I&EC research
Industrial & Engineering Chemistry Research
pubs.acs.org/IECR
Article

Data-Driven Machine Learning Predictor Model for Optimal Operation of a Thermal Atomic Layer Etching Reactor

Published as part of Industrial & Engineering Chemistry Research special issue "AI/ML in Chemical Engineering".

Henrik Wang, Feiyang Ou, Julius Suherman, Matthew Tom, Gerassimos Orkoulas, and Panagiotis D. Christofides*

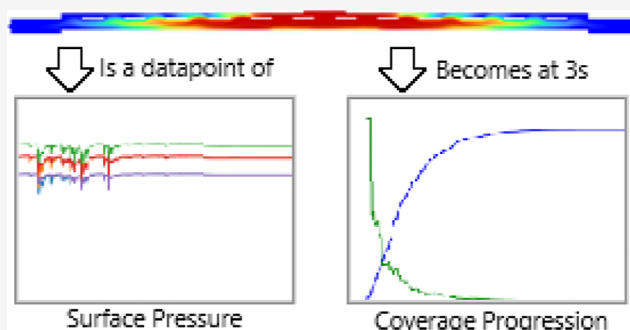
Cite This: *Ind. Eng. Chem. Res.* 2024, 63, 19693−19706

Read Online

ACCESS | Metrics & More | Article Recommendations

Downloaded via UNIV OF CALIFORNIA LOS ANGELES on November 21, 2024 at 15:05:33 (UTC).
See https://pubs.acs.org/sharingguidelines for options on how to legitimately share published articles.

ABSTRACT: As semiconductor devices continue to shrink to nanoscale dimensions and take on increasingly complex geometries, novel fabrication processes and techniques must emerge in response. One such technique is atomic layer etching (ALE), which uses half-reactions to etch away monolayers of substrate per cycle. Due to the self-limiting nature of the composite half-reactions of ALE, the reaction progresses in a nonlinear fashion, making it difficult to precisely forecast when the reaction has reached completion. This work aims to construct a predictor model based on simulated data that determines the etch rate of any ALE process given the real-time wafer surface pressure data. This model is trained on a machine learning transformer structure, which is commonly employed in natural language processing, as the transformer structure is well-suited for handling large amounts of time-series/sequence data. To determine how to create the best predictor model, both data aggregation and data set selection heuristics are explored. The resulting models indicate that, for a sufficiently controlled process, it is not optimal to aggregate all available data sets. Rather, data sets must be carefully selected with a heuristic to yield the optimal ALE predictor model.



INTRODUCTION

The past two decades has observed technological innovations in high-performance electronics that possess favorable characteristics including reduced feature sizes, computing speed, and energy efficiency.[1] However, this continued advancement is facing hurdles attributed to a vital and necessary component of electronic devices: semiconductors. Per Moore's law, a semiconductor chip will improve in performance by strategic designing and downscaling of transistors that facilitate stacking and densification;[2] however, the manufacturing of these products is time-consuming and inconsistent at maintaining overall quality in nanoscale dimensions. In terms of productivity, the overall semiconductor fabrication process comprises around 500 processing steps from raw materials to the finished product that are exclusive to the product material and design.[3] Additionally, some components of the finished product have stringent design criteria.[4] For example, the nanowire, a critical component of transistors that enables current transport between the source and drain of complementary metal-oxide transistor technology,[5,6] demands oxide film thicknesses in the nanoscale. These nanoscale dimensions are reproducible through sequential cycles of thin-layer deposition and etching processes that are intended to add or remove monolayers of substrate material by exhibiting self-limiting characteristics. However, it is difficult to streamline a high quality fabrication method for this process at an industrial setting due to the inherent limitations in process- and operation-dependent techniques. One way to help meet the growing demand for both semiconductor chip throughput and quality is by developing models that can accurately predict the process quality as a function of real-time process data.

Digital twin modeling is an effective tool intended to replicate real-world processes through computational modeling with feedback validation to ensure the efficacy of the model. For example, Shao et al. (2019) discussed the prevalence of digital twin modeling in the semiconductor manufacturing

Received: August 21, 2024
Revised: October 22, 2024
Accepted: October 25, 2024
Published: October 30, 2024

© 2024 American Chemical Society

https://doi.org/10.1021/acs.iecr.4c03150
Ind. Eng. Chem. Res. 2024, 63, 19693−19706

industry with applications to smart manufacturing, which enables process optimization and advanced process control by extending data sets across wider ranges of operating conditions.[7] Besides a single process, Moyne et al. (2020) hypothesized the potential for expanding beyond a singular process (i.e., across processes with slightly dissimilar characteristics) through simulated modeling.[8] Kanarik et al. (2023) also entertained the idea of implementing artificial intelligence (AI) to construct data-driven and predictive models to enable process optimization.[9] The aforementioned works discuss clever approaches for integrating a network of simulated models that is applicable for this study.

With the growing complexity of semiconductor manufacturing processes, manually measuring wafer quality has become increasingly time-consuming and labor-intensive.[10] For instance, a quartz crystal microbalance is traditionally employed to measure the thickness of deposited or etched films on wafers in industry, thereby assessing the etch coverage and quality of the product in an off-line analysis manner. However, the use of a quartz microbalance requires careful operation, bears a low sampling rate, and has significant expenses.[11] To address these challenges, soft sensing methods have emerged in recent years.[12] A soft sensor functions like a traditional sensor by utilizing models that interpret process data that is easier and less expensive to obtain, to predict product properties such as the etch coverage across the wafer. The performance of a soft sensor is highly dependent on the accuracy of its prediction model. Deep learning methods, including recurrent neural networks (RNN), convolutional neural networks (CNN), and transformers, have gathered significant attention for this purpose due to their superior performance and wide applications.[13]

In industry, large fabrication plants often have multiple product flows that share the same etch process.[14] For example, multiple product flows that each create a specific semiconductor device may all use the same 1000 Å Al_2O_3/SiO_2 etch process. Subtle differences between these product flows, such as the underlying substrate geometry, often cause their respective substrates to exhibit different kinetic behaviors for the same process recipe. Thus, a predictor model trained on one specific product line and reactor is not guaranteed to perform similarly for other product lines or reactors, which is why most such predictor models thus far have been focused on a specific process on a specific tool.[15]

However, with the increasing demand for more semiconductor products, fabrication plants must output numerous semiconductor chips. This has led to an increased number of both processing tools and process flows.[16] As the number of tools and flows in a fabrication plant increases, the number of predictor models needed will increase dramatically, leading to numerous unique process data sets. Thus, it is important to explore if data aggregation techniques can be used to create a general predictor model that performs well for a set of process flows. And as the number of unique process data sets continues to increase, it will also become important to create a methodology to determine which data sets should be aggregated for the training of a predictor model.

Modern ALE processes operate quickly, taking less than 10 min to complete a single cycle.[17] However, to capture the inherent process variability for a specific product flow and reactor, process data must be collected from multiple runs. Thus, if the required process data were to be gathered experimentally at an academic scale, it may take months or even years to collect enough data to train an accurate predictor model. An alternative method is to use simulated process data. Simulated process data offers the convenience of being able to directly manipulate kinetic parameters, which drastically decreases the number of runs required to collect enough process data to train a predictor model. With simulated data, valuable insights into the efficacy and optimization of data aggregation techniques within the purview of creating predictor models for semiconductor manufacturing processes can be found.

This work investigates the development of a cross-operation and cross-design predictive model of a two-step thermal ALE procedure to fabricate an aluminum oxide (Al_2O_3) film by utilizing a precursor trimethylaluminum (TMA) and an etching reagent hydrogen fluoride (HF) under high operating temperatures that volatize all the reactants and products. This process is simulated by a computational model with a multiscale framework that intersects features from various time and length domains for atomic, molecular, kinetic, and fluidic properties, all of which govern the kinetics of this etching process. In the Ångstrom and picosecond length and time scales, respectively, atomistic modeling is relevant for discussing the electronic, thermophysical, and kinetic properties of the materials involved in the ALE process by employing ab initio molecular dynamics simulations such as density functional theory and nudged elastic band methods. Additionally, the spontaneity or tendency for a reaction to occur is characterized through elementary reaction pathways that are defined using a statistical mechanics via collision theory (CT) and the Arrhenius model. To establish this stochastic behavior of reaction in a larger length and time scales of micrometers and milliseconds, respectively, a mesoscopic approach through a kinetic Monte Carlo (kMC) method is beneficial for this purpose. Lastly, ALE is dependent on the fluid dynamics when the substrate is exposed to reagents and byproducts that affect the composition and characterization of the substrate surface. Computational fluid dynamics (CFD) is utilized to study the effects of fluid transfer on the substrate surface, in which CFD is not bounded by a maximum length- and time scale exemplified by atomistic and mesoscopic models. The conjunction of these three simulations establish a multiscale model that can[1] resemble realistic experiments conducted in vitro,[2] produce synthetic data at an efficient rate, and ref 3 enable optimization of process operation and design by conducting numerous case studies for similar systems defined by quantifiable variables.

Traditional model reduction methods like proper orthogonal decomposition and approximate inertial manifold techniques work well for the construction of low-dimensional models for transport-reaction processes modeled by one- or two-dimensional (2D) parabolic partial differential equations (PDEs) and can lead to low-order models for controller design; however, this work deals with the development of input/output models capturing nonlinear relationships between process operating variables as inputs (e.g., input flow rates and pressure) and product metrics as outputs (e.g., film spatial uniformity, coverage) that cannot be captured by the traditional model reduction methods for PDEs.[18] Furthermore, the model of the ALE process considered in our work is a multiscale CFD model whose complexity makes the implementation of traditional model reduction methods very challenging given the use of different continuum (e.g., dynamic conservation equations) in the gas phase and discrete (e.g., kMC models)

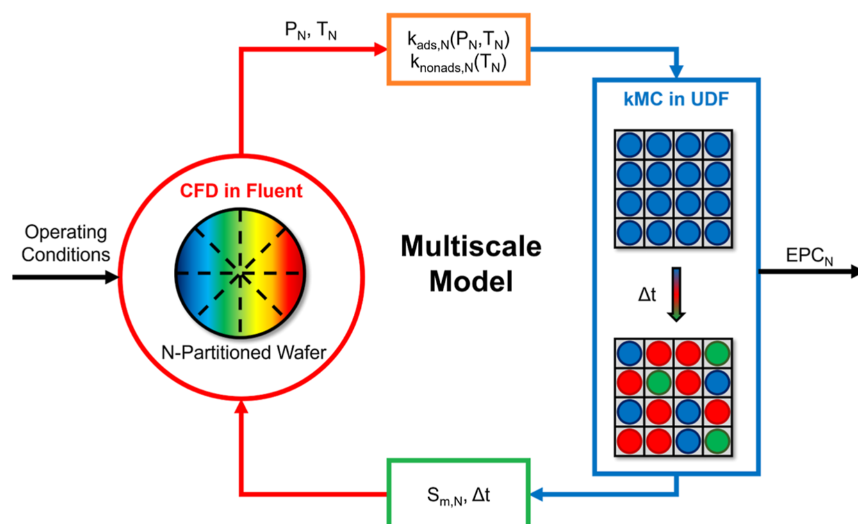


Figure 1. Figure showing the flow of information between the macroscopic simulation in ANSYS Fluent and the mesoscopic simulation executed through the UDFs.

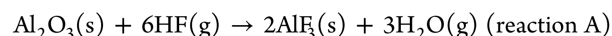
on the film surface. Thus, this work explores using neural network models and the associated statistical analysis as an efficient way to develop predictive operational models for these complex processes.

In summary, this paper investigates the application of a machine learning-based soft sensor for predicting ALE etch rates using process data that can be realistically collected in real-time during the process. The necessary process data were generated through multiscale simulations of an ALE process in a discrete feed reactor, wherein kinetic parameters were adjusted to represent different process flows. To enhance model performance, a data aggregation approach is proposed, which increases the volume of training data by integrating multiple data sets. This manuscript is organized by first describing the overall structure of what data sets are generated and aggregated, and how they are compared. The subsequent section goes into detail regarding the multiscale modeling simulations that were used to generate process data. The following section describes how the predictor model was developed and trained. The last section analyzes the performance of the models trained on different data sets and their implications.

■ PROCESS DESCRIPTION AND DATA GENERATION METHODS

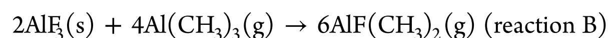
The effectiveness of data aggregation depends greatly on the data being aggregated. To that end, this work seeks to elucidate the possible benefits of aggregating data and explore methods of choosing which data sets should be aggregated by focusing on a specific fabrication process. The ideal process would be one that is relatively simple, has at least two sources of variance, and would benefit from a robust predictor model. One semiconductor manufacturing process that meets all three of the aforementioned criteria is atomic layer etching (ALE).

Specifically, this work explores data aggregation in the context of using ALE to etch away Al_2O_3 while preserving the underlying SiO_2 substrate. This process is simple in that it is only composed of two-half reactions, whereas other processes such as area-selective atomic layer deposition may consist of three or more.[19] The two half-reactions are shown and described below.



$$Al_2O_3(s) + 6HF(g) \rightarrow 2AlF_3(s) + 3H_2O(g) \text{ (reaction A)}$$

In reaction A, gaseous HF prepares the Al_2O_3 surface for etching.



$$2AlF_3(s) + 4Al(CH_3)_3(g) \rightarrow 6AlF(CH_3)_2(g) \text{ (reaction B)}$$

In reaction B, gaseous TMA etches away the prepared ALF_3 surface created in reaction A, releasing a byproduct of dimethylaluminum fluoride (DMAF). These etching reactions were chosen due to their high selectivity toward Al_2O_3 rather than SiO_2. To ensure that all the reagents are in a gaseous form, the process temperature is 573 K and the process pressure is 300 Pa. Additionally, the process consists of reaction A and reaction B cycling back and forth; to reduce reagents from the other half-reaction mixing, a purge step is taken before and after each reaction. Specifically, N_2 is pumped into the chamber until all reagent is removed. For reaction A, 1×10^{-5} kg/s of gas with a mole fraction of 0.1 HF and 0.9 N_2 is pumped into the chamber for 1.2 s and then purged with N_2 for 0.8 s. For reaction B, 1×10^{-5} kg/s of gas with a mole fraction of 0.5 TMA and 0.5 N_2 is pumped into the chamber for 1.5 s and then purged with N_2 for 0.5 s. From experimental testing, these conditions with no disturbances result in complete reactions for both half-reactions.[20]

To simulate these reactions, both half-reactions can be decomposed into multiple intermediate reactions, as detailed in Yun et al. (2021).[20] These intermediate reactions can be classified into two types: adsorption and nonadsorption. By varying the kinetic parameters of these intermediate reactions, variance can be naturally introduced into the process. These varied parameters represent differences in the substrates, such as the device geometry or substrate material.[21] They exist because multiple product flows and devices share process recipes along the manufacturing pipeline, and one of the goals of this paper is to demonstrate that aggregating process data will result in improved model performance despite the fact that the data originate from different sets, each with their own unique kinetic parameters.

For the final criterion, a robust predictor model would, at a minimum, be able to improve manufacturing efficiency by eliminating inspection steps. These inspection steps are conducted at state-of-the-art fabrication centers, where each

https://doi.org/10.1021/acs.iecr.4c03150
Ind. Eng. Chem. Res. 2024, 63, 19693−19706

wafer is inspected after an etch step to ensure that the product specifications and quality standards are met. However, with a robust predictor model, this step could be omitted (or at least executed less frequently) for a majority of processed wafers, greatly increasing overall manufacturing efficiency.

After selecting a process, the next step is to gather data for that process. Generally, most machine learning processes improve in performance when trained on larger data sets. Thus, each data subset needs to be large enough to produce a functioning model to properly compare the effects of data aggregation. To generate all the requisite data, it would take months, if not years, to do so with industrial data. An alternative method is to use simulations to generate the requisite data. By constructing a reactor model and using advanced simulation methods, it is possible to accurately represent an ALE reactor and the reaction kinetics occurring within it.[19] Additionally, simulation runs are much faster to execute than their physical counterparts; with powerful computer processors, it is possible to complete over 16 unique process runs in a single day, for example.[22] Thus, all the data sets analyzed in this work are generated through simulations.

Multiscale CFD Modeling. All simulations must balance accuracy and computational efficiency while generating data at a sufficient rate. Generally speaking, the more accurate a simulation is, the more computational power it takes to carry out that simulation. Because the simulated data is meant to be used as training data for a machine learning algorithm, the simulations must be fast while sacrificing the least amount of accuracy.

To meet these requirements, this work carried out 2D multiscale CFD simulations of an ALE system in a discrete feed reactor. The discrete feed reactor is a stationary reactor with a single inlet at the top, two outlets on the sides, the wafer substrate at the bottom, and a showerhead plate between the wafer and the inlet to promote ideal reagent distribution as shown in Figure 1. The reactor operates by first purging the chamber with N_2 gas, then injecting a reagent gas into the chamber for a preset process time, and finally purging the chamber with N_2 gas once again. This configuration makes it easy to optimize reagent usage while maintaining etch uniformity, which ensures that the end product reaches process specifications.[23] The simulation was bound to 2D geometry to reduce the computational complexity and is valid because both the substrate and reactor have radial symmetry. Multiscale CFD simulations decrease computational complexity (with respect to carrying out a microscopic simulation for the entire process domain) while preserving accuracy by simultaneously carrying out macroscopic and microscopic simulations that constantly communicate between each other. Specifically, the simulation, shown in Figure 1, takes place within the multiphysics software, ANSYS Fluent, which simulates the macroscopic mass and energy dynamic balances within the reactor, which is represented by the left, red section in Figure 1. At each integration time step of the gas-phase continuum model, custom user-defined functions (UDFs) simulate the mesoscopic reaction kinetics on the wafer surface through a kMC simulation scheme over several locations on the wafer surface, which is represented in the right, blue section in Figure 1. Additionally, the macroscopic simulation receives information regarding how much reagent is consumed and product is produced at each point on the wafer, which is the green box at the bottom of Figure 1, while the kMC simulation receives information regarding the partial pressures at each point of the wafer, which is the orange box at the top of Figure 1. These two simulations are carried out simultaneously through UDFs in ANSYS Fluent that enable customizable features applicable to multiscale modeling, and by working in tandem, they improve the overall accuracy of the ALE simulation .

While the macroscopic simulation takes place inside ANSYS Fluent, the reactor was designed in ANSYS SpaceClaim, a 3D CAD software. The discrete feed reactor shown in Figure 2 has

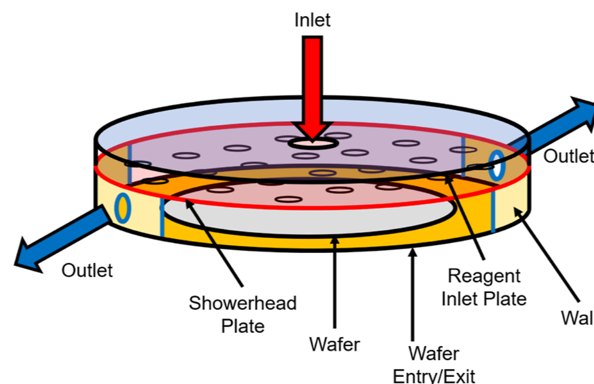


Figure 2. Schematic of the reactor model. For reaction A, a mixture of HF and N_2 enters from the inlet, and HF, H_2O, and N_2 are purged through the outlet. For reaction B, a mixture of TMA and N_2 enters from the inlet, and DMAF and N_2 are purged through the outlet.

a cylindrical shape and consists of a wafer plate at the bottom, an inlet plate in the middle to ensure even reagent distribution, an inlet hole at the top for reagents to enter, and 2 outlet holes at the sides for the etch byproducts and purge gas to exit from. In a previous work, it was found that the efficacy of these reactors mainly depends on 2 factors: the gap distance and inlet plate geometry.[23] Based on previous research into reactor optimization, the reactor used in this work has a gap distance of 5 mm and an inlet plate of 13 equally spaced holes with a diameter of 10 mm.[19]

To observe the different reaction rates across the wafer surface while minimizing excessive calculations, the wafer surface was separated into 5 equidistant sections. Each section has its own kMC simulation, which takes in the average partial pressures of that wafer section and returns the mass fluxes of that wafer section. This allows each section to progress on their own and ultimately allows for analysis of the etching uniformity across the wafer.

The macroscopic simulation evaluates the spatiotemporal behavior of the fluid transport within the reactor by numerically solving the characteristic mass, momentum, and energy transport equations, which are respectively described as follows

$$\frac{\partial \rho}{\partial t} + \nabla \cdot (\rho \vec{v}) = S_{\mathrm{m}} \tag{1}$$

$$\frac{\partial}{\partial t}(\rho \vec{v}) + \rho(\vec{v} \cdot \nabla)\vec{v} = -\nabla P + \nabla \cdot (\overline{\overline{\tau}}) + \rho \vec{g} + \vec{F} \tag{2}$$

$$\frac{\partial}{\partial t}(\rho E) + \nabla(\vec{v}(\rho E + P)) = -\nabla(\Sigma h_j \vec{J}_j) + S_{\mathrm{h}} \tag{3}$$

where ρ is the gas-phase species density, $\vec{v}$ is the velocity of said species, S_{m} is the source generation and consumption flux of that species, P is the operating pressure of the reactor, $\overline{\overline{\tau}}$ is

https://doi.org/10.1021/acs.iecr.4c03150
Ind. Eng. Chem. Res. 2024, 63, 19693−19706

the normal two-rank stress tensor, $\vec{g}$ is the gravitational acceleration constant, $\vec{F}$ is the force acting on the system, E is the accumulated rate of system energy, S_h is the energy source generation or consumption, h_j is the sensible enthalpy flux of gas species j, and $\vec{J}_j$ is the mass diffusion flux of gas species j. Equations 1−3 are numerically solved at each integration time step of 0.001 s. The integration time step used for the solution of the continuum gas-phase dynamic model, 0.001 s, is small enough to ensure stable numerical integration and high simulation accuracy while keeping the computational burden at an affordable cost. Smaller integration time steps have been tested, and they lead to the same numerical results even though the computational burden is increased. Increasing the integration time step to higher values will speed up the calculations but may compromise numerical stability and reduce accuracy, and as the computational burden under the 0.001 s time step is computationally affordable, there is no reason to do so. Clearly, there is an upper bound on the time step that maintains numerical stability of the simulation, but 0.001 s is well within this upper bound.

The only reactions occurring within the reactor take place on the wafer surface. Thus, all boundaries are defined as impermeable walls, save for a few exceptions: the inlet hole at the top is defined as an inlet with a mass flow rate of 1×10^{-5} kg/s, the outlet holes at the sides are defined as outlets with a dynamic pressure of −200 Pa, and the wafer surface is defined as a reaction zone where the molecular species flux calculated in the kMC section are implemented. More details about the macroscopic simulation can be found in ref 23.

The mesoscopic simulation receives the partial pressures calculated in the macroscopic simulation and returns the mass flux of products and reactants. This computation is conducted through a kMC algorithm that simulates the reaction rates of the nonadsorption reactions on the wafer as a function of the kinetic rate constants. For this ALE process, two general types of reactions are considered: adsorption/desorption reactions and nonadsorption reactions. The rate constant of the former is calculated through CT, and that of the latter is calculated with the Arrhenius equation, which are respectively shown below.

$$k_{\mathrm{ads},d} = \frac{2P_d A_{\mathrm{site}} \sigma}{Z_d \sqrt{2\pi m_d k_B T}} \quad (4)$$

where $k_{\mathrm{ads},d}$ is the reaction rate constant for gas species d to adsorb onto the wafer surface, P_d is the partial pressure of gas species d, A_{site} is the surface area of the binding site, σ_s is the experimentally determined sticking coefficient of gas species d, Z is the coordination number of gas species d, m_d is the atomic mass of gas species d, k_B is the Boltzmann constant, and T is the absolute temperature of the wafer surface.

$$k_{\mathrm{nads}} = \nu \exp\left(\frac{-E_a}{RT}\right), \qquad \nu = \frac{k_B T Q^{\ddagger}}{hQ} \quad (5)$$

where k_{nads} is the reaction rate constant for a nonadsorption reaction, ν is the pre-exponential factor, E_a is the activation energy, R is the universal gas constant, T is the absolute temperature of the wafer surface, and h is the Planck constant. Note that when calculating ν, the ratio of the partition functions $Q^{\ddagger}$, Q is assumed to be unity.[24] This assumption was validated by comparing the resulting simulation results to experimental results. Additionally, all the other kinetic properties, such as σ, E_a, and ν, for both half-reactions of the ALE process were also determined in that same work.[20]

The kMC algorithm uses a grid with 300 × 300 sites to represent a larger swath of the wafer surface and randomly generated numbers to represent the stochastic nature of the ALE half-reactions. In a previous work, it was found that this grid size allowed for accurate simulations for a low computational cost.[20] At each integration time step of the macroscopic CFD simulation, the kMC algorithm calculates the rate constants of each possible reaction with eqs 4 and 5 and then uses them to update the 300 × 300 grid.

The basic steps of the algorithm are as follows:

1. Randomly select a site on the grid.
2. Randomly select 2 random numbers, $\gamma_1, \gamma_2 \in (0, 1]$.
3. Sum up all possible reaction rate constants into k_{tot}.
4. Select a reaction by comparing γ_1 to k_{tot}.
5. Calculate the time evolution as $\delta t = -\ln(\gamma_2)/(nk_{\mathrm{tot}})$, where n is the number of active sites on the grid.
6. Terminate when $(\sum \delta t) \geq t_{\mathrm{int}}$, where t_{int} is the integration time step of 0.001 s.

Note that both γ_1 and γ_2 are randomly selected from a uniform distribution of (0,1] as they represent the stochastic nature of atomistic surface reactions. Additionally, for step 3, the reaction rate constants are calculated with eqs 4 and 5 and the atomistic constants found in ref 20. Equation 4 is dependent on the partial pressure of the reactant species, which means that all the reaction rate constants must be recalculated at each time step as the partial pressures of each species cannot be guaranteed to be constant.

Steps 1, 2, 3, 5, and 6 are relatively simple, but step 4 requires a more in-depth explanation. To select a reaction, the following expression is evaluated for each possible reaction

$$\sum_{i=1}^{r-1} k_i < \gamma_1 k_{\mathrm{tot}} \leq \sum_{i=1}^{r} k_i \quad (6)$$

where r is the reaction the expression is being evaluated for, k_i is the ith reaction, γ_1 is a randomly selected number created in step 2, and k_{tot} is as described in step 3. Additionally, for all grid sites and a selected γ_1, eq 6 will only be true for a single, unique reaction, which is the selected reaction. A more detailed explanation of the kMC algorithm, including pseudocode, can be found in a previous work.[19]

At the end of the multiscale simulation, the reaction coverage across the wafer can be estimated by examining the 300 × 300 of the kMC simulation. As the coverage is a fraction of how much of the surface has completed the reaction, it is simply the number of sites that have reached the final product divided by the total number of grid sites. The coverage can then be converted into the etch rate by multiplying by 0.46 Å/cycle, as that is the amount of Al_2O_3 that would be etched away if the wafer reached full coverage in both half-reactions.[20]

With the simulation settings described above, the accuracy of the simulations is uncompromised while the computational efficiency remains high. When run on powerful CPU-based (central processing unit) nodes with 24 and 48 cores with 384 GB and 512 GB of DRAM (dynamic random-access memory), respectively, a full simulation of both half-reactions took approximately 8 h. Thus, by using 4 such nodes, simulating 100 process runs can be completed in 200 h.

Process Data Sets. For this project, two types of data sets were created: one that represents a process, and one that imitates raw industrial data. The main difference between these

https://doi.org/10.1021/acs.iecr.4c03150
Ind. Eng. Chem. Res. 2024, 63, 19693−19706

two data set types is that the former data sets are meant to be compact data sets that represent many years of process data. Their kinetic parameters are set to a specific number within a range and do not include any noise; thus, they will be called process-specified data sets. The latter data set, which is used to compare model performances, is intended to represent a series of process runs. The kinetic parameters for each run are randomly selected from a Gaussian distribution that represents a specific process, which introduces variation into each process run. Thus, this data set will be called the random-run data set.

To train and validate the predictor models, four separate process-specified data sets with varying kinetic parameters were created. The varied kinetic parameters were chosen to represent common process shifts in a manufacturing environment, such as deposition of reactants to reactor sidewalls affecting the nonadsorption reactions or complex device geometries affecting sticking coefficients.[25]

The four process data sets are differentiated by their kinetic constants, which have been uniquely modified for each data set. Specifically, the sticking coefficients (σ) and pre-exponential factors (ν), are multiplied by a constant ranging from 0.5 to 1.5. The four processes are listed in Table 1 where

Table 1. Kinetic Parameter Ranges for Each Process

	f_σ	f_ν
CT	[0.5,1.5]	
TST		[0.5,1.5]
MIX	[0.5,1.3]	[0.5,1.5]
INV	[0.5,1.3]	[1.5,0.5]

f_σ is the constant that the sticking coefficient (σ) found in eq 4 is multiplied by and f_ν is the constant for the pre-exponential factor (ν) found in eq 5. Furthermore, in CT, only f_σ is varied; in TST, only f_ν is varied; in MIX, f_σ and f_ν share the same value; finally, in INV, f_σ and f_ν are inversely correlated such that their sum is always 2. For example, the data set that represents MIX consists of 25 simulations, where the values of f_ν and f_σ are represented as

$$f_{\nu,i} = f_{\sigma,i} = 0.47 + 0.03 \cdot i$$

where i is the ith simulation. By carrying out a similar procedure for the CT, TST, and INV processes, 4 unique data sets comprising 25 simulations each are generated, making for a total of 100 simulations. These data sets are then aggregated in various combinations before being used to generate a prediction model.

To analyze the performance of the prediction models generated with the process-specified data sets, each model will be used to estimate the etch rate of a random-run data set comprising 60 process runs. The kinetic parameters for each run will be randomly selected from a Gaussian distribution that has an average of 1 and a standard deviation of 0.1. To consider both the average etch rate and the standard deviation of the etch rate across the wafer, the etch rate is measured at 5 inspection points across the wafer that correspond to the 5 wafer sections described in the "Multiscale CFD Modeling" section. The resulting etch rate mean and spread for the random-run data set are illustrated in Figures 3 and 4, respectively.

Note that the etch rate distribution is not Gaussian, as the process metric of etch rate does not vary linearly with respect to the kinetic parameters. As both ALE half-reactions are naturally self-limiting, high kinetic parameters that increase the reaction rate of each half-reaction will minimally impact the overall etch rate.[20] Thus, the etch rate distribution is skewed.

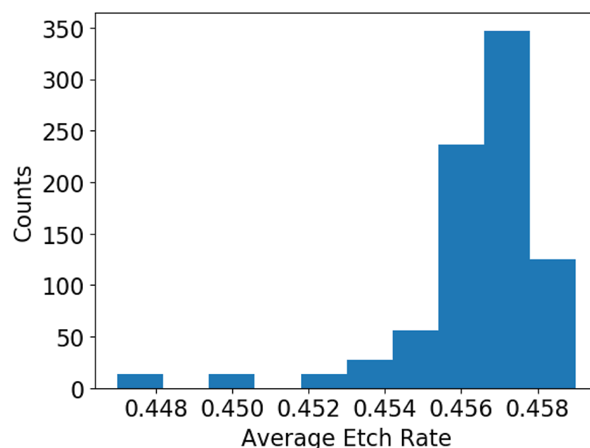


Figure 3. Histogram of the average etch rate across the entire wafer for each run in the random-run data set.

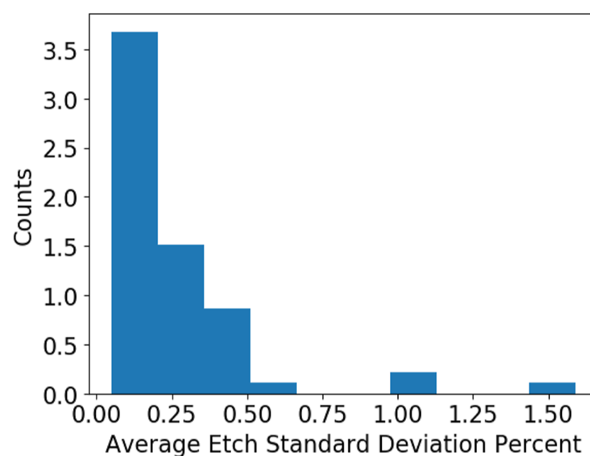


Figure 4. Histogram of the standard deviation of the etch rate across the wafer as a percentage of the average etch rate for each run in the random-run data set.

Besides measuring the performance of each model as a predictor, it is also possible to convert them to classifiers and measure their efficacy in that regard. This approach is accomplished by adding a filter to both the random-run data set and the model output that analyzes the etch rate mean and standard deviation and classifies it as either a "pass" or "fail." The specific pass/fail metrics were chosen such that the fail rate is around 2% for a process with kinetic parameters with a mean of 1 and standard deviation of 0.1. This means that, for the etch rate mean and etch rate standard deviation, if either process variable is more than 2 standard deviations from the mean, the run has failed. For the random-run data set, this means that a run fails if the etch rate mean is less than 0.444 or if the etch rate standard deviation is greater than 2% of the mean. When applied to the random-run data set, these criteria result in a fail rate of 2.3%. This fail rate is consistent with industrial data sets the authors have worked with.

■ TRANSFORMER MODEL TRAINING METHOD

The optimal model training method will differ for each process, depending on what process data are readily available and able

https://doi.org/10.1021/acs.iecr.4c03150
Ind. Eng. Chem. Res. 2024, 63, 19693−19706

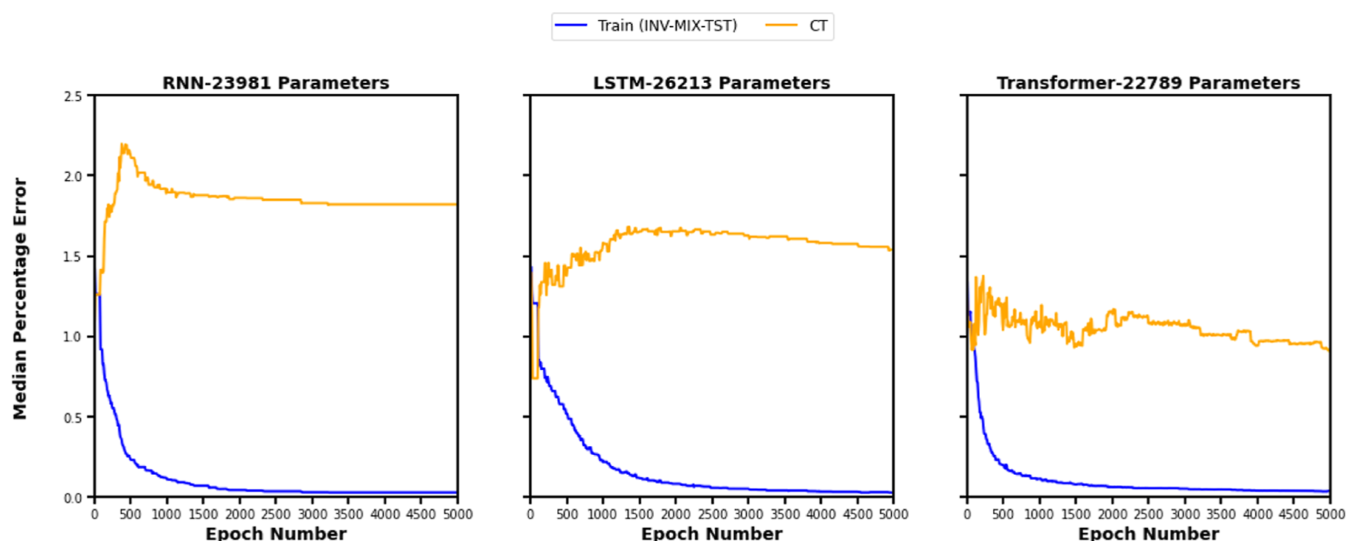


Figure 5. Comparison between RNN (left), LSTM (center), and transformer (right) models. Two data sets of real-time process data were considered: a mixed data set from multiple processes that was used for both training and validation (blue), and a data set from a single process that was only used for validation (orange).

to be collated into data sets. For the purposes of this work, we aim to train a transformer model that takes real-time process data as an input and outputs the expected etch rate.

Input Variables for Model Training. For an ALE reactor operating in an industrial environment, pressure and temperature are critical process variables that can be measured at high frequencies with low latency, enabling the generation of comprehensive time-series data sets for process modeling. To ensure consistency and simplicity, the wafer temperature and reactor operating pressure are maintained at near constant values by feedback control systems. Then, the surface pressure of the wafer, measured in Pascals, is recorded every 0.1 s, which aligns with the limitations of real-life industrial sensors. However, to prevent information loss, because the simulation is conducted with an integration time step of 0.001 s, the pressure data is also extracted at every integration time step. Prior studies on the ALE of Al_2O_3 in discrete feed reactors indicate that a 2 s period is sufficient for achieving full wafer coverage under most conditions.[23] Consequently, the pressure measurement period is limited to 2 s for each half-cycle, resulting in a maximum of 20 points of time-series wafer surface pressure data per half-cycle. Each data point contains 5 pressure readings, which are sampled from the 5 different inspection points on the wafer surface described in the "Multiscale CFD Modeling" section. The time-series pressure data collected for both the HF and TMA half-cycles are then concatenated into a complete data set for a single ALE run.

Output Variables for Model Training. The output variable of the ALE process for the soft sensor is the etch coverage calculated for each of the previously described inspection points at the end of the overall process. This measurement offers valuable information about both the overall etch rate and the etch rate uniformity across the wafer. The total coverage after two half-cycles is defined by

$$\text{cov}_i = \text{cov}_{\text{HF},i} \cdot \text{cov}_{\text{TMA},i}, \qquad i \in \{1, 2, 3, 4, 5\} \tag{7}$$

where cov_i is the total coverage at the end of the process on inspection point i, $\text{cov}_{\text{HF},i}$ is the coverage at the end of HF half-cycle on inspection point i, and $\text{cov}_{\text{TMA},i}$ is the coverage at the end of TMA half-cycle on inspection point i. Each half-cycle is programmed to end at $t = 2.0$ s, where t is the processing time.

The true total coverages are obtained through multiscale simulations as described in the previous section and then compiled into data sets consisting of their respective input and output data. When effectively trained, the soft sensor model is expected to output the coverage data with minimal absolute percentage error compared to the true values.

Development of the Predictor Model. *Introduction to Time-Series Modeling.* To develop a prediction model based on time-series input data, deep learning methods such as RNN and long-short-term memory (LSTM) neural networks have been widely explored and applied across various fields, including chemical engineering.[26] These neural networks are capable of processing long sequences of data and learning about the information between each element. This results in the models learning about the correlations within the data sequence. This capability enables RNNs and LSTMs to better understand and fit time-series data compared to traditional feedforward neural networks (FNN).

A novel deep learning model, the transformer, has emerged in recent years and rapidly gained significant attention in the field of natural language processing (NLP) due to its outstanding and record-breaking performance on most NLP tasks.[27] The transformer network structure has also demonstrated equally impressive performance in computer vision tasks such as image classification and object detection.[28] Additionally, one of the most popular and advanced examples of AI, the large language model (LLM), which includes products such as ChatGPT and BERT, employs the transformer architecture. By utilizing billions of training data points and model parameters, LLMs are capable of generating human-like responses and solving a wide range of problems, offering substantial potential for future applications.[29]

The transformer model employs an encoder-decoder architecture to handle a sequence of time-series data. A multihead self-attention mechanism[30] is used within each block to compute the relevance of each element in the sequence relative to every other element. This approach enables the model to effectively capture intercorrelations

https://doi.org/10.1021/acs.iecr.4c03150
Ind. Eng. Chem. Res. 2024, 63, 19693−19706

across the entire sequence of time-series data. The attention mechanism calculates attention scores for each pair of elements, which are then normalized using a softmax function to aggregate the attention values. This process allows the transformer to maintain a comprehensive understanding of the relationships within the data sequence, surpassing the capabilities of traditional LSTM networks.[30,31]

LSTM networks, while effective in handling time-series data and addressing the vanishing gradient problem occur in simpler RNNs, can still suffer from memory loss over long data sequences. This limitation arises because LSTMs process data sequentially, which struggles to capture long-range dependencies. In contrast, the self-attention mechanism of the transformer allows it to simultaneously consider all positions in the sequence, ensuring that long-term dependencies are preserved and effectively learned. This ability to maintain a global perspective on the data sequence makes the transformer particularly well-suited for tasks that require a deep understanding of a long sequence of time-series data.

In Figure 5, a transformer, RNN, and LSTM model are compared to demonstrate their respective effectiveness at training models. The number of optimized parameters for the three tested models are intentionally selected to have similar values for the purpose of comparison. Both the RNN and LSTM networks have one recurrent layer with 24 neurons, and the output vectors of each recurrent unit are concatenated together and fed to a FNN network with one hidden layer that contains 32 neurons to generate the output vector. Figure 5 shows that the transformer model outperforms the other two models in terms of both learning efficiency on the blue training data set and median percentage error on the orange unseen test data set, which has a different disturbance from the training data set. Thus, the ability of the transformer model to maintain a general, global perspective on the data sequence makes it particularly well-suited for tasks that require a deep understanding of a long sequence of time-series data.

Transformer Soft Sensor Structure. The overall structure of the transformer soft sensor for the ALE process is shown in Figure 6.

Following the input layer, a dense layer, or embedding layer, is used to linearly encode the input data into a suitable format for the transformer encoder blocks. The applied transformer network consists solely of encoder blocks, as the task only involves regression and does not require the model to generate a series of future predictions.

Positional encoding is added to the input sequence of the embedded vector of wafer surface pressure time-series data because the transformer blocks do not inherently recognize the order of the input elements. This encoding layer provides the necessary information regarding the order of the input elements to the transformer encoder blocks. The positional encoding equations have the following form

$$PE(\text{pos}, 2i) = \sin\left(\frac{\text{pos}}{10,000^{2i/d_{\text{model}}}}\right) \quad (8)$$

$$PE(\text{pos}, 2i+1) = \cos\left(\frac{\text{pos}}{10,000^{2i/d_{\text{model}}}}\right) \quad (9)$$

where PE is the positional encoding value, i is the index indicator within a single vector in the sequence that defines the pressure measurement vector at a specific time, $2i$ represents an even index position in the vector, $2i + 1$ corresponds to an odd index position, pos is the index number of the vector in the sequence, and d_{model} is the embedded vector dimension.

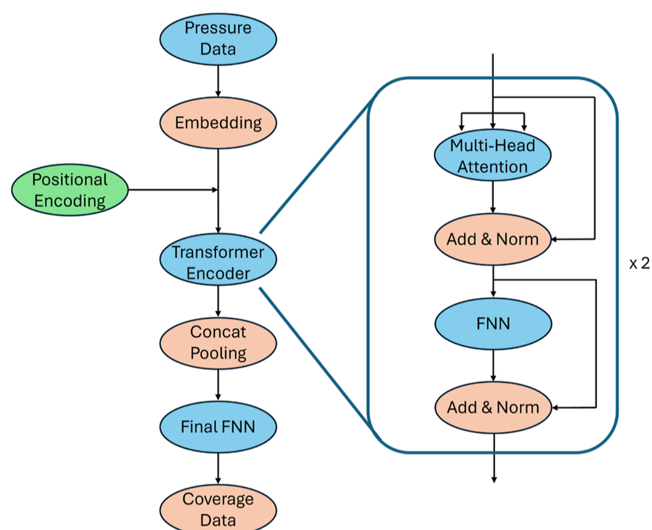


Figure 6. Structure of the soft sensor transformer network. Pressure data is embedded by a dense layer with dimension 8, undergoes positional encoding, and then is fed into 2 identical multihead encoder blocks. Inside the encoder block, there is an internal FNN with a hidden layer of 64 neurons. The output of the encoders are combined by concatenation pooling and then fed into the final FNN, which has 2 layers of 64 neurons each, and it outputs the reaction coverage.

The encoder block employs the principle of residual learning and includes a single-layer neural network to process the output data from the transformer encoder blocks. Subsequently, a concatenation pooling operation is applied, which connects and aggregates the outputs from all the transformer blocks from each element in the sequence to prevent information loss. This pooled data is then fed into a final FNN to produce the total wafer coverage values. The hyperparameters of the soft sensor transformer model are summarized in Table 2.

Table 2. Hyperparameters for the Transformer Model

model hyperparameter	value
Input dimension	5
Embedding dimension	8
Number of heads	2
FNN neurons	64
Dropout ratio	0.25
Encoder layer number	2
Final FNN layer number	2
Final FNN neurons	64
Output dimension	5

Transformer Model Training. The data is separated into two parts: 80% of the data is used for training, while the remaining 20% is used for validation testing. The training-testing split is conducted randomly using the data set splitting module from the scikit-learn package. The mean squared error (MSE) loss function is applied as the loss criterion, and the ADAM optimizer is employed for model parameter updates. The loss value on the validation data set is recorded at each training epoch, and the model with the lowest MSE on the validation data set is saved as the most up-to-date candidate

https://doi.org/10.1021/acs.iecr.4c03150
Ind. Eng. Chem. Res. 2024, 63, 19693−19706

model. The training hyperparameters, including batch size, number of training epochs, learning rate, and initialization algorithm, are all optimized using a trial-and-error method based on the model performance on the validation data set. This approach is feasible and time efficient both because the data set size is not large and because the transformer architecture is well suited for parallel computing and training. The final optimized hyperparameters for the model included a batch size of 64, 700 total training epochs, a learning rate of 0.001, and the Xavier initialization method. The random seeds used in data splitting and model training were also fixed to improve reproducibility.

HEURISTIC ANALYSIS METHOD

One method to guarantee creating the most optimal model is to simply create models for every possible combination of process-specified data sets. By doing so, and then comparing their MSE, it is trivial to pick out the model with the best performance. However, this process becomes increasingly inefficient as the number of possible data sets increase. Specifically, there will be $n!$ possible models for n data sets. And in industry, where there are often hundreds of data sets for a particular etch process, it is unrealistic to exhaustively search through every single data set combination. Thus, there is a motivation to create a heuristic that determines which data sets should be aggregated.

Statistical Methods Introduction. A well-designed heuristic is essentially a distillation of the intrinsic statistical characteristics of a data set. One way to extract this statistical information is to train models on exactly one process-specified data set and then compare their performances on the overall data set that is the combination of all process-specified data sets. This method reduces the number of models that must be trained to determine the most optimal model, changing it to scale linearly with the number of data sets.

The ideal data sets to aggregate are ones that are different from one another. Training the model on a wide variety of data will allow it to better understand and predict edge-case scenarios, which are oftentimes when the run fails. This selection criteria for data set aggregation can be translated into selection criteria for the models trained on single process-specified data sets: aggregate data sets whose representative models are complementary to each other. If one model performs sufficiently on data set B and poorly on data set C, then the ideal data set to aggregate it with would be one whose representative model performs well on data set C and poorly on data set B.

Two statistical methods of capturing this relationship are explored in this work: covariance and the Pearson correlation coefficient (PCC). The covariance measures the general strength of the relationship between any 2 data sets, and the PCC measures the level of linear correlation between any 2 data sets. Thus, both statistical methods will be used to create heuristics that evaluate which data sets should be aggregated. Then, by comparing those results to that of the brute-force exhaustive search, the better heuristic will be determined.

Heuristic Evaluation Method. First, a model is trained on each process-specified data set as specified in the "Process Data Sets" section. Then, these single-process models are run on an aggregate data set consisting of all 4 process-specified data sets to create a set of 4 MSE data points for each single-process model. Finally, covariance and PCC is calculated between each pair of models for their MSE data points. Covariance is specifically calculated with the formula below

$$\mathrm{Covar}_{X,Y} = \frac{\sum_{i=1}^{N} (x_i - \bar{x})(y_i - \bar{y})}{N - 1} \tag{10}$$

where X, Y represent any two models, x_i is MSE of model X for data set i, $\bar{x}$ is the average MSE of model X, y_i is MSE of model Y for data set i, $\bar{y}$ is the average MSE of model Y, and N is the number of data sets. In this case, $N = 4$. The PCC is similarly calculated with the formula below.

$$\rho_{X,Y} = \frac{\mathrm{Covar}_{X,Y}}{\sigma_X \sigma_Y} \tag{11}$$

where $\mathrm{Covar}_{X,Y}$ is the covariance between any two models X, Y as calculated in eq 10, σ_X is the standard deviation of model X, and σ_Y is the standard deviation of model Y.

Finally, for any given model trained on aggregated data sets, we can determine the heuristic value of that model by averaging the covariance or PCC values for the composite pairs of the aggregated data set. For example, if the aggregated data set contains data sets X, Y, Z, then the composite pairs are every unique pair: (X, Y); (X, Y); (Y, Z). Then, the covariance heuristic value of a model trained on data sets X, Y, Z can be found with

$$h_c(M(X, Y, Z)) = \frac{\mathrm{Covar}_{X,Y} + \mathrm{Covar}_{X,Z} + \mathrm{Covar}_{Y,Z}}{3} \tag{12}$$

where h_c is a function that returns the covariance heuristic value of a model, $M(X, Y, Z)$ is the model trained on data sets X, Y, Z, and $\mathrm{Covar}_{X,Y}$ is the covariance as defined in eq 10. Similarly, the PCC heuristic value is defined as

$$h_\rho(M(X, Y, Z)) = \frac{\rho_{X,Y} + \rho_{X,Z} + \rho_{Y,Z}}{3} \tag{13}$$

where $h_\rho()$ is a function that returns the PCC heuristic value of a model and $\rho_{X,Y}$ is the PCC as defined in eq 11. With eqs 12 and 13, we can now calculate heuristic values for any model based on the data sets used to train that model.

PREDICTOR MODEL RESULTS AND DISCUSSION

As shown in Table 1, there are 4 process-specified data sets. This means that there are 15 possible data set combinations that can be used to train a model: 4 models are trained on only 1 data set, 6 models are trained on 2 data sets, 4 models are trained on 3 data sets, and 1 model is trained on all 4 data sets. There are 2 methods to evaluate the performance of these 15 models. The first is to examine the MSE of the model when it is run on the validation portion of the process-specified data sets, and the second is to examine the accuracy of the model when run on the random-run data set described in the "Process Data Sets" section. The full results of the MSE test are shown in Table 3.

Where the columns represent the validation data set the model was run on, the rows represent the training data sets of the model, and the values in the table are the resulting MSE.

Multi-Process Model Performance. From Table 3, the model with the best performance across all 4 processes is the one trained on all 4, with an MSE of 0.16. However, it is difficult to grasp why just by examining the complete tabulated results. To better compare the 4 data sets, we begin by

Table 3. MSE of Each Model for Each Validation Dataset[a]

Processes Trained On	Processes Tested on CT	INV	MIX	TST	All
CT	0.03	0.63	2.46	1.82	1.24
INV	0.63	0.03	2.46	1.69	1.20
MIX	2.36	2.60	0.05	0.18	1.67
TST	3.35	3.69	2.21	0.18	2.36
CT+INV	0.05	0.04	2.46	1.95	1.12
CT+MIX	0.21	0.95	0.32	1.75	0.81
CT+TST	0.08	1.02	2.31	0.13	0.89
INV+MIX	1.38	0.31	0.37	1.66	0.93
INV+TST	0.75	0.25	2.41	0.33	0.94
MIX+TST	2.16	2.41	0.24	0.24	1.26
CT+INV+MIX	0.13	0.15	0.16	1.78	0.56
CT+INV+TST	0.11	0.12	2.26	0.15	0.66
CT+MIX+TST	0.13	1.07	0.17	0.18	0.39
INV+MIX+TST	1.15	0.08	0.10	0.11	0.36
All	0.16	0.15	0.17	0.17	0.16

[a]The MSE values are colored such that low MSE scores are green and high MSE scores are red.

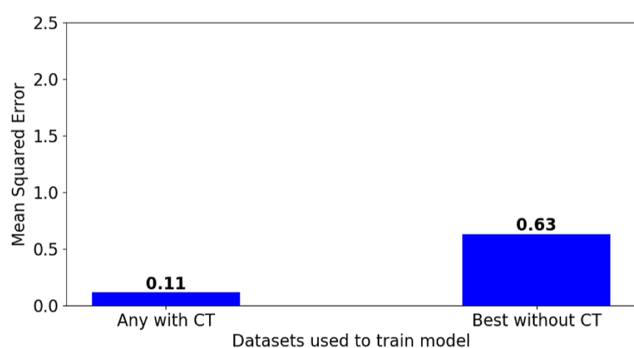
examining the degree of independence between them. In other words, we want to know if it is possible to create a strong model for a data set without training the model on that very same data set.

To answer this question, we compared the average performances of the models trained on a particular process to the performance of the best model not trained on the process. Figure 7a−d are bar charts showing this comparison for each process.

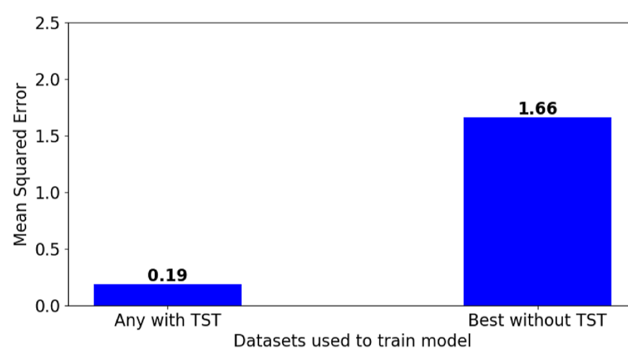
Figure 7a−d demonstrates that, for all processes, the model performance is drastically improved whenever the model is trained on the corresponding data set of the process. This illustrates that the failure mechanism for each data set is unique. Thus, if a model is missing any one data set, it will perform badly for the validation portion of that data set. On the other hand, if there was a process that was not independent, then the data set of that process can be excluded from the training data set for the best performing model.

To demonstrate this point, let p_d be a process that is not independent; that is to say, there must be a model m_{nd} that is trained on the processes p_i, p_j but performs as well as a model trained on p_d. As other independent process data sets are aggregated on, any models whose training data sets include p_i, p_j and omit p_d will still perform well on the p_d. Thus, the model that performs best on the union of all the process data sets can omit p_d as long as p_i, p_j is included. By repeating this procedure, all the nonindependent processes can be omitted to yield the ideal training data set.

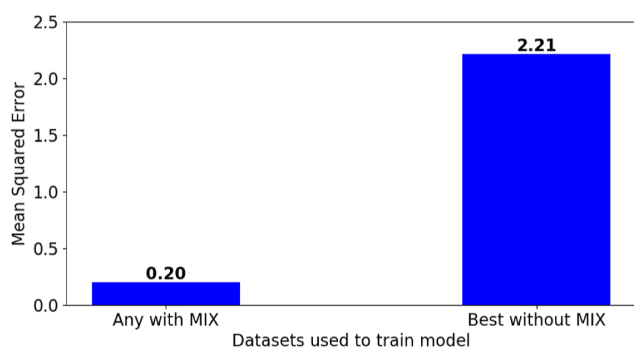
So, for a manufacturing environment that spans the full range of all the individual process data sets, the best predictor model will be trained on an aggregated data set that includes the data set of all independent processes. This environment is equivalent to one where the process runs have relatively high fail rates due to volatile processing conditions.



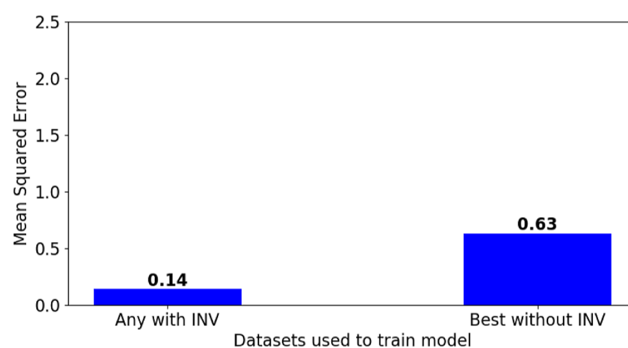
(a) CT Process



(b) TST Process



(c) MIX Process



(d) INV Process

Figure 7. These subfigures measure model performance by comparing their MSEs when tested on the validation data set of the stated process. The left bar is the average performance of all models trained on the stated process, and the right bar represents the best model that was not trained on the stated process.

https://doi.org/10.1021/acs.iecr.4c03150
Ind. Eng. Chem. Res. 2024, 63, 19693−19706

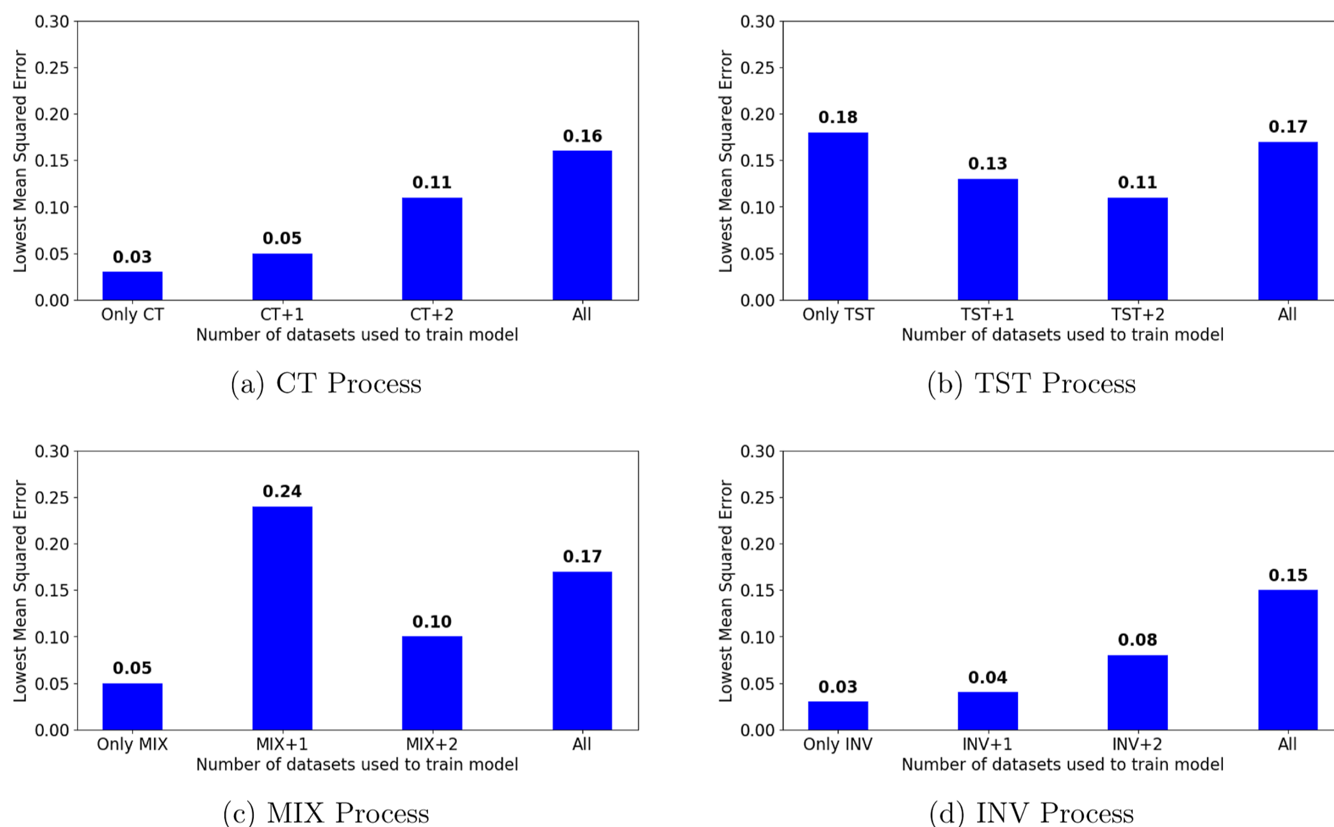


Figure 8. These subfigures measure model performance by comparing their MSEs when tested on the validation data set of the stated process. From left to right, the number of data sets used to train the model increases from 1 to 4 as stated in the *x*-axis, with the stated process data set always being included. Each bar represents the highest performing model for that number of data sets.

Single-Process Model Performance. It is also important to understand how to best optimize a model for performance on a single process, which is more representative of a low variance process than multiple processes. This is found by examining Table 3 and determining which model has the smallest MSE in each column. The best model for each process is the model trained only that process, except for TST; the best model for that process was trained on CT + TST. This demonstrates that adding the data of other processes into the training data dilutes the amount of representative process data for that process, which generally causes the model performance to decrease. To gain more insight into how aggregating data affects the model performance on a single process, Figure 8a−d illustrate the MSE of the best performing model as a function of the number of data sets used to train it.

Figure 8a−d demonstrate that, generally, the performance of a model on a single process decreases as data from other processes is aggregated into the training data set. This holds true for both the CT and INV processes, but the MIX and TST processes deviate from the other 2 processes.

While the general trend for MIX as shown in Figure 8c is that the single-process model performance decreases with increasing aggregation, the model trained on 2 data sets performs unexpectedly poorly. This can be attributed to the fact that MIX is the most independent process. In the previous section, MIX had the highest MSE when looking at models not trained on the process. This means that MIX cannot be represented well by an aggregation of the other process data sets. So in the case of single-process models trained on 2 data sets, MIX will be diluted the most compared to the other 3 processes, which causes it to have an outlier in terms of performance.

TST actually demonstrates the inverse of the expected response where the single-process model performance increases with increasing aggregation. This is likely due to the fact that TST had the highest MSE when looking at models trained on that process among Figure 7a−d. Because even the models trained on the TST process data set had mediocre performances on the TST validation data set, it demonstrates that the transformer method struggles to create a good model for this data set. But as more data sets are aggregated into the model, the model performance improves. This demonstrates that data aggregation can also help improve model performance for processes that are difficult to model with just their own process data.

From the multiprocess and single-process analyses, it is seen that the ideal amount of data aggregation depends on not just the characteristics of the process data sets, but also on the process environment. If there are multiple processes with poor controls and large variability in between runs, then the model should be trained on as many independent data sets as possible. For single process models, while there is a general pattern for how data aggregation affects model performance on single processes, there are enough exceptions that the pattern cannot be used to determine what data sets will yield the model with the best performance. Thus, to further improve manufacturing efficiency, there needs to be a method to choose which data sets to aggregate together to train a predictor model for low-variance processes.

https://doi.org/10.1021/acs.iecr.4c03150
Ind. Eng. Chem. Res. 2024, 63, 19693−19706

Heuristic-Based Assessment of Data Sets. Another way to analyze model performance for low-variance processes is to observe their performance on the random-run data set as described in the "Process Data Sets" section. These data sets represent a process environment where runs are expected to pass and where most runs have similar kinetic parameters. Additionally, we will explore the heuristics described in the "Heuristic Analysis Method" section to see if they can predict which data sets should be aggregated for a low-variance process environment. The 2 proposed heuristics of covariance and PCC are examined, and the values of the statistical methods for each pair of single-process models are shown in Figures 9 and 10. Note that the sign of the heuristic represents whether the relationship between the 2 data sets is positively or negatively correlated. This relationship is not important, as we are only concerned with how strong the correlation between 2 data sets is, not the direction of said correlation. Thus, we are only interested in the magnitude of the calculated covariance and PCC. So, for each data set, we find the covariance and PCC between it and all the other data sets, which is shown in Tables 4 and 5.

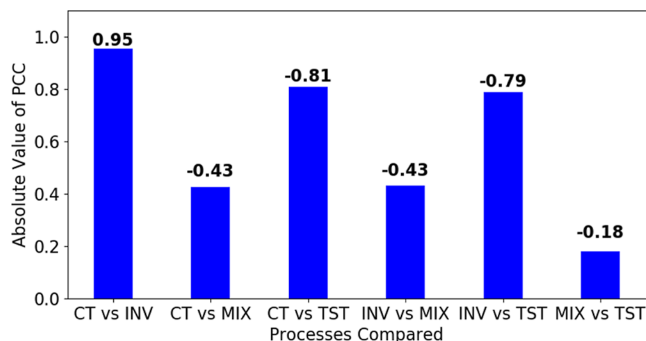


Figure 9. Comparison of the PCC values for each pair of processes. The data sets used to calculate the PCC are the MSE values of a single-process model for each process.

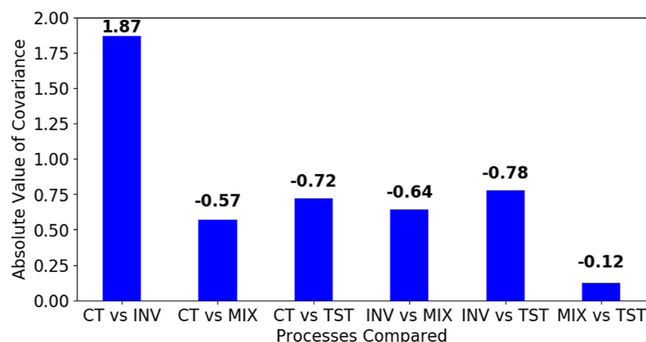


Figure 10. Comparison of the covariance values for each pair of processes. The covariance is calculated with the same data sets used to calculate the PCC.

With Tables 4 and 5, we can predict what the covariance and PCC will be for a model trained on multiple data sets by following the procedure described in the "Heuristic Evaluation Method" section. Note that it is desired for the predicted heuristic value to be low. The lower the predicted value, the less overlap there is between each constituent data set, making that combination of data sets more likely to better represent the overall process.

Table 4. Average Absolute Covariance Values for Each Dataset[a]

	CT	INV	MIX	TST
CT	1.00	0.95	0.43	0.81
INV	0.95	1.00	0.43	0.79
MIX	0.43	0.43	1.00	0.18
TST	0.81	0.79	0.18	1.00

[a]Each entry is colored such that lower values are green and higher values are red.

Table 5. Average Absolute PCC Values for Each Dataset[a]

	CT	INV	MIX	TST
CT	1.00	0.95	0.43	0.81
INV	0.95	1.00	0.43	0.79
MIX	0.43	0.43	1.00	0.18
TST	0.81	0.79	0.18	1.00

[a]Each entry is colored such that lower values are green and higher values are red.

To test the performance of the two proposed heuristics, we created and ran four models on the random-run data set described earlier. Then, we evaluated the covariance and PCC heuristic for each model to see how well they predict the ranking of the models. The results are summarized in Table 6.

The true performance of each model shown in Table 6 is represented in the acc. column. From it, we can see that the model trained on the INV, MIX, and TST process data sets performed the best. The PCC heuristic predicts this correctly, assigning it the lowest score. However, the covariance heuristic failed to do so, instead predicting that the model trained on the CT, MIX, and TST process data sets would perform the best. Another point of comparison is to see which heuristic makes the most correct predictions. The PCC heuristic correctly predicts that CT + INV + MIX is ranked fourth, incorrectly predicts that CT + INV + TST is ranked fifth, correctly predicts that CT + MIX + TST is ranked second, correctly predicts that INV + MIX + TST is ranked first, and incorrectly predicts that all is ranked third. Overall, it made 3 correct predictions. On the other hand, the covariance heuristic correctly predicts that CT + INV + MIX is ranked fourth, incorrectly predicts that CT + INV + TST is ranked fifth, incorrectly predicts that CT + MIX + TST is ranked first, incorrectly predicts that INV + MIX + TST is ranked second, and incorrectly predicts that all is ranked third for a total of only 1 correct prediction. For both metrics, the PCC heuristic outperformed the covariance heuristic. Thus, these results demonstrate that the PCC heuristic is more accurate at predicting the performance of aggregated data set models than the covariance heuristic.

■ CONCLUSION

To supplement the growing need for increased semiconductor manufacturing efficiency, a novel real-time etch rate predictor model was created with simulated process data. A multiscale method that encompasses both the macroscopic and

https://doi.org/10.1021/acs.iecr.4c03150
Ind. Eng. Chem. Res. 2024, 63, 19693−19706

Table 6. Ranking of 4 Models[a]

		etch MSE median	SD % MSE median	acc. (%)	avg. corr (%)	avg. covar
trained on	CT + INV + MIX	4.87×10^{-7}	1.95×10^{-2}	95.00	0.603	1.027
	CT + INV + TST	5.22×10^{-7}	2.41×10^{-2}	96.67	0.850	1.123
	CT + MIX + TST	5.30×10^{-7}	3.55×10^{-2}	96.67	0.473	0.470
	INV + MIX + TST	4.95×10^{-7}	2.42×10^{-2}	98.33	0.467	0.513
	all	1.78×10^{-6}	4.15×10^{-2}	93.33	0.598	0.783

[a]The first 2 columns represent the true prediction ability of the model, the 3rd represents the true classification ability of the model, the 4th represents the predicted performance by the PCC heuristic, and the 5th represents the predicted performance by the covariance heuristic.

mesoscopic domains was used to simulate the real-time evolution of ALE of aluminum oxide. With this method, four different, unique process data sets were created with varying kinetic parameters. Then, predictor models were trained on various combinations of these data sets. It was found that, for systems with high process variance, aggregating all the data sets resulted in the best performance, but for systems with low process variance, aggregating all the data sets would not result in the best performance. Because most manufacturing environments strive for low process variance, it is thus necessary to determine a way to estimate which data sets to aggregate for low process variance environments. We proposed two possible heuristics to choose data sets to aggregate: covariance and the PCC. After comparing model performances on a data set of consecutive process runs, it was found that the PCC heuristic was the best predictor of performance for models trained on aggregated data. Further research into other possible heuristics, the many applications of an accurate real-time predictor model, and the scalability of these findings to larger groups of data sets is still needed, but the initial results indicate that such models can be effectively and easily created. In another forthcoming work, we have demonstrated the approach presented in the present paper using industrial data.

AUTHOR INFORMATION

Corresponding Author

Panagiotis D. Christofides − *Department of Chemical and Biomolecular Engineering, University of California, Los Angeles, California 90095, United States; Department of Electrical and Computer Engineering, University of California, Los Angeles, California 90095, United States;* orcid.org/0000-0002-8772-4348; Email: pdc@seas.ucla.edu

Authors

Henrik Wang − *Department of Chemical and Biomolecular Engineering, University of California, Los Angeles, California 90095, United States;* orcid.org/0000-0001-5863-1209

Feiyang Ou − *Department of Chemical and Biomolecular Engineering, University of California, Los Angeles, California 90095, United States*

Julius Suherman − *Department of Chemical and Biomolecular Engineering, University of California, Los Angeles, California 90095, United States*

Matthew Tom − *Department of Chemical and Biomolecular Engineering, University of California, Los Angeles, California 90095, United States*

Gerassimos Orkoulas − *Department of Chemical Engineering, Widener University, Chester, Pennsylvania 19013, United States*

Complete contact information is available at:
https://pubs.acs.org/10.1021/acs.iecr.4c03150

Notes

The authors declare no competing financial interest.

ACKNOWLEDGMENTS

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

REFERENCES

(1) Li, Z.-J.; Ren, T.-L. In *Microsystems and Nanotechnology*; Zhou, Z., Wang, Z., Lin, L., Eds.; Springer Berlin Heidelberg: Berlin, Germany, 2012; pp 3−43.

(2) Ajayan, J.; Nirmal, D.; Tayal, S.; Bhattacharya, S.; Arivazhagan, L.; Fletcher, A. A.; Murugapandiyan, P.; Ajitha, D. Nanosheet field effect transistors-A next generation device to keep Moore's law alive: An intensive study. *Microelectron. J.* **2021**, *114*, 105141.

(3) Sun, C.; Rose, T.; Ehm, H.; Heilmayer, S. *Complexity Management in the Semiconductor Supply Chain and Manufacturing Using PROS Analysis*, Information and Knowledge Management in Complex Systems: Cham, Switzerland, 2015; pp 166−175.

(4) Orji, N. G.; Badaroglu, M.; Barnes, B. M.; Beitia, C.; Bunday, B. D.; Celano, U.; Kline, R. J.; Neisser, M.; Obeng, Y.; Vladar, A. E. Metrology for the next generation of semiconductor devices. *Nat. Electron.* **2018**, *1*, 532−547.

(5) Zheng, P.; Connelly, D.; Ding, F.; Liu, T.-J. K. FinFET Evolution Toward Stacked-Nanowire FET for CMOS Technology Scaling. *IEEE Trans. Electron Devices* **2015**, *62*, 3945−3950.

(6) Appenzeller, J.; Knoch, J.; Bjork, M. T.; Riel, H.; Schmid, H.; Riess, W. Toward Nanowire Electronics. *IEEE Trans. Electron Devices* **2008**, *55*, 2827−2845.

(7) Shao, G.; Jain, S.; Laroque, C.; Lee, L. H.; Lendermann, P.; Rose, O. Digital Twin for Smart Manufacturing: The Simulation Aspect. In *2019 Winter Simulation Conference (WSC)*: National Harbor: MD, USA, 2019; pp 2085−2098.

(8) Moyne, J.; Qamsane, Y.; Balta, E. C.; Kovalenko, I.; Faris, J.; Barton, K.; Tilbury, D. M. A Requirements Driven Digital Twin Framework: Specification and Opportunities. *IEEE Access* **2020**, *8*, 107781−107801.

(9) Kanarik, K. J.; Osowiecki, W. T.; Lu, Y. J.; Talukder, D.; Roschewsky, N.; Park, S. N.; Kamon, M.; Fried, D. M.; Gottscho, R. A. Human−machine collaboration for improving semiconductor process development. *Nature* **2023**, *616*, 707−711.

(10) Mei, Z.; Luo, Y.; Qiao, Y.; Chen, Y. A novel joint segmentation approach for wafer surface defect classification based on blended network structure. *J. Intell. Manuf.* **2024**, *35*, 1−15.

(11) Kadlec, P.; Gabrys, B.; Strandt, S. Data-driven soft sensors in the process industry. *Comput. Chem. Eng.* **2009**, *33*, 795−814.

(12) Jiang, Y.; Yin, S.; Dong, J.; Kaynak, O. A Review on Soft Sensors for Monitoring, Control, and Optimization of Industrial Processes. *IEEE Sensor. J.* **2021**, *21*, 12868−12881.

(13) Sun, Q.; Ge, Z. A Survey on Deep Learning for Data-Driven Soft Sensors. *IEEE Trans. Ind. Inf.* **2021**, *17*, 5853−5866.

(14) Chang, C.-C.; Pan, T.-H.; Wong, D. S.-H.; Jang, S.-S. A G&EWMA algorithm for high-mix semiconductor manufacturing processes. *J. Process Control* **2011**, *21*, 28−35.

(15) Zhang, C.; Yella, J.; Huang, Y.; Qian, X.; Petrov, S.; Rzhetsky, A.; Bom, S. Soft Sensing Transformer: Hundreds of Sensors are Worth a Single Word. In *2021 IEEE International Conference on Big Data (Big Data)*, 2021; pp 1999−2008.

(16) Correa, D. *Global Semiconductor Production Equipment Market Is Expected to Reach $209.9 Billion by 2031*, NASDAQ OMX's News Release Distribution Channel, 2023; Vol. *1*.

(17) Kanarik, K. J.; Lill, T.; Hudson, E. A.; Sriraman, S.; Tan, S.; Marks, J.; Vahedi, V.; Gottscho, R. A. Overview of atomic layer etching in the semiconductor industry. *J. Vac. Sci. Technol., A* **2015**, *33*, 020802.

(18) Wu, Z.; Tran, A.; Rincon, D.; Christofides, P. D. Machine Learning-Based Predictive Control of Nonlinear Processes. Part I: Theory. *AIChE J.* **2019**, *65*, No. e16729.

(19) Wang, H.; Tom, M.; Ou, F.; Orkoulas, G.; Christofides, P. D. Multiscale computational fluid dynamics modeling of an area-selective atomic layer deposition process using a discrete feed method. *Digit. Chem. Eng.* **2024**, *10*, 100140.

(20) Yun, S.; Tom, M.; Luo, J.; Orkoulas, G.; Christofides, P. Microscopic and Data-Driven Modeling and Operation of Thermal Atomic Layer Etching of Aluminum Oxide Thin Films. *Chem. Eng. Res. Des.* **2022**, *177*, 96−107.

(21) Moyne, J. In *Encyclopedia of Systems and Control*; Baillieul, J., Samad, T., Eds.; Springer London: London, 2015; pp 1248−1254.

(22) Tom, M.; Wang, H.; Ou, F.; Orkoulas, G.; Christofides, P. D. Machine Learning Modeling and Run-to-Run Control of an Area-Selective Atomic Layer Deposition Spatial Reactor. *Coatings* **2023**, *14*, 38−47.

(23) Tom, M.; Wang, H.; Ou, F.; Yun, S.; Orkoulas, G.; Christofides, P. D. Computational fluid dynamics modeling of a discrete feed atomic layer deposition reactor: Application to reactor design and operation. *Comput. Chem. Eng.* **2023**, *178*, 108400.

(24) *An Introduction to Kinetic Monte Carlo Simulations of Surface Reactions*; Jansen, A. P. J., Ed.; Academic Press, 2012; Vol. *1*, pp 38−119.

(25) Xie, W.; Parsons, G. N. Thermal atomic layer etching of metallic tungsten via oxidation and etch reaction mechanism using O_2 or O_3 for oxidation and WCl_6 as the chlorinating etchant. *J. Vac. Sci. Technol., A* **2020**, *38*, 022605.

(26) Ren, Y. M.; Alhajeri, M. S.; Luo, J.; Chen, S.; Abdullah, F.; Wu, Z.; Christofides, P. D. A tutorial review of neural network modeling approaches for model predictive control. *Comput. Chem. Eng.* **2022**, *165*, 107956.

(27) Lin, T.; Wang, Y.; Liu, X.; Qiu, X. A survey of transformers. *AI Open* **2022**, *3*, 111−132.

(28) Liu, Y.; Zhang, Y.; Wang, Y.; Hou, F.; Yuan, J.; Tian, J.; Zhang, Y.; Shi, Z.; Fan, J.; He, Z. A Survey of Visual Transformers. *IEEE Transact. Neural Networks Learn. Syst.* **2024**, *35*, 7478−7498.

(29) Zhao, W. X.; Zhou, K.; Li, J.; Tang, T.; Wang, X.; Hou, Y.; Min, Y.; Zhang, B.; Zhang, J.; Dong, Z.; Others A survey of large language models. **2023**, arXiv preprint arXiv:2303.18223.

(30) Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A. N.; Kaiser, Ł.; Polosukhin, I. Attention Is All You Need. *Advances in Neural Information Processing Systems*: Long Beach, CA, USA, 2017; pp 1−11.

(31) Devlin, J.; Chang, M.-W.; Lee, K.; Toutanova, K. *BERT: Pre-Training of Deep Bidirectional Transformers for Language Understanding*; North American Chapter of the Association for Computational Linguistics: Minneapolis, MN, USA, 2019; pp 4171−4186.

https://doi.org/10.1021/acs.iecr.4c03150
Ind. Eng. Chem. Res. 2024, 63, 19693−19706