25	0.27	-0.73
$light_{gbm}$	✓	**0.73**	0.50	0.83	0.58	-0.14
$hist_{gbm}$	✓	0.72	0.44	0.86	**0.60**	**-0.07**
rf	✓	0.68	0.52	0.74	0.48	-0.29
xgb	✓	0.67	0.55	0.71	0.50	-0.40
xgb_{rf}	✓	0.57	0.46	0.61	0.53	-0.27

Table 3. Confusion matrix of $hist_{gbm}$ predicting tuning success (0) or tuning failure (1) on $\mathcal{E}$, with $P = 0.44$ and $R = 0.86$.

True \ Pred.	**Positive (1)**	**Negative (0)**	**Total**
Positive (1)	16.83%	2.84%	19.68%
Negative (0)	21.40%	58.92%	80.33%

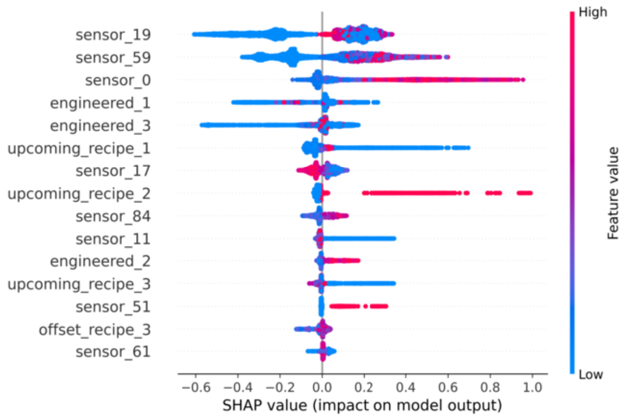

Figure 6. SHAP beeswarm provides insights into classification behavior

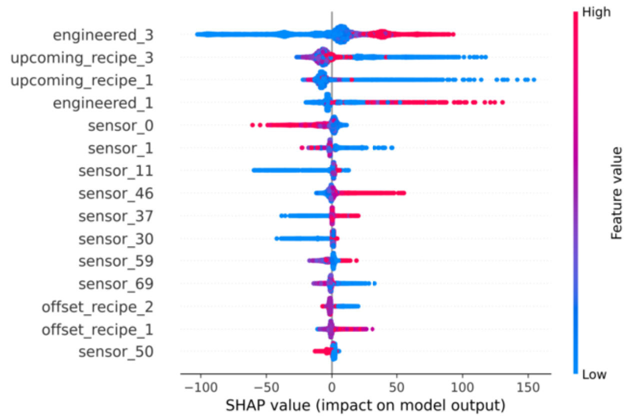

Figure 7. Feature contributions vary in duration regression models.

7.2 *Predicting Tuning Duration*

Scheduling solutions aim to optimize lot sequences based on different optimization criteria, e.g., focusing on minimizing the makespan of certain lots or reducing the idle time of equipment. However, factors like unplanned events and missing contextual data can disrupt these computed ideal sequences. Our tuning duration predictions help fill this contextual data gap, thereby enabling more informed decisions. We evaluate the top-performing ML models from our classification task against two static baselines, using Mean Absolute Error (MAE), Root Mean Squared Error (RMSE) and R^2. These baselines include predicting the mean duration (*mean*) and the Chebyshev distance (D_{cheby}), as outlined in Section 5.2. As displayed in Table 4, $hist_{gbm}$ achieves the highest scores and significantly outperforms both baselines. By incorporating information about equipment conditions and upcoming recipes into ML models, we surpass the *mean* baseline performance. It is important to note, that D_{cheby} acts as a heuristic to guide setup decisions, defaulting to high values for species changes to minimize respective recipe transitions. Despite the dynamic nature of our models, their predictions still exhibit limitations in capturing values at the far right of our distribution. Our attempts to enhance performance by adjusting loss functions to penalize incorrect predictions of prolonged durations do not yield significant improvements. Moreover, logarithmic label scaling results in merely slightly enhanced metrics. To further improve performance metrics, we might (have to) add auxiliary information from yet unexplored data sources.

When inspecting the SHAP beeswarm plot for tuning duration prediction in Figure 7, we observe deviating feature value contributions to the regression model output. These differences are most evident when, comparing *sensor_0* in Figures 6 and 7, indicating an inverse effect based on the task, with high values contributing to a short tuning duration, in contrast to pushing predictions towards tuning fail.

Table 4. Regression metrics for ML and baseline methods on dataset $\mathcal{E}$.

Method	MAE	RSME	R^2
mean	79	126	0.00
D_{cheby}	308	487	-14
$hist_{gbm}$	61	**105**	**0.31**
$light_{gbm}$	**58**	107	0.28
rf	67	110	0.25
xgb	67	110	0.24
xgb_{rf}	66	110	0.23
mlp	72	117	0.13

8 Conclusion

A novel approach for integrating equipment condition-derived constraints into scheduling has been presented for a real-world semiconductor manufacturing scenario. By predicting the ion beam tuning outcome we improve equipment uptime by at least 1 percentage point, yielding more than 100 hours of uptime per equipment annually. We also showed that by informing ML models about the current equipment condition, it is possible to make more precise schedules than solely relying on static heuristics. We developed a sampling strategy to deal with the vast amounts of data generated during our manufacturing processes and subsequently merged our sensor data with corresponding label information. As a result, we provide only the most recent equipment conditions for model training and inference. The most prominent contextual implications of past production operations were added as engineered features. Aspects similar to our engineered boron counter, which indicates the detrimental effect of excessively processing boron recipes on the ion source, could potentially be learned from time series aware models. However, our DL models could not outperform the best tree-based models, despite providing them with more data per prediction. In our study, $hist_{gbm}$ demonstrated superior performance after tuning hyperparameters to improve generalizability by regularization. Additionally, we leveraged Shapley-additive explanations to determine if the model learned plausible dependencies to foster trust. Apart from tuning success prediction, we also investigated ways to predict the tuning duration with moderate success. To better detect prolonged tuning upfront, one could explore the classifying whether the tuning duration would exceed a predefined threshold. The anticipated uptime increase from predicting prolonged tuning is approximately 0.25 percentage points.

To summarize, by integrating additional recipe transition constraints into scheduling, we were able to increase the OEE of our medium current implantation work center by reducing unsuccessful tuning attempts. Regarding profitability, it's more valuable to roll out our solution to comparable equipment , rather than pursuing further improvements in the performance metrics. The application of our developed ML models results in significant uptime improvements, covering the implementation costs by more than 25 times.

Acknowledgements

This project is funded by the European Union's Horizon 2020 research and innovation program under grant agreement number 957407 from 12/2020 through 11/2024.

References

[1] T. Akiba, S. Sano, T. Yanase, T. Ohta, and M. Koyama. Optuna. In *KDD '19*, New York, 2019. Association for Computing Machinery.

[2] Applied Materials. Tuning Materials for Device Optimization: Ion Implant, 2024. URL https://www.appliedmaterials.com/us/en/semiconductor/products/processes/implant.html.

[3] A. Bousdekis, K. Lepenioti, D. Apostolou, and G. Mentzas. Decision Making in Predictive Maintenance: Literature Review and Research Agenda for Industry 4.0. *IFAC-PapersOnLine*, 52(13):607–612, 2019.

[4] Breiman Leo. Random Forests. *Machine Learning*, 45:5–32, 2001.

[5] N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer. SMOTE: Synthetic Minority Over-Sampling Technique. *Journal of Artificial Intelligence Research*, 16:321–357, 2002.

[6] T. Chen and C. Guestrin. XGBoost: A Scalable Tree Boosting System. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '16, pages 785–794, New York, NY, USA, 2016. ACM.

[7] D. Chicco, M. J. Warrens, and G. Jurman. The Matthews Correlation Coefficient (MCC) is More Informative Than Cohen's Kappa and Brier Score in Binary Classification Assessment. *IEEE Access*, 9, 2021.

[8] F. Chollet et al. Keras, 2015. URL https://github.com/fchollet/keras.

[9] A. Cucchetti, D. Distaso, R. Mollica, J. C. Olson, A. Renau, J. T. Scheuer, and D. L. Smatlak. Beam Autotuning on the VIISta 810 Ion Implanter. In *2000 International Conference on Ion Implantation Technology Proceedings. Ion Implantation Technology - 2000 (Cat. No.00EX432)*, pages 452–455. IEEE, 2000.

[10] M. Current, L. M. Rubin, and F. Sinclair. Commercial Ion Implantation Systems. In *Ion Implantation Science and Technology*. Ion Implant Technology Co., 2018.

[11] S. Goeke, K. Staight, and R. Vrijen. Scaling AI in the sector that enables it: Lessons for semiconductor-device makers, 2021.

[12] B. Gori, S. Chang, B. Leavitt, D. Timberlake, C. Krüger, W. Schustereder, M. Welsch, J. Klein, and K. Decker-Lucke. Medium Energy High Dose Ion Implanter. In *2016 21st International Conference on Ion Implantation Technology (IIT)*, pages 1–4, 2016.

[13] S. Hochreiter and J. Schmidhuber. Long Short-Term Memory. *Neural Computation*, 9(8):1735–1780, 1997.

[14] C.-Y. Hsu, Y.-W. Lu, and J.-H. Yan. Temporal Convolution-Based Long-Short Term Memory Network With Attention Mechanism for Remaining Useful Life Prediction. *IEEE Transactions on Semiconductor Manufacturing*, 35(2):220–228, 2022.

[15] A. Jalali, C. Heistracher, A. Schindler, B. Haslhofer, T. Nemeth, R. Glawar, W. Sihn, and P. de Boer. Predicting Time-to-Failure of Plasma Etching Equipment using Machine Learning. In *2019 IEEE International Conference on Prognostics and Health Management (ICPHM)*, pages 1–8. IEEE, 2019.

[16] M. Joseph and H. Raj. GANDALF: Gated Adaptive Network for Deep Automated Learning of Features. URL http://arxiv.org/pdf/2207.08548v6.

[17] A. A. Kalir, S. K. Lo, G. Goldberg, I. Zingerman-Koladko, A. Ohana, Y. Revah, T. B. Chimol, and G. Honig. Leveraging Machine Learning for Capacity and Cost on a Complex Toolset: A Case Study. *IEEE Transactions on Semiconductor Manufacturing*, 36(4):611–618, 2023.

[18] G. Ke, Q. Meng, T. Finley, T. Wang, W. Chen, W. Ma, Q. Ye, and T.-Y. Liu. LightGBM: A Highly Efficient Gradient Boosting Decision Tree. *Advances in Neural Information Processing Systems*, 30, 2017.

[19] D. H. Kim and S. J. Hong. Use of Plasma Information in Machine-Learning-Based Fault Detection and Classification for Advanced Equipment Control. *IEEE Transactions on Semiconductor Manufacturing*, 34 (3):408–419, 2021.

[20] S. H. Kim, C. Y. Kim, H. Da Seol, J. E. Choi, and S. J. Hong. Machine Learning-Based Process-Level Fault Detection and Part-Level Fault Classification in Semiconductor Etch Equipment. *IEEE Transactions on Semiconductor Manufacturing*, 35(2):174–185, 2022.

[21] S. R. Kurakula and J. Trujillo. Data Mining to Detect Ion Source Failures in Varian VIISta Implanters. In *2016 27th Annual SEMI Advanced Semiconductor Manufacturing Conference (ASMC)*, pages 149–150. IEEE, 2016.

[22] A. Laber, M. Gebser, K. Schekotihin, and Y. Yang. Predicting Ion Beam Tuning Success in Semiconductor Manufacturing. In *2022 14th International Conference on Advanced Semiconductor Devices and Microsystems (ASDAM)*, pages 1–4, 2022.

[23] C. I. Lang, A. Jansen, S. Didari, P. Kothnur, and D. S. Boning. Modeling and Optimizing the Impact of Process and Equipment Parameters in Sputtering Deposition Systems Using a Gaussian Process Machine Learning Framework. *IEEE Transactions on Semiconductor Manufacturing*, 35(2):229–240, 2022.

[24] C. I. Lang, R. Sprenkle, E. Wilson, A. Samolov, and D. S. Boning. Intelligent Optimization of Dosing Uniformity in Ion Implantation Systems. *IEEE Transactions on Semiconductor Manufacturing*, 35(3), 2022.

[25] C. I. Lang, F.-K. Sun, B. Lawler, J. Dillon, A. A. Dujaili, J. Ruth, P. Cardillo, P. Alfred, A. Bowers, A. Mckiernan, and D. S. Boning. One Class Process Anomaly Detection Using Kernel Density Estimation Methods. *IEEE Transactions on Semiconductor Manufacturing*, 35(3):457–469, 2022.

[26] J.-H. Lee, H.-J. Kim, Y. Kim, Y. B. Kim, B.-H. Kim, and G.-H. Chung. Machine Learning-Based Periodic Setup Changes for Semiconductor Manufacturing Machines. In *2021 Winter Simulation Conference (WSC)*, pages 1–10. IEEE, 2021.

[27] S.-y. Lin and S.-c. Horng. A Classification-Based Fault Detection and Isolation Scheme for the Ion Implanter. *IEEE Transactions on Semiconductor Manufacturing*, 19(4):411–424, 2006.

[28] S. Lundberg and S.-I. Lee. A Unified Approach to Interpreting Model Predictions, 2017. URL https://arxiv.org/pdf/1705.07874.

[29] L. Mönch, J. W. Fowler, and S. J. Mason. *Production Planning and Control for Semiconductor Wafer Fabrication Facilities*, volume 52. Springer New York, New York, NY, 2013. ISBN 978-1-4614-4471-8.

[30] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine Learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

[31] A. Renau. The Beam Line Architecture of the VIISta 810 Medium Current Ion Implanter. In *1998 International Conference on Ion Implantation Technology. Proceedings (Cat. No.98EX144)*. IEEE, 1998.

[32] T. Saito and M. Rehmsmeier. The Precision-Recall Plot Is More Informative than the ROC Plot When Evaluating Binary Classifiers on Imbalanced Datasets. *PLoS ONE*, 10(3), 2015.

[33] F. Sinclair, J. Olson, D. Rodier, A. Eidukonis, T. Thanigaivelan, and S. Todorov. VIISta 900 3D: Advanced Medium Current Implanter. In *2014 20th International Conference on Ion Implantation Technology (IIT)*, pages 1–4, 2014.

[34] M. Stehli, D. Zschabitz, and T. Jahnig. Bridging the Gap - Integrating APC Constraints and WIP Flow Optimization to Enhance Automated Decision Making in Semiconductor Manufacturing. In *2015 26th Annual SEMI Advanced Semiconductor Manufacturing Conference (ASMC)*, pages 48–52. IEEE, 2015.

[35] G. A. Susto and A. Beghi. Dealing with time-series data in Predictive Maintenance problems. In *2016 IEEE 21st International Conference on Emerging Technologies and Factory Automation (ETFA)*. IEEE, 2016.

[36] G. A. Susto, S. McLoone, D. Pagano, A. Schirru, S. Pampuri, and A. Beghi. Prediction of Integral Type Failures in Semiconductor Manufacturing through Classification Methods. In *2013 IEEE 18th Conference on Emerging Technologies & Factory Automation (ETFA)*, pages 1–4. IEEE, 2013.

[37] G. A. Susto, A. Schirru, S. Pampuri, S. McLoone, and A. Beghi. Machine Learning for Predictive Maintenance: A Multiple Classifier Approach. *IEEE Transactions on Industrial Informatics*, 11(3), 2015.

[38] I. Tomek. Two Modifications of CNN. *IEEE Transactions on Systems, Man, and Cybernetics*, SMC-6(11):769–772, 1976.

[39] G. L. Viviani and M. Falco. Optimal Control of Ion Beam Dynamics. In *Proceedings of 11th International Conference on Ion Implantation Technology*, pages 241–244. IEEE, 1997.

[40] K.-T. Yang, L. Ke, and T. Shen. Modeling and Dispatching Refinement for Implantation to Reduce the Probability of Tuning Beam. In *2015 26th Annual SEMI Advanced Semiconductor Manufacturing Conference (ASMC)*, pages 190–194. IEEE, 2015.

[41] W.-T. Yang, J. Blue, A. Roussy, M. Reis, and J. Pinaton. Advanced Run-to-Run Controller in Semiconductor Manufacturing with Real-Time Equipment Condition. In *2018 29th Annual SEMI Advanced Semiconductor Manufacturing Conference (ASMC)*. IEEE, 2018.

[42] H.-M. Yu, C.-C. Lin, M.-H. Hsu, Y.-T. Chen, K.-W. Chen, T. Luoh, L.-W. Yang, T. Yang, and K.-C. Chen. CMP Process Optimization Engineering by Machine Learning. *IEEE Transactions on Semiconductor Manufacturing*, 34(3):280–285, 2021.

[43] J. Yu and P. Guo. Run-to-Run Control of Chemical Mechanical Polishing Process Based on Deep Reinforcement Learning. *IEEE Transactions on Semiconductor Manufacturing*, 33(3):454–465, 2020.

[44] C. Yugma, J. Blue, S. Dauzère-Pérès, and V. P. Integration of Scheduling and Advanced Process Control in Semiconductor Manufacturing: Review and Outlook. In *2014 IEEE International Conference on Automation Science and Engineering (CASE)*, pages 93–98, 2014.