

Wafer Defect Root Cause Analysis with Partial Trajectory Regression

DM: Big Data Management and Machine Learning

Kohei Miyaguchi*
IBM Research – Tokyo
Tokyo, Japan
koheimiyaguchi@gmail.com

Masao Joko
IBM Semiconductors
Tokyo, Japan
mjoko@jp.ibm.com

Rebekah Sheraw
IBM Semiconductors
Albany, NY, USA
rebekah.sheraw@ibm.com

Tsuyoshi Idé
IBM Semiconductors
Yorktown Heights, NY, USA
tide@us.ibm.com

Abstract—Identifying upstream processes responsible for wafer defects is challenging due to the combinatorial nature of process flows and the inherent variability in processing routes, which arises from factors such as rework operations and random process waiting times. This paper presents a novel framework for wafer defect root cause analysis, called Partial Trajectory Regression (PTR). The proposed framework is carefully designed to address the limitations of conventional vector-based regression models, particularly in handling variable-length processing routes that span a large number of heterogeneous physical processes. To compute the attribution score of each process given a detected high defect density on a specific wafer, we propose a new algorithm that compares two counterfactual outcomes derived from partial process trajectories. This is enabled by new representation learning methods, *proc2vec* and *route2vec*. We demonstrate the effectiveness of the proposed framework using real wafer history data from the NY CREATES fab in Albany.

I. INTRODUCTION

Identifying the root causes of wafer defects is one of the most critical yet challenging tasks in semiconductor manufacturing. Several unique factors contribute to this complexity, including the intricate dependencies between processes, the heterogeneity of individual operations and recipes, and the dynamic variations in processing routes.

Particularly during process integration and yield ramp-up stages, classical design-of-experiment methodologies, which analyze process outcomes across systematically varied parameters, are often impractical due to too many adjustable parameters. Moreover, although a wide range of off-the-shelf machine learning (ML) tools are publicly available, most of these tools are designed for prediction tasks, such as estimating real-valued outputs (regression) or categorical outcomes (classification). As a result, these tools are poorly suited to the task of root cause analysis, and fab-wide defect diagnosis still heavily depends on manual, ad hoc analysis by domain experts.

To enable more systematic wafer defect root-cause analysis, two major directions have been pursued in the literature in semiconductor analytics to date. The first approach treats defect diagnosis as a by-product of cross-process virtual metrol-

A: Prediction using a partial trajectory **not** including k

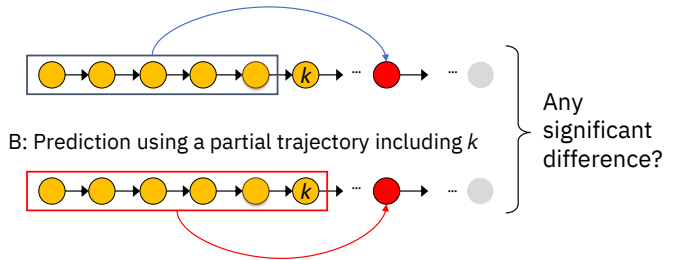


Fig. 1. Causal analysis framework with partial trajectory regression (PTR).

ogy (VM) modeling. Regularized linear regression combined with variable selection techniques is commonly used (e.g., [1]–[3]). However, linear models struggle to capture complex nonlinear relationships across heterogeneous fabrication processes. Furthermore, defect attribution based on linear models is essentially reduced to variable-wise correlation analysis, which is known to yield only weak attribution signals [4]. While recurrent neural networks (RNNs) and Transformers can capture complex nonlinear dependencies in sequential processes (e.g., [5]–[9]), they operate as black boxes, making input attribution a non-trivial task.

The second approach involves leveraging explainable artificial intelligence (XAI) techniques applicable to black-box prediction models. This may be a promising direction in that it can potentially enhance expressive prediction models with interpretability, helping to identify which process steps should be adjusted to improve defect rates. However, for cross-process defect attribution, existing methods such as those based on Shapley values (e.g., [10]–[13]) fail to reflect key characteristics of semiconductor processes. Limitations include a disregard for the sequential nature of fabrication and dependence on arbitrarily selected baseline inputs. Ironically, these methods attempt to explain black-box models by introducing another layer of black-box reasoning.

Those two technical directions highlight two main technical tasks in data-driven root-cause analysis. The first task is to predict process outcomes (e.g., defect density), given a

* Kohei Miyaguchi is currently affiliated with LY Research, Japan.

sequence of physical processes. The second task is to provide an explanation to an observed pair of the process input and its outcome. We refer the first and second tasks the forward problem (prediction) and inverse problems (attribution), respectively. The main technical difficulty of wafer defect root-cause analysis is that these do not fit the standard problem settings of machine learning.

Specifically, *first*, process routes vary in length and tools used, making it difficult to represent inputs as fixed-dimensional vectors—a key assumption of many algorithms like random forests or gradient boosting. Related to this point, *second*, each process step may involve completely different physical and chemical mechanisms, making it extremely difficult to extract features that are comparable across steps. *Third*, most off-the-shelf prediction and attribution tools ignore sequential structure; for example, linear regression and Shapley values assume unordered feature vectors. *Finally*, many XAI algorithms make assumptions that may not be appropriate in semiconductor manufacturing. Examples include the dependency on baseline inputs of the Shapley values.

To overcome these significant challenges, we propose a novel framework called the partially trajectory regression (PTR) for cross-process defect root-cause analysis. We formalize the prediction task as trajectory regression [14], [15], rather than conventional vector-based regression, to accommodate variable-length sequences. To handle process heterogeneity along processing routes, we introduce `proc2vec` and `route2vec`, two new representation learning approaches for encoding process trajectories, designed for effective learning even under limited sample sizes. Finally, as illustrated in Fig. 1, we propose a novel attribution algorithm based on the new idea of comparing the outcomes between two *partial* trajectories—one including the target process and the other excluding it. We demonstrate its effectiveness on real wafer history data for the state-of-the-art FEOL (front end-of-line) process from the NY CREATES fab in Albany.

II. RELATED WORK

There is a growing body of research on data-driven root cause analysis (RCA) methods in manufacturing. In a recent systematic review, Pietsch et al. [16] argue that traditional knowledge-based approaches have suffered from challenges in systematically acquiring domain knowledge. While data-driven approaches effectively mitigate the knowledge acquisition bottleneck, they still face methodological limitations, particularly in evaluating the causal relevance of variables in high-dimensional settings. This work aims to address this critical issue in data-driven RCA.

A. Handling Process Heterogeneity

A key characteristic that distinguishes semiconductor manufacturing from other industrial domains is its process complexity. A typical semiconductor process involves hundreds of intricate and highly specialized physical operations, including photolithography, thermal annealing, polishing, wet and

dry etching, ion implantation, electroplating, sputtering, and chemical vapor deposition, among others.

For cross-process RCA, these heterogeneous operations must be mapped to a shared representation space to enable meaningful comparisons. Existing literature offers three general approaches for this. The first approach utilizes process trace data [17], [18], such as temperature and pressure recorded from processing chambers. While effective within individual process tools, this method requires extensive tool-specific preprocessing, and the quality of analysis heavily depends on the chosen preprocessing strategy, making it less suitable for cross-process analysis. The second approach uses inline measurements as proxies for physical processes. Since these measurements partially absorb the physical heterogeneity across processes, this method has become common practice in recent studies [11], [13], [19], [20]. However, these approaches typically disregard the sequential order of processes and perform RCA as a by-product of virtual metrology (VM), often via univariate correlation analysis, which is known to yield only weak attribution signals [4].

The third approach involves embedding techniques, where data objects (e.g., process steps) are transformed into numerical vector representations. This idea, widely adopted in natural language processing (NLP), such as in Word2Vec [21] or token embeddings in Transformer models [22], treats individual processes as tokens and process sequences as sentences. Embedding allows process similarity to be learned from data, enabling richer representations. Recently, embedding techniques have been introduced in the semiconductor context. For instance, Fan et al. [23] use one-hot encoding to unify categorical and numerical data in the VM setting. Schulz et al. [24] propose defining a fab state vector using known interdependencies among processing tools under an unsupervised setting, without the context of wafer defect analysis.

The proposed `proc2vec` and `route2vec` fundamentally differ from those recent works in that they incorporate the similarity between related tools and recipes via the kernel function. This is critical in process integration stage since many similar but different recipes are tested with multiple tools and chambers of the same type. Naive one-hot encoding amounts to treating them as independent objects, which drastically deteriorates sample efficiency.

B. Explainable AI (XAI)

Numerous methods have been developed to improve the interpretability of machine learning models under the umbrella of explainable AI (XAI) [25]. One widely used category is additive explanation methods [26], [27], which provide mathematically justified decompositions of a model’s output into individual contributions of input variables. Other common XAI techniques include gradient-based methods [28] and attention-based methods [29].

Several XAI methods have been proposed for sequential input variables, such as wafer processing routes [30]. The integration of model-agnostic XAI techniques with advanced quality prediction models has become a major research trend

in semiconductor analytics, aiming to balance predictive power with interpretability. Among various XAI methods (see, e.g., [31] for an overview), the Shapley value (SV) is the most widely used in semiconductor analytics [10]–[13].

Ironically, despite its widespread adoption, most studies apply XAI methods in a black-box fashion, without scrutinizing their modeling assumptions. In fact, SV disregards both the sequential nature of semiconductor processes and the wafer-lot hierarchy. Additionally, mainstream attribution algorithms such as the Shapley values, integrated gradients, and local linear surrogate models are deviation agnostic [32], meaning they do not account for how much a given input deviates from a normal or baseline process state, making them less suitable for defect diagnosis tasks.

III. PROBLEM SETTING

As discussed, there are two major machine learning tasks towards wafer defect root cause analysis. The first task is to learn a prediction model for the occurrence of a certain defect type, while the second task is to establish a scoring model for defect attribution, given the learned prediction model.

Prediction modeling: We formalize the prediction task as trajectory regression for defect density. Specifically, we assume to have a training dataset including N pairs of a defect density value y and a process trajectory ξ :

$$\mathcal{D}_{\text{train}} := \{(y^{(n)}, \xi^{(n)})\}_{n=1}^N, \quad (1)$$

where the superscript (n) denotes that the quantity belongs to the n -th wafer. A process trajectory ξ is a sequence of process attributes and timestamps as

$$\xi = ((x_1, t_1), \dots, (x_L, t_L)), \quad (2)$$

where L is the number of process steps included in ξ . The goal of the prediction task is to find a function $f(\xi)$ such that it gives a prediction of defect density for a given process trajectory ξ that is not included in $\mathcal{D}_{\text{train}}$ in general.

Attribution modeling: On the other hand, the attribution task is to find a function $\alpha_k(\xi)$, which computes the responsibility score for the k -th process on the process trajectory ξ , possibly making use of a pre-trained prediction function $f(\xi)$.

Considerations in modeling: Before delving into the technical details, we highlight the unique characteristics of semiconductor analytics. Although Eq. (1) might have given the impression that the dataset comprises independent samples, wafers are not generally independent due to lot- and batch-based processing. Additionally, in the process integration and yield ramp-up stage, multiple processing recipes are tested simultaneously. Hence, some trajectories can be quite similar except for a few processes.

Although the nominal diversity of processing routes can be extremely large, the *effective sample size*—defined as the number of wafers per unique processing or measurement condition—can be very small, potentially even as low as one. Nonetheless, the net variety of trajectories may be much smaller than their nominal diversity. Thus, leveraging similarities between tools and recipes is key to successful modeling.

The `proc2vec` embedding approach described in the next section is designed to achieve that goal.

IV. PROCESS EMBEDDING

The proposed partial trajectory regression (PTR) framework covers both prediction and attribution problems. The embedding module featuring `proc2vec` is used by the prediction and attribution function as a common building block.

Now, let us take a look at how it works. The `proc2vec` algorithm generates a vector representation for each process using a kernel embedding technique. In our setting, we assume that high-level process attributes, such as process ID and recipe ID are available from the manufacturing execution system (MES) but detailed tool traces are unavailable at least in some tools. hence, for cross-process attribution, we use only high-level attributes as the first attempt.

First, we pull common attributes out of the MES across different processes, such as equipment IDs, recipe IDs, tool types, photo layer IDs, route IDs, and others. We then create a single synthetic ‘token’ for each process by concatenating those strings as:

$$(\text{process token}) = \text{eqp} \oplus \text{recipe} \oplus \text{tool_type} \oplus \text{photo_layer} \oplus \text{route} \oplus \dots, \quad (3)$$

where $\oplus$ denotes concatenation with a suitable separator.

Considering the challenges of low effective sample size, we employ a relatively simple embedding approach. Specifically, based on the string representation, we first construct a dictionary of the tokens. Let V_d is the size of the vocabulary, i.e., the number of unique tokens. For the dictionary, we compute the kernel matrix $K \in \mathbb{R}^{V_d \times V_d}$, where $K_{i,j}$ is the similarity between token i and token j .

Our preliminary analysis indicated that a custom variant of the substring kernel [33], [34] effectively captured the clustering structure of processes. Once the kernel matrix K is computed, the vector representation of token i is obtained as:

$$\mathbf{x}_i = (\sqrt{\lambda_1} v_i^{(1)}, \dots, \sqrt{\lambda_k} v_i^{(k)}, \dots, \sqrt{\lambda_D} v_i^{(D)})^\top, \quad (4)$$

where λ_k is the k -th largest eigenvalue of K , and $v_i^{(k)}$ is the i -th element of the eigenvector corresponding to λ_k . D is the embedding dimensionality as a hyperparameter.

V. TRAJECTORY REGRESSION

Now that processes have been converted into D -dimensional vectors, we consider developing a predictive model for defect density y as a function of the sequence of process vectors. This remains a non-trivial task due to the variable length of trajectories, making it an instance of *trajectory regression*, distinct from standard regression problems.

In trajectory regression modeling, a module called the `route2vec` plays the key role. It converts the entire trajectory, represented as an ordered set of `proc2vec` vectors, into another vector representation. We achieve this by introducing

a constrained version of recurrent neural network (C-RNN), where each cell operates as:

$$z_k = \text{Cell}(z_{k-1}, x_k, t_k), \quad (5)$$

where z_k is the vector representation of the *partial trajectory* up to the k -th process from the first. Hence, the vector representation of the full trajectory of $\xi^{(n)}$ is obtained as $z^{(n)} := z_{L^{(n)}}^{(n)}$. We refer to this mapping from a (partial or full) trajectory to its corresponding vector representation z_k ($k = 1, \dots, L$) as `route2vec`.

Our preliminary experiments showed that off-the-shelf RNN models did not yield promising results, as expected due to the small effective sample size. Instead, we propose a custom cell architecture:

$$z_k = \psi(t_k, t_{k-1})x_k + z_{k-1}, \quad (6)$$

where $\psi(\cdot, \cdot)$ is a temporal mapping function, set to $\log_{10}(1 + \cdot)$ in our evaluation. Extending Eq. (6) to more expressive models such as multi-layer perceptrons (MLP) is straightforward; however, we leave further analysis for future work.

The regression module defines a function to predict y from ξ . Since `route2vec` converts trajectories of varying lengths into fixed-dimension vectors, the prediction function can be formulated as:

$$f(z_k) = \text{MLP}(z_k), \quad (7)$$

where MLP denotes a multi-layer perceptron. We deliberately abused the notation for $f(\cdot)$ so that it admits the vector representation of not only a full trajectory but also partial trajectories. For the specific implementation, we used a fully connected network without hidden layers, defining a linear prediction function, in our empirical evaluation. The function $f(\cdot)$ is trained to minimize a properly regularized squared loss, such as

$$L := \frac{1}{2N} \sum_{n=1}^N \sum_{k=1}^{L^{(n)}} \frac{1}{L^{(n)}} \left(y^{(n)} - f(z_k) \right)^2 + \nu \|\theta\|_1, \quad (8)$$

where θ is the model parameters in $f(\cdot)$ and $\|\cdot\|_1$ is the ℓ_1 -norm. ν is the regularization strength and is treated as a hyperparameter. The inner summation represents data augmentation for numerical stability. The details on training are left to a longer version of the paper in preparation.

VI. ATTRIBUTION USING PARTIAL TRAJECTORY REGRESSION

One of the most important features of the PTR framework is its ability to admit *partial* trajectories as input. To quantify the influence of the k -th process, we ask: Is there a significant difference in the prediction outcome between the partial trajectories z_k and z_{k-1} ? This comparison between two *counterfactual* inputs measures the potential outcome when including process k .

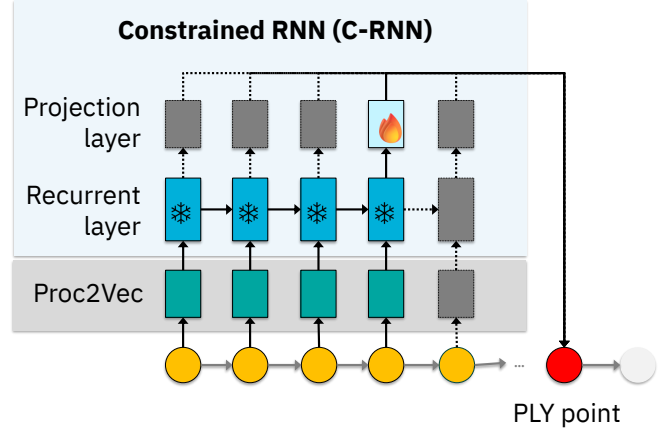


Fig. 2. System architecture of the PTR framework. Only the projection layer is learnable in this architecture to ensure numerical stability under limited sample sizes.

Following Rubin’s potential outcome framework [35], the attribution module quantifies the *causal intervention* of process k as:

$$\alpha_k(\xi) = f(z_k) - f(z_{k-1}). \quad (9)$$

This defines the attribution score of the k -th process in ξ . The PTR attribution score satisfies the following additive property:

Theorem VI.1 (Additive Property of PTR). *The PTR attribution score satisfies the additive property:*

$$\sum_{l=1}^k \alpha_l = f(z_k) - f(z_0). \quad (10)$$

Proof. This follows directly from summing individual contributions:

$$\begin{aligned} \sum_{l=1}^k \alpha_l &= [f(z_k) - f(z_{k-1})] + [f(z_{k-1}) - f(z_{k-2})] \\ &\quad + \dots + [f(z_1) - f(z_0)] = f(z_k) - f(z_0). \end{aligned}$$

□

This additive property significantly enhances the interpretability of the attribution score, as demonstrated in the cumulative attribution plots in the next section.

Figure 2 illustrates the overall architecture of the PTR framework. While we have presented a simple version of the model with the embedding (Eq. (4)) and the recurrent layer (Eq. (6)) fixed and ‘frozen’ during the training, extending to a more general architecture is straightforward. Further analysis along this direction is left for future work.

VII. EMPIRICAL EVALUATION

We applied the PTR framework to conduct a root cause analysis (RCA) for a specific defect type in a state-of-the-art front-end-of-line (FEOL) process at the NY CREATES Albany NanoTech fab. The dataset consisted of process histories for

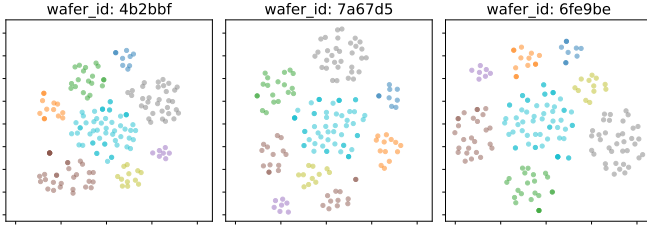


Fig. 3. Distribution of the `proc2vec` embeddings for three wafer samples, visualized with *t*-SNE (the perplexity is set to be 30). The color coding corresponds to distinction among the equipment types including Wet Process, Rapid Thermal Processing, Inspection, Lithography, Reactive Ion Etching, Ion Implantation, Furnace, Chemical Mechanical Polishing, although the exact mapping as well as axis scales are undisclosed.

$N = 787$ wafers, covering hundreds of processes, along with corresponding defect density measurements obtained at a process-limited yield (PLY) evaluation point.

First, we validated the embedding module by comparing three scenarios: 1) $x_i = 1$ (a constant value), 2) x_i represented as a one-hot vector, and 3) the proposed kernel embedding approach. The correlation coefficients with defect density y were 0.27, 0.52, and 0.61, respectively, demonstrating the effectiveness of the kernel embedding technique.

Figure 3 shows the distribution of the resulting embeddings, where distances between processes reflect the similarity of tools and recipes. The visualization was performed with `scikit-learn`'s implementation of *t*-SNE (*t*-distributed stochastic neighbor embedding) [36]. The figure exhibits a clear clustering structure according to equipment types, which suggests that the string kernel approach effectively captures inter-process similarities.

Next, we evaluated the regression module. While the linear model used in Eq. (7) provided moderate predictive performance, the correlation coefficient improved to 0.87 when incorporating additional process features and optimizing the prediction model. Further details are omitted here for brevity.

Finally, Fig. 4 presents the cumulative attribution score, which plots

$$\sum_{i=1}^k \alpha_i + f(z_0) \quad (11)$$

for a specific wafer from the held-out dataset at each timestamp τ , where $t_k \leq \tau$. The plot illustrates the progression from $f(z_0)$ (intercept) to $f(z)$, the final prediction based on the wafer's complete trajectory. Notable jumps, labeled A and B, are highlighted in the plot. Upon further inspection, these jumps correspond to unusually long waiting times at certain tools, suggesting potential root causes of high defect density. This example demonstrates how PTR effectively identifies problematic processes, providing actionable insights for RCA.

VIII. CONCLUSION

We have proposed a novel data-driven framework for root cause analysis, termed Partial Trajectory Regression (PTR).

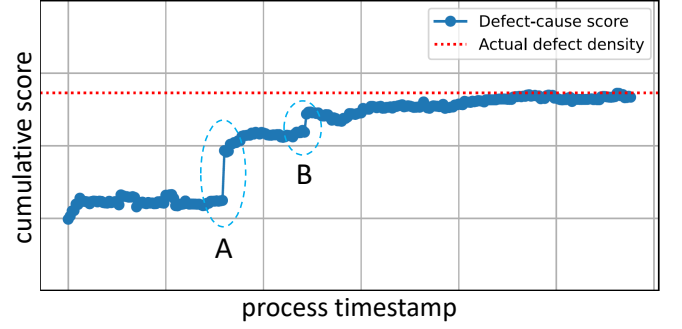


Fig. 4. Cumulative attribution score (“defect-cause score”) for a specific wafer with high defect density, highlighting potential problematic processes at points A and B.

The PTR framework consists of three key modules: Embedding, Regression, and Attribution. In the embedding module, we introduced `proc2vec`, a novel process representation approach based on string-level similarity. For the regression module, we proposed a unique trajectory regression framework that enables prediction from partial process trajectories. In the attribution module, we demonstrated that the proposed attribution scoring model satisfies the additive property and allows interpretations based on Rubin’s causal model.

Empirical evaluation of PTR on a state-of-the-art FEOL process at the NY CREATES Albany NanoTech fab demonstrated its effectiveness in identifying potential sources of wafer defects, providing actionable insights for process optimization.

ACKNOWLEDGEMENT

The authors gratefully acknowledge the support of NY CREATES and the Albany NanoTech Complex for providing access to state-of-the-art fabrication and characterization resources. They also extend their gratitude to Dr. Monirul Islam and Dr. Ishtiaq Ahsan for providing the PLY data and their valuable support throughout the project.

REFERENCES

- [1] G. A. Susto, S. Pampuri, A. Schirru, A. Beghi, and G. De Nicola, “Multi-step virtual metrology for semiconductor manufacturing: A multilevel and regularization methods-based approach,” *Computers & Operations Research*, vol. 53, pp. 328–337, 2015.
- [2] M. A. Jebri, E. El Adel, G. Graton, M. Ouladsine, and J. Pinaton, “Virtual metrology on semiconductor manufacturing based on just-in-time learning,” *IFAC-PapersOnLine*, vol. 49, no. 12, pp. 89–94, 2016.
- [3] K.-J. Kim, K.-J. Kim, C.-H. Jun, I.-G. Chong, and G.-Y. Song, “Variable selection under missing values and unlabeled data in semiconductor processes,” *IEEE Transactions on Semiconductor Manufacturing*, vol. 32, no. 1, pp. 121–128, 2018.
- [4] K. Miyaguchi, M. Joko, R. Sheraw, and T. Idé, “Sequence-aware inline measurement attribution for good-bad wafer diagnosis,” in *2025 SEMI Advanced Semiconductor Manufacturing Conference (ASMC)*, p. TBD, IEEE, 2025.
- [5] J. Yella, C. Zhang, S. Petrov, Y. Huang, X. Qian, A. A. Minai, and S. Bom, “Soft-sensing conformer: A curriculum learning-based convolutional transformer,” in *2021 IEEE International Conference on Big Data (Big Data)*, pp. 1990–1998, IEEE, 2021.
- [6] S. Han, et al., “Deep learning-based virtual metrology in multivariate time series,” in *2023 IEEE International Conference on Prognostics and Health Management (ICPHM)*, pp. 30–37, IEEE, 2023.

- [7] F. Dalla Zuanna, N. Gentner, and G. A. Susto, "Deep learning-based sequence modeling for advanced process control in semiconductor manufacturing," *IFAC-PapersOnLine*, vol. 56, no. 2, pp. 8744–8751, 2023.
- [8] K. B. Lee and C. O. Kim, "Recurrent feature-incorporated convolutional neural network for virtual metrology of the chemical mechanical planarization process," *Journal of Intelligent Manufacturing*, vol. 31, no. 1, pp. 73–86, 2020.
- [9] C.-Y. Hsu and Y.-W. Lu, "Virtual metrology of material removal rate using a one-dimensional convolutional neural network-based bidirectional long short-term memory network with attention," *Computers & Industrial Engineering*, vol. 186, p. 109701, 2023.
- [10] J. A. Torres, I. Kissiov, M. Essam, C. Hartig, R. Gardner, K. Jantzen, S. Schueler, and M. Niehoff, "Machine learning assisted new product setup," in *2020 31st Annual SEMI Advanced Semiconductor Manufacturing Conference (ASMC)*, pp. 1–5, IEEE, 2020.
- [11] J. Senoner, T. Netland, and S. Feuerriegel, "Using explainable artificial intelligence to improve process quality: evidence from semiconductor manufacturing," *Management Science*, vol. 68, no. 8, pp. 5704–5723, 2022.
- [12] Y. Lee and Y. Roh, "An expandable yield prediction framework using explainable artificial intelligence for semiconductor manufacturing," *Applied Sciences*, vol. 13, no. 4, p. 2660, 2023.
- [13] P. Guo and Y. Chen, "Enhanced yield prediction in semiconductor manufacturing: Innovative strategies for imbalanced sample management and root cause analysis," in *2024 IEEE International Symposium on the Physical and Failure Analysis of Integrated Circuits (IPFA)*, pp. 1–6, IEEE, 2024.
- [14] T. Idé and S. Kato, "Travel-time prediction using gaussian process regression: A trajectory-based approach," in *Proceedings of the 2009 SIAM International Conference on Data Mining*, pp. 1185–1196, SIAM, 2009.
- [15] T. Idé and M. Sugiyama, "Trajectory regression on road networks," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 25, pp. 203–208, 2011.
- [16] D. Pietsch, M. Matthes, U. Wieland, S. Ihlenfeldt, and T. Munkelt, "Root cause analysis in industrial manufacturing: A scoping review of current research, challenges and the promises of ai-driven approaches," *Journal of Manufacturing and Materials Processing*, vol. 8, no. 6, p. 277, 2024.
- [17] H.-W. Xu, Q.-H. Zhang, Y.-N. Sun, Q.-L. Chen, W. Qin, Y.-L. Lv, and J. Zhang, "A fast ramp-up framework for wafer yield improvement in semiconductor manufacturing systems," *Journal of Manufacturing Systems*, vol. 76, pp. 222–233, 2024.
- [18] S.-K. S. Fan, C.-Y. Hsu, D.-M. Tsai, M. C. Chou, C.-H. Jen, and J.-H. Tsou, "Key feature identification for monitoring wafer-to-wafer variation in semiconductor manufacturing," *IEEE Transactions on Automation Science and Engineering*, vol. 19, no. 3, pp. 1530–1541, 2022.
- [19] S. Wang and Y. Chen, "Improved yield prediction and failure analysis in semiconductor manufacturing with xgboost and shapley additive explanations models," in *2024 IEEE International Symposium on the Physical and Failure Analysis of Integrated Circuits (IPFA)*, pp. 01–08, IEEE, 2024.
- [20] T. Ni, W. Rui, C. Zhuo, Y. Li, X. Wen, and M. Nie, "A novel approach to reducing testing costs and minimizing defect escapes using dynamic neighborhood range and shapley values," *ACM Transactions on Design Automation of Electronic Systems*, 2025.
- [21] T. Mikolov, "Efficient estimation of word representations in vector space," *arXiv preprint arXiv:1301.3781*, 2013.
- [22] A. Vaswani, "Attention is all you need," *Advances in Neural Information Processing Systems*, 2017.
- [23] S.-K. S. Fan, W.-K. Lin, and C.-H. Jen, "Data-driven optimization of accessory combinations for final testing processes in semiconductor manufacturing," *Journal of Manufacturing Systems*, vol. 63, pp. 275–287, 2022.
- [24] B. Schulz, C. Jacobi, A. Gisbrecht, A. Evangelos, C. W. Chan, and B. P. Gan, "Graph representation and embedding for semiconductor manufacturing fab states," in *2022 Winter Simulation Conference (WSC)*, pp. 3382–3393, IEEE, 2022.
- [25] F. Xu, H. Uszkoreit, Y. Du, W. Fan, D. Zhao, and J. Zhu, "Explainable AI: A brief survey on history, research areas, approaches and challenges," in *Proceedings of the 8th cCF international conference on Natural language processing and Chinese computing, part II* 8, pp. 563–574, 2019.
- [26] S. M. Lundberg and S.-I. Lee, "A unified approach to interpreting model predictions," in *Proceedings of the 31st International Conference on Neural Information Processing Systems*, p. 4768–4777, 2017.
- [27] M. T. Ribeiro, S. Singh, and C. Guestrin, "'Why Should I Trust You?': Explaining the predictions of any classifier," in *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '16, (New York, NY, USA), p. 1135–1144, 2016.
- [28] R. R. Selvaraju, M. Cogswell, A. Das, R. Vedantam, D. Parikh, and D. Batra, "Grad-cam: visual explanations from deep networks via gradient-based localization," *International journal of computer vision*, vol. 128, pp. 336–359, 2020.
- [29] A. Ali, T. Schnake, O. Eberle, G. Montavon, K.-R. Müller, and L. Wolf, "Xai for transformers: Better explanations through conservative propagation," in *International Conference on Machine Learning*, pp. 435–451, PMLR, 2022.
- [30] T. Rojat, R. Puget, D. Filliat, J. Del Ser, R. Gelin, and N. Díaz-Rodríguez, "Explainable artificial intelligence (XAI) on timeseries data: A survey," *arXiv preprint arXiv:2104.00950*, 2021.
- [31] C. Molnar, *Interpretable machine learning*. Lulu.com, 2020.
- [32] T. Idé and N. Abe, "Generative perturbation analysis for probabilistic black-box anomaly attribution," in *Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pp. 845–856, 2023.
- [33] H. Lodhi, C. Saunders, J. Shawe-Taylor, N. Cristianini, and C. Watkins, "Text classification using string kernels," *Journal of machine learning research*, vol. 2, no. Feb, pp. 419–444, 2002.
- [34] J. Shawe-Taylor and N. Cristianini, *Kernel methods for pattern analysis*. Cambridge university press, 2004.
- [35] D. B. Rubin, "Causal inference using potential outcomes: Design, modeling, decisions," *Journal of the American statistical Association*, vol. 100, no. 469, pp. 322–331, 2005.
- [36] L. Van der Maaten and G. Hinton, "Visualizing data using *t*-SNE," *Journal of machine learning research*, vol. 9, no. 11, 2008.