Path-Sampling and Machine Learning for Rare Abnormal Events: Application to Polymerization CSTRs

Vikram Sudarshan, Warren D. Seider*, and Amish J. Patel

Department of Chemical and Biomolecular Engineering
University of Pennsylvania
Philadelphia, PA 19104-6393

Ulku G. Oktem
Near-Miss Management LLC
Philadelphia, PA 19103

Jeffrey E. Arbogast
Air Liquide, La Digital Factory
91 avenue Ledru Rollin
Paris, France 75011

Submitted to *Chem. Eng. Sci.*

ABSTRACT

Previously, we developed novel, unidirectional, static, multivariate, alarm systems for rare un-postulated abnormal events, demonstrated successfully for an exothermic CSTR. Herein, our techniques are improved significantly for a more-complex polystyrene CSTR, operating in its unstable region, capable of abnormal shifts to two undesirable regions; i.e., unsafe and unreliable regions. BG-FFS, a path-sampling algorithm, is utilized to simulate efficiently multiple rare abnormal trajectories; then, the XGBoost machine learning algorithm is utilized to develop accurate predictive models for committer probabilities; i.e., p_B as a function of key process variables – such models, when deployed in real-time, result in improved bidirectional dynamic multivariate alarm systems, capable of response actions using real-time p_B predictions. Then, using our rationalization strategies, the initial alarm systems are evaluated and modified, followed by DRAn (Dynamic Risk Analysis) studies and sensitivity analyses to investigate the effects of varying other process parameters to achieve more effective response actions.

Key words: BG-FFS (branched-growth forward-flux sampling), Machine Learning, XGBoost, Bidirectional Multivariate Alarm Systems, DRAn (Dynamic Risk Analysis).

*Corresponding author
Email address: seider@seas.upenn.edu (W.D. Seider)

1. INTRODUCTION AND BACKGROUND

Industry 4.0 is responsible for a transformational revolution across several industries, (Ghobakhloo, 2018; Suleiman et al., 2022; Vaidya et al., 2018; Yang and Gu, 2021), with novel, promising technologies and applications, including: the internet of things; i.e., IoT (Domova and Dagnino, 2017; Manavalan and Jayakrishna, 2019; Maqbool et al., 2023; Soori et al., 2023), artificial intelligence and machine learning (Candanedo et al., 2018; Dingli et al., 2021; Javaid et al., 2022; Lee and Lim, 2021; Lee et al., 2018; Rai et al., 2021), big data analytics (Gokalp et al., 2016; Nguyen et al., 2020; Yan et al., 2017), cybersecurity and cyber-physical systems (Culot et al., 2019; Ervural and Ervural, 2018; Lezzi et al., 2018), and the like. Given the success of Industry 4.0, discussions are already in place regarding Industry 5.0, consisting of more-advanced technologies such as robotics and human-robot interactions (Barata and Kayser, 2023; Demir et al., 2019; Ghobakhloo et al., 2023). However, despite such significant advancements, a major concern of the chemical and manufacturing industries is to ensure safe and reliable operation of their processes. Unsafe, extreme operating conditions (e.g., high temperatures or pressures leading to thermal runaway reactions), poor handling of hazardous or highly reactive chemicals (e.g., improper cleaning of reactors and storage tanks), and the like, can result in devastating consequences for operators, equipment, and the environment. Additionally, while automated Safety Instrumented Systems (SIS) prevent accidents by shutting down plants that approach unsafe operating regions, they could contribute to poor plant reliability through production losses due to shutdowns, maintenance, and start-up delays. With these advances, motivation grows to improve techniques that mitigate unsafe and unreliable situations resulting from such abnormal events.

1.1. Rare and Undesirable Safety Abnormal Events

However, these abnormal events are extremely *rare* and *undesirable*, with very little occurrence data available to anticipate such events. Additionally, while near-miss analyses are informative and available extensively, they may have limitations in quantifying accurately the likelihood of such rare abnormal events. Often, rare abnormal events with disastrous consequences are highly unanticipated and *un-postulated* (i.e., do not take a specific shape or form, cannot be postulated in HAZOP studies, and occur randomly). Analyses of rare and extreme events have been conducted across multiple research domains, including: molecular dynamics (Berne, 1985; Ciccotti and Ferrario, 2000; Sarich et al., 2014; Shivpuje et al., 2019), economics and finance (Jalali et al., 2010; Stanley et al., 2007), climate modeling (Beniston et al., 2007; Webber et al., 2019), medical research (Bhaumik et al., 2012; Cai et al., 2010; Donnenberg and Donnenberg, 2007), and the like.

Within chemical process safety too, extensive research has been conducted for quantitative estimation of rare-events. From the perspective of accident modeling, often, rare events are defined as low-frequency high-consequence events (Aven, 2020). First, a novel method was introduced for probability estimation of rare events, using maximum-likelihood maximum-entropy principles, particularly for historical data not containing any occurrences of rare-events (Ahooyi Mohseni et al., 2014). Next, a precursor-based hierarchical Bayesian framework was developed for estimating frequencies of rare-events, as well as consequence analyses of these events, demonstrated on the BP Deepwater Horizon accident case-study (Yang et al., 2013, 2015). Given the key limitation of Bayesian models in considering source-

to-source variability in data, a new framework was developed that conducts root-cause analysis based on deviations in key process variables, demonstrated on the Tennessee Eastman process (Kumari et al., 2020). Additionally, several machine learning-based approaches have been introduced – an approach was developed to leverage advancements in data science and machine learning to build predictive models for severity prediction using accident precursor data (Tamascelli et al., 2022). Additionally, a parametric reduced-order modeling framework was introduced for consequence estimation of rare-events, based on the k-Nearest Neighbors machine learning algorithm, demonstrated on a cardon dioxide release case study (Kumari et al., 2021). In related work, optimized machine-learning algorithms were applied to predict accident outcomes using occupational accident data (Sarkar et al., 2019). However, most rare event analysis methods that are purely data-driven, relying solely on historical data, have limitations due to the scarcity of rare-event data. **Hence, to circumvent this, it is crucial to complement data-driven techniques with simulation-based techniques, capable of simulating the various pathways indicative of rare unsafe or unreliable abnormal events, in unanticipated and <u>un-postulated</u> ways, especially in cases with very limited historical occurrences; e.g., when utilizing Markov Chain-Monte Carlo (MCMC)-based techniques.**

1.2. Path-Sampling Algorithms from Molecular Dynamics

In molecular dynamics (MD), path-sampling algorithms are MCMC-based techniques applied routinely to explore, simulate, and quantify rare events computationally in stochastic nonequilibrium systems (Allen et al., 2006), crystal nucleation of hard spheres (Filion et al., 2010), methane hydrate nucleation (Arjun and Bolhuis, 2023; Bi and Li, 2014), and nucleation of sodium chloride crystals (Jiang et al., 2018). In MD, a rare event is an event whose initiation time (i.e., time taken to initiate the rare event) is multiple orders of magnitude greater than its duration(Borrero and Escobedo, 2007; Hartmann et al., 2014).

Previously, in our research, the application of two specific algorithms: transition path-sampling (TPS) and forward-flux sampling (FFS), was explored. As a first approach toward combining chemical process modeling with MD-based path-sampling to investigate rare abnormal events, we explored the application of TPS for the operation of an exothermic CSTR and an air separation unit (ASU) (Moskowitz et al., 2018). Then, to improve on the limitations of the TPS approach, we explored the application of FFS to simulate efficiently and analyze rare abnormal events resulting from random perturbations in one or more process variables (Sudarshan et al., 2021). Our analyses were demonstrated successfully for a relatively simple exothermic CSTR and resulted in numerous transition pathways between the desirable and undesirable regions, simulated efficiently in a piecewise manner. **However, while path-sampling algorithms are capable of simulating the reaction pathways and quantifying the probabilities and rates of rare abnormal events, the data generated during the simulations need to be analyzed carefully – hence, it is crucial to develop accurate predictive models that leverage the data generated during the simulations.**

1.3. Predictive Modeling and Analytics, Machine Learning and Artificial Intelligence, Decision Sciences

Predictive modeling and analytics are crucial to advance scientific research, enabling researchers to analyze data of varying quantities and explore patterns or insights not previously envisioned. These powerful techniques have transformed our understanding of data information, resulting in significant breakthroughs across scientific domains. Machine learning (ML) serves as one of the fundamental pillars of predictive analytics. A subset of the vast and rapidly evolving field of artificial intelligence (AI), ML refers to advanced statistical algorithms that generalize effectively from data and perform tasks without being explicitly programmed (Koza et al., 1996; Sarker, 2021; Wang et al., 2009). Compared to traditional statistical methods that focus more on inference, ML focuses on predictions by using general-purpose learning algorithms, enabling one to find patterns not envisioned previously, especially in rich and unwieldy data (Bzdok, 2017; Bzdok et al., 2018, 2017) – this led to an exponential increase in ML-based research across several scientific domains in the past decade.

Within chemical engineering too, as part of Industry 4.0, numerous research contributions that leverage advanced algorithms in the ML ecosystem have been conducted, such as for: catalysis (Kitchin, 2018; Toyao et al., 2020), computational fluid dynamics (Hanna et al., 2020; Kochkov et al., 2021), process monitoring and fault detection (Angelopoulos et al., 2020; Arunthavanathan et al., 2022; Tran et al., 2021), smart manufacturing and predictive maintenance (Çınar et al., 2020; Kotsiopoulos et al., 2021), to name a few. More recently, in line with the rapid increase in research on generative AI and large language models (LLMs), a novel time-series transformer (TST) - based model-predictive control framework was developed, demonstrated for a batch crystallization system (Sitapure and Kwon, 2023a, 2023b). In related work, a generative transformer model was developed for autocompletion of process flowsheets, (Vogel et al., 2023). While the prospects of ML-based techniques and applications are promising, limitations must be addressed. For instance, complex ML models, including LLMs, involve several parameters, requiring extensive computational resources. Additionally, from amongst enumerable ML algorithms, selection of relevant algorithms suitable for target applications is crucial, with model transparency and interpretability also a key concern (Carvalho et al., 2019).

Note that "decision science" is an umbrella term, encompassing interdisciplinary quantitative techniques (e.g., risk assessment, machine learning, optimization, and the like), utilized to analyze, quantify, and improve decision-making across scientific domains (Kleinmuntz, 1990; Sharda et al., 2021). For instance, given predictions or forecasts by ML-based models, what are the actionable strategies that can be taken to achieve the specific goals and objectives. **However, given the increased focus on developing more-accurate ML algorithms in literature, this decision science aspect of ML is unlikely to be addressed adequately. It seems clear that response actions taken by leveraging real-time predictions from ML models need to be addressed more carefully (Varshney and Alemzadeh, 2017) – a key focus of this paper.**

1.4. Multivariate Alarm Systems for Rare, Un-postulated Abnormal Events

Based on our findings from initial analyses of TPS and FFS algorithms, we introduced novel, improved, multivariate alarm systems to mitigate rare un-postulated abnormal events, resulting from random perturbations (i.e., statistical noise) in one or more process variables, demonstrated for an exothermic CSTR (Sudarshan et al., 2023). The branched-growth forward-flux sampling algorithm (BG-

FFS) was applied to locate rare trajectories that proceed from the high-conversion, high-temperature basin *A* to the low-conversion, low-temperature basin *B*, in an efficient and piecewise manner. It yielded variable values at discrete *crossing points*, with committer probabilities, p_B, computed at the crossing points, enabling regression to yield a linear exponential model for p_B as a function of the key process variables. This p_B model was utilized to develop a multivariate alarm system, consisting of initial guesses for process variable limits associated with each p_B threshold, and their response actions. Then, to evaluate the quality and effectiveness of the multivariate alarm systems, we introduced improved rationalization strategies, with the alarm thresholds and response actions modified using key statistical metrics (Sudarshan et al., 2024). For the exothermic CSTR, our strategies resulted in a significant reduction in total alarms annunciated, with key focus on quality alarms, which, if ignored, were more likely to result in shifts to undesirable operating regions. Finally, to evaluate the risk associated with the rationalized multivariate alarm systems, we utilized dynamic risk analysis (DRAn), wherein, using Bayesian statistics, probability distributions were constructed for the failure probabilities of the alarm systems, estimated using multiple dynamic simulations. But, these multivariate alarm systems consisted of static alarm thresholds (i.e., based on static process variable limits) obtained using simple OLS-based linear exponential models for p_B. Herein, this and other limitations are eliminated, as summarized next.

1.5. Polymerization CSTR– Unsafe and Unreliable Operating Regions – Real-time p_B Models

In this manuscript, we extend our analyses to more complex polymerization CSTRs, particularly, in a PID-controlled polystyrene CSTR operating in its intermediate *unstable* operating region, considered desirable for polymerization reactors. To model rare un-postulated abnormal events, random, statistical *noise* is introduced into the monomer feed concentration. For the polystyrene CSTR operating in the unstable region, adequate statistical noise can lead to rare un-postulated abnormal shifts to only two possible regions; i.e., *unsafe* or *unreliable*.

For data generation, the BG-FFS algorithm is utilized to simulate independently rare abnormal trajectories between desirable (i.e., unstable) and undesirable (i.e., unsafe or unreliable) operating regions, saving key process variable values at distinct crossing points, followed by computation of p_B at all crossing points. Then, to develop p_B-process variable models in real-time, advanced ML algorithms are used to obtain accurate, predictive models for dynamic, multivariate alarm systems. Next, our rationalization strategies, developed previously (Sudarshan et al., 2024), are used to improve alarm thresholds and response actions. Finally, using DRAn, failure probabilities of the safety systems and their probability distributions are estimated, for the rationalized alarm systems, using multiple dynamic simulations, followed by sensitivity analyses to investigate the effects of varying other process parameters in response to alarms. **Hence, given the response action strategies developed as part of the multivariate alarm systems that are based on real-time p_B predictions, this paper addresses the decision science component of accident modeling, risk assessment, as well as machine learning.**

2. MATERIALS AND METHODS

2.1. Description of Polystyrene CSTR Model

Figure 1 shows a temperature-controlled polystyrene CSTR that manipulates the coolant flowrate. Polymerization CSTRs have been studied extensively, including continuous polymerization of vinyl chloride and styrene (Brooks, 1981). Additionally, the homopolymerization of styrene in a series of two CSTRs was investigated (Kim et al., 1991). Then, a nonlinear model-predictive control (NMPC) algorithm was developed for control of a water-cooled styrene polymerization CSTR at its unstable steady-state (Hidalgo and Brosilow, 1990). In related work, an approach was proposed to verify the stability and performance of controllers for a polystyrene CSTR in the presence of uncertainty (Gazi et al., 1996). Additionally, steady-state multiplicity and bifurcation analyses were conducted to study its operability; i.e., how the process dynamics varies with operating conditions and process parameters at steady-state (Russo and Bequette, 1998).

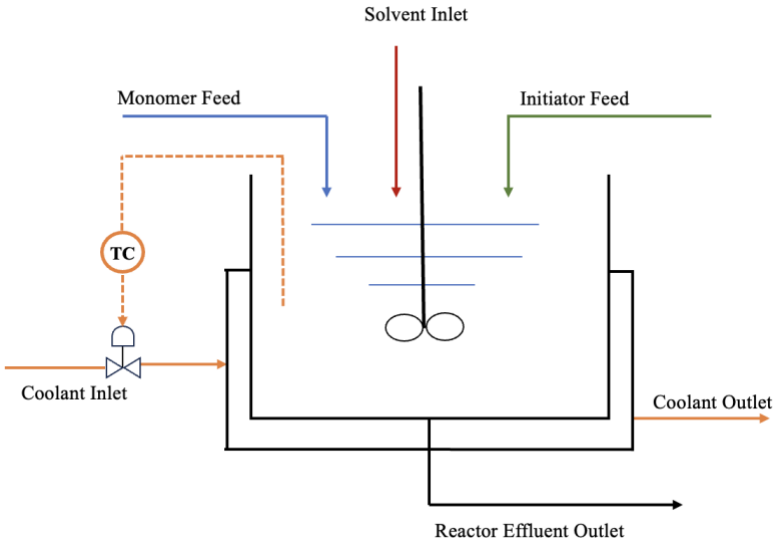


Figure 1. Schematic of the polystyrene CSTR process model

For homopolymerization of styrene, Hidalgo and Brosilow (1990) and Russo and Bequette (1998) utilized a free-radical polymerization kinetic mechanism:

$$\mathrm{I} \xrightarrow{k_\mathrm{d}} 2\mathrm{R} \qquad \text{(initiator decomposition)} \tag{1}$$

$$\mathrm{M} + \mathrm{R} \xrightarrow{k_\mathrm{i}} \mathrm{P}_1 \qquad \text{(chain initiation)} \tag{2}$$

$$\mathrm{P}_n + \mathrm{M} \xrightarrow{k_\mathrm{p}} \mathrm{P}_{n+1} \qquad \text{(chain propagation)} \tag{3}$$

$$\mathrm{P}_n + \mathrm{P}_m \xrightarrow{k_\mathrm{td}} \mathrm{T}_n + \mathrm{T}_m \qquad \text{(termination} - \text{by disproportionation)} \tag{4}$$

$$\mathrm{P}_n + \mathrm{P}_m \xrightarrow{k_\mathrm{tc}} \mathrm{T}_{n+m} \qquad \text{(termination} - \text{by coupling)} \tag{5}$$

where I is the initiator (i.e., azobisisobutyronitrile – AIBN), M is the monomer (i.e., styrene), R is the free radical produced by initiator decomposition, P represents the growing polymer chain, T is the terminated polymer chain, and k_d, k_i, k_p, k_{td}, k_{tc} are the rate constants for initiator decomposition, chain initiation, propagation, termination by disproportionation, and termination by coupling. The model assumptions, as mentioned in Hidalgo and Brosilow (1990) and Russo and Bequette (1998), are:

i) The lifetimes of the radical species are short compared to the system time constants; hence, the quasi steady-state approximation is valid (i.e., net rates of reaction associated with the radicals ~ 0).
ii) The CSTR is assumed to be well-mixed.
iii) The physical properties of the reaction mixture (e.g., density, heat capacity, heat transfer coefficient, etc.) are assumed constant.
iv) Gel effect; i.e., sudden increase in the overall polymerization rate due to increases in the viscosity of the reaction mixture, thereby reducing the termination rate, is assumed to be negligible.
v) The sum of the Arrhenius rate expressions for termination by coupling and disproportionation is expressed as a single overall chain termination rate (i.e., $k_t = k_{tc} + k_{td}$)

With these assumptions, the material and energy balance equations for the *dimensionless* model, as developed by Russo and Bequette (1998), are:

$$\frac{dx_1}{d\tau} = q_i x_{1f} - (q_i + q_m + q_s)x_1 - \phi_d \kappa_d(x_3) x_1 \tag{6}$$

$$\frac{dx_2}{d\tau} = q_m x_{2f} - (q_i + q_m + q_s)x_2 - \phi_p \kappa_p(x_3) x_2 x_5 \tag{7}$$

$$\frac{dx_3}{d\tau} = (q_i + q_m + q_s)(x_{3f} - x_3) + \beta\phi_p \kappa_p(x_3) x_2 x_5 - \delta(x_3 - x_4) \tag{8}$$

$$\frac{dx_4}{d\tau} = \delta_1[q_c(x_{4f} - x_4) + \delta\delta_2(x_3 - x_4)] \tag{9}$$

$$x_5 = \sqrt{\frac{2f\phi_d \kappa_d(x_3) x_1}{\phi_t \kappa_t(x_3)}} \tag{10}$$

$$\kappa_d(x_3) = \exp\left(\frac{\gamma_d x_3}{1 + \frac{x_3}{\gamma_p}}\right) \tag{11}$$

$$\kappa_t(x_3) = \exp\left(\frac{\gamma_t x_3}{1 + \frac{x_3}{\gamma_p}}\right) \tag{12}$$

$$\kappa_p(x_3) = \exp\left(\frac{x_3}{1 + \frac{x_3}{\gamma_p}}\right) \tag{13}$$

where x_1, x_2, x_3, x_4, x_5 are the dimensionless initiator concentration, monomer concentration, reactor temperature, coolant temperature, and concentration of growing polymer; q_i, q_m, q_s, q_c are the

dimensionless flow rates for initiator, monomer, solvent and coolant streams; $\phi_{\mathrm{d}}, \phi_{\mathrm{p}}, \phi_{\mathrm{t}}$ are Damkohler numbers for initiator decomposition, propagation, and termination; $\gamma_{\mathrm{d}}, \gamma_{\mathrm{p}}, \gamma_{\mathrm{t}}$ are dimensionless activation energies for initiator decomposition, propagation, and termination; β is the dimensionless heat of reaction, δ is the dimensionless heat-transfer coefficient, δ_1 is the dimensionless reactor volume, δ_2 is the dimensionless specific heat, f is the initiator efficiency; $x_{1\mathrm{f}}$, $x_{2\mathrm{f}}$, $x_{3\mathrm{f}}$, $x_{4\mathrm{f}}$ are the dimensionless initiator feed concentration, monomer feed concentration, reactor feed temperature, and coolant feed temperature. Table 1 displays the values for the dimensionless process parameters. Next, based on studies conducted by Russo and Bequette, (1998), steady-state bifurcation analysis is performed, with the dimensionless cooling-water flow rate; i.e., q_c, selected as the free bifurcation parameter, enabling identification of desirable and undesirable operating regions. Looking ahead, in Section 3, Results and Discussion, see Figure 6.

Table 1. Dimensionless Process Parameters

Parameter	**Value**
q_{i}	0.1
q_{m}	0.4
q_{s}	0.48571
ϕ_{d}	0.01688
ϕ_{p}	2.1956×10^{7}
ϕ_{t}	9.6583×10^{12}
$x_{1\mathrm{f}}$	0.06769
$x_{2\mathrm{f}}$	1.0
$x_{3\mathrm{f}}$	0.0
$x_{4\mathrm{f}}$	-1.5
δ	0.74074
δ_1	0.90569
δ_2	0.37256
β	13.17936
f	0.6
$x_{3,\mathrm{sp}}$	0.85
K_{c}	50
τ_{D}	0.9
τ_{I}	5

2.2. Overview of Key Steps and Methods

Figure 2 represents an overview of the key steps and methods utilized in this paper, with each step described extensively in subsequent sections.

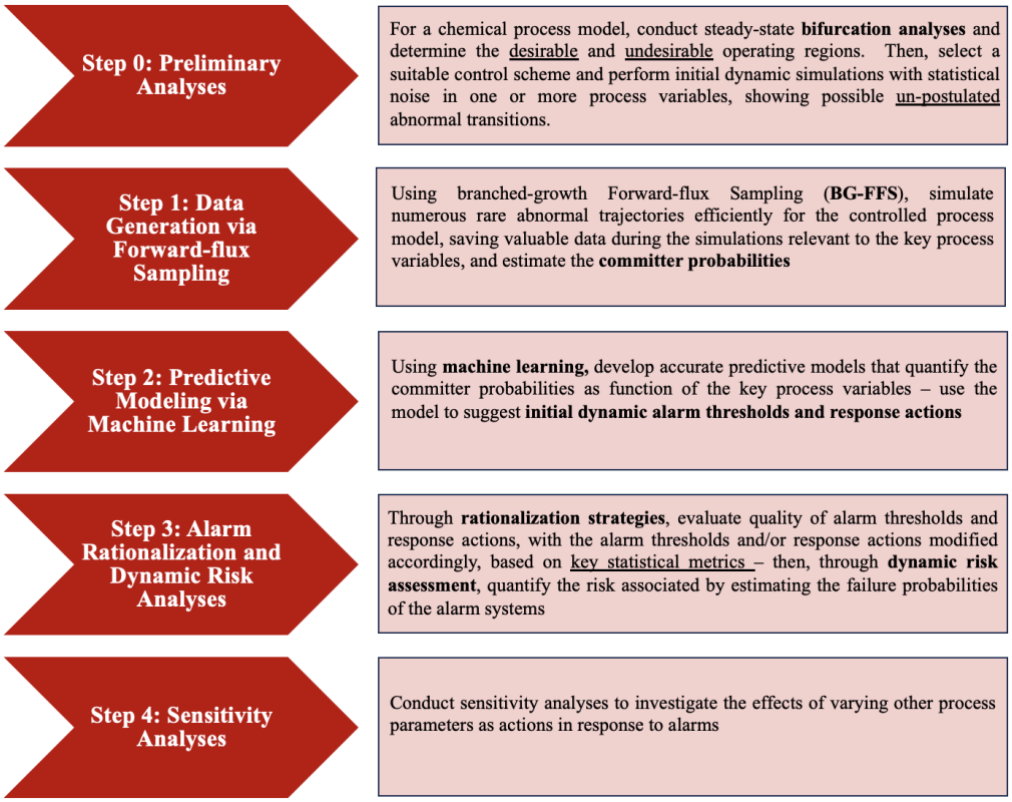


Figure 2. Overview of key steps utilized in developing multivariate alarms

2.3. Step 0: Preliminary Analyses

As part of preliminary studies, steady-state bifurcation analysis is conducted for the polystyrene CSTR, with q_c chosen as the free bifurcation parameter – from the resulting solution diagrams, stable and unstable branches are identified, based on which the desirable and undesirable operating regions are determined. Then, an appropriate control scheme, along with controller parameters, is selected experimentally, to ensure sufficiently tight control, while allowing rare abnormal shifts to the undesirable operating regions. Given the desirable and undesirable operating regions, initial dynamic simulations are conducted inclusive of control and statistical noise in one or more process variables, recognizing rare un-postulated abnormal shifts, when they occur.

2.4. Step 1: Data Generation via Forward-flux Sampling

FFS conducts simulations of the process trajectories in a discrete, piecewise manner, using evenly or unevenly placed interfaces; i.e., λ_i, $i = 0,1, 2, \ldots, n$, where λ is an appropriate order parameter variable (e.g., temperature), and, λ_0 and λ_n represent the bounds for basins A and B. Initially, we utilized the direct

forward-flux sampling variant (DFFS) for the exothermic CSTR process model to simulate numerous rare abnormal trajectories between the desirable *basin A* and undesirable *basin B*, thereby, computing the associated transition probabilities and rates. However, the DFFS variant is limited, because it simulates piecewise trajectories from *random* crossing points across each interface. For accurate computations of the committer probabilities, it is desirable to simulate multiple trajectories from *every* crossing point at λ_i, as implemented in the BG-FFS algorithm (Allen et al., 2009; Borrero and Escobedo, 2007), demonstrated when developing the multivariate alarm systems for the exothermic CSTR (Sudarshan et al., 2023). The steps involved in the BG-FFS algorithm, shown schematically in Figure 3, are:

i) Define the initial desirable basin A and terminal undesirable basin B.
ii) Pick a suitable order parameter variable; i.e., λ; typically, this is a process variable that has a strong influence on the process dynamics; i.e., is able to capture process deviations more-rapidly than other variables, and is not perturbed significantly using statistical noise; e.g., the reactor temperature.
iii) Based on the chosen λ, divide the space between the two basins into finite interfaces; i.e., $\lambda_0, \lambda_1 \ldots \lambda_\mathrm{n}$, where λ_0 and λ_n represent the bounds for basins A and B.
iv) Simulate a long initial trajectory that generates finite crossings across λ_0; if required, repeat this step for multiple trajectories to generate sufficient crossing points, with all process variables saved at every crossing point.
v) Compute the initial rate of transition across λ_0, r_0, as the total crossings divided by the total time spent in basin A by all the initial trajectories.
vi) Select a crossing point from among the saved crossings across λ_0 and simulate m_0 trajectories from that point, each of which continues until λ_1 is crossed. Save the variables at all such crossing points.
vii) Simulate k_1 trajectories from every crossing point across λ_1 that generate crossing points across λ_2. Save the variables at all such crossing points.
viii) Iterate step vii) for all subsequent interfaces till λ_n; i.e., simulate m_i trajectories from all crossing points at λ_i that continue until λ_{i+1} is reached; save the variables at all such crossing points at λ_{i+1}; $\forall\ i = 2, 3 \ldots n-1$.
ix) Compute the overall transition probability of reaching basin B from basin A:

$$p_{\mathrm{A}\to\mathrm{B}}(\lambda_n|\lambda_0) = \frac{N(\lambda_n|\lambda_0)}{\prod_{i=0}^{n-1} m_i} \tag{14}$$

where $N(\lambda_n|\lambda_0)$ is the number of branches that reach basin B (i.e., from λ_{n-1}) and $\prod_{i=0}^{n-1} m_i$ are the total possible number of branches.
x) Compute the overall rate of transition, $r_{\mathrm{A}\to\mathrm{B}}$, as the product of r_0 and $p_{\mathrm{A}\to\mathrm{B}}(\lambda_n|\lambda_0)$.
xi) Repeat steps iv) – ix) for other crossing points at λ_0 and compute the average overall probability and rate of transition, i.e., $\bar{p}_{\mathrm{A}\to\mathrm{B}}$ and $\bar{r}_{\mathrm{A}\to\mathrm{B}}$.

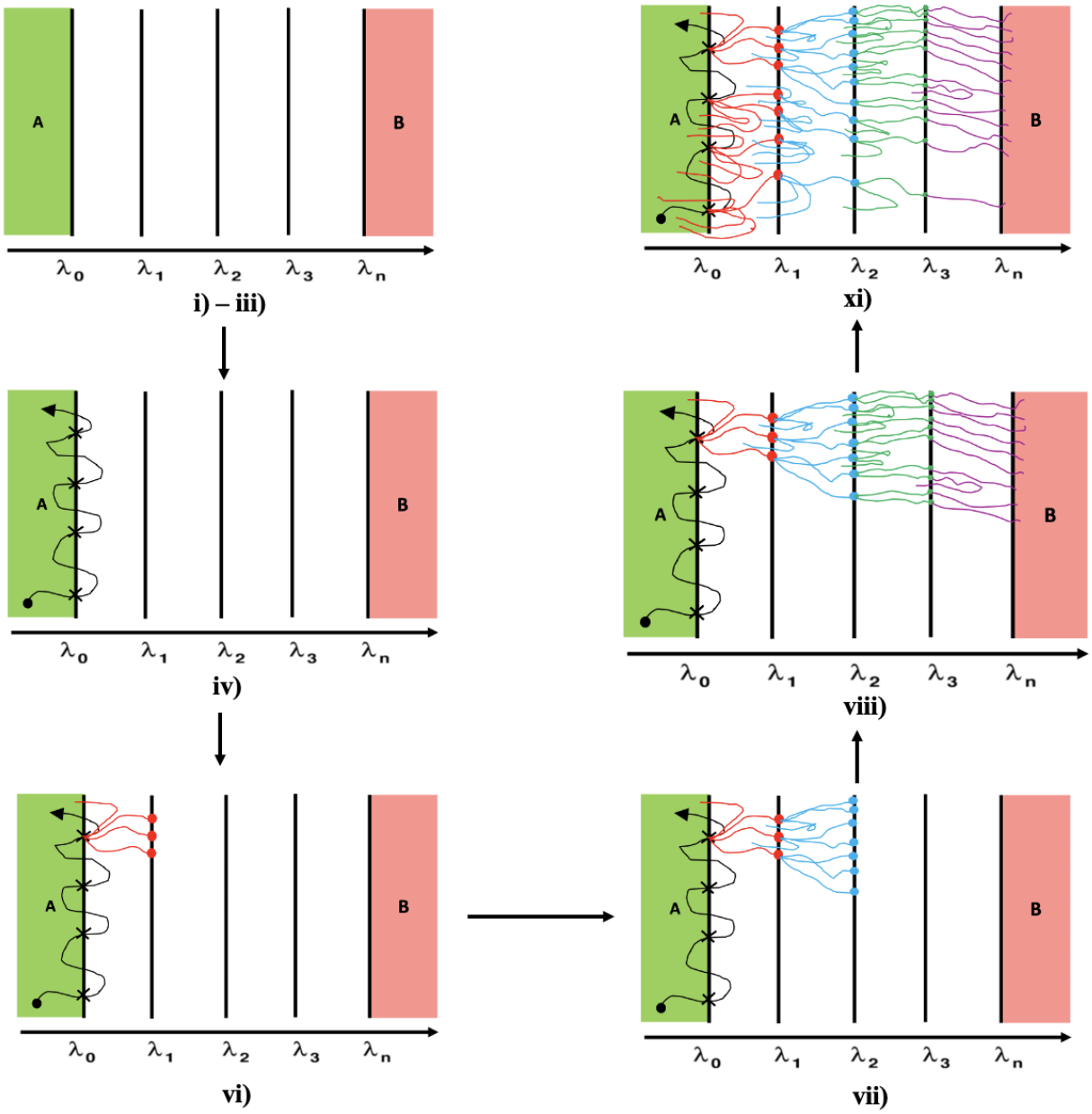


Figure 3. Schematic showing key steps for simulating abnormal trajectories using BG-FFS algorithm (refer to the points in Section 2.4.)

Given multiple rare abnormal trajectories simulated efficiently and values of the key process variables saved, the BG-FFS simulations are conducted only a few times *offline* and need not be repeated. Note that every stored crossing point, with variables x, has an associated *committer probability*, $p_B(x)$ — defined as the probability that a trajectory fired from that point reaches; i.e., 'commits' to, basin B (Allen et al., 2009; Borrero and Escobedo, 2007; Peters and Trout, 2006). To estimate p_B for a crossing point across λ_i, the variables at that point and the number of trajectories initiated from that point that successfully

crossed λ_{i+1}, need to be saved. Then, the probability that a trajectory initiated from a crossing point j at λ_i reaches the next interface; i.e., λ_{i+1}, is (Borrero and Escobedo, 2007; Sudarshan et al., 2023):

$$p_j{}^i(\lambda_{i+1}|\lambda_i) = \frac{N_j{}^i}{m_i} \tag{15}$$

where $N_j{}^i$ is the number of successful trajectories reaching λ_{i+1} from point j at λ_i, and m_i is the total number of trajectories initiated from that point. Next, the $p_j{}^i(\lambda_{i+1}|\lambda_i)$ can be used to compute the committer probability to reach basin B of point j at λ_i, $p_{Bj}{}^i$:

$$p_{\mathrm{B}j}{}^i = \; p_j{}^i(\lambda_{i+1}|\lambda_i) \times \frac{\sum_{k=1}^{N_j{}^i} p_{\mathrm{B}k}{}^{i+1}}{N_j{}^i} = \frac{\sum_{k=1}^{N_j{}^i} p_{\mathrm{B}k}{}^{i+1}}{m_i}; i = N_{\text{interfaces}} - 1, N_{\text{interfaces}} - 2, \ldots 0 \tag{16}$$

Note that Eqs. (15) and (16) are recursive formulae – stated differently, the committer probabilities are computed for all crossing points generated during BG-FFS simulations in reverse, i.e., from i = $N_{\text{interfaces}}$ - 1 to i = 0, where $N_{\text{interfaces}}$ is the number of interfaces. For instance, consider a simple example for demonstration purposes (Sudarshan et al., 2023): assuming $N_{\text{interfaces}}$= 7, $p_{Bj}{}^7$ = 1, because at the last interface, it is concluded that the process has transitioned to basin B. Next, $p_{Bj}{}^6 = p_j{}^6(\lambda_7|\lambda_6) = \frac{N_j{}^6}{m_6}$, since $\sum_{k=1}^{N_j{}^6} p_{Bk}{}^7 = N_j{}^6$. Next, $p_{Bj}{}^5$ can be computed using the previously computed $p_{Bj}{}^6$. This process is continued until $p_{Bj}{}^0$ — in this way, the committer probabilities are computed for all crossing points obtained during the BG-FFS simulations.

Additionally, note that depending on the process parameter selected initially as the *response-action variable* (i.e., a variable that is varied in real-time in response to alarms; e.g., q_i), the BG-FFS simulations and p_B calculations are repeated for multiple discrete values of the response-action variable.

2.5. Step 2: Predictive Modeling via Machine Learning

Figure 4 shows the pipeline encompassing all key stages for developing predictive models using machine learning. With data generated during BG-FFS simulations (Figure 4, step I) pertaining to the key process variables and the estimated p_B, followed by data preprocessing; i.e., Figure 4, step II (e.g., eliminating extreme outliers for crossing points generated across each λ_i), the next step is to develop a predictive ML model that quantifies p_B as a function of the key process variables (Figure 4, steps III-V). Note – next, Figure 4, step III, is introduced briefly – followed by discussions of Figure 4, steps IV and V.

Recently, Sudarshan et al. (2023), when developing a multivariate alarm system for an exothermic CSTR, created a simple, linear-exponential predictive model, with the coefficients estimated using ordinary least-squares (OLS) regression:

$$\widehat{p_{\mathrm{B}}} = \; 0.81\left[exp\left(-8.3T_{\text{dim}} + \; 4.64F_{\text{c,dim}} + \; 0.04T_{\text{c,dim}} + \; 0.04C_{\text{dim}}\right)\right] \tag{17}$$

where T_{dim} is the dimensionless temperature; $F_{c,dim}$ is the dimensionless CW flow rate; $T_{c,dim}$ is the dimensionless CW temperature, and C_{dim} is the dimensionless reactant concentration. However, for the polystyrene CSTR, as changes to the process are made in response to alarms (e.g., varying q_i), the process dynamics also varies significantly, unable to be captured by *parametric* models with fixed mathematical structure (e.g., linear-exponential models), leading to *inaccurate* p_B estimates and *incorrect* response actions. Additionally, it is often cumbersome and inconvenient to experiment with different parametric models, one-by-one, and evaluate their applicability to the data.

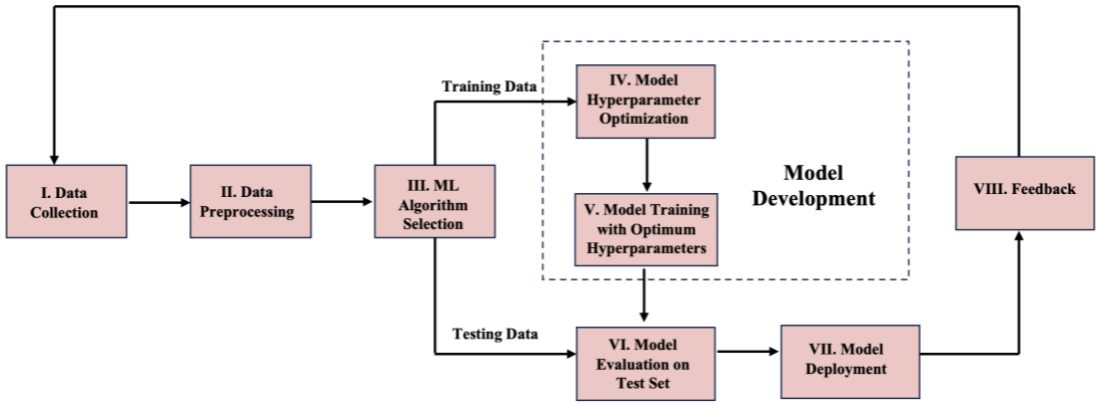


Figure 4. Pipeline encompassing all key stages in machine learning.

Machine learning algorithms that do not assume structure, as in Eq. (17), are referred to as *nonparametric* learning algorithms. Such algorithms are free to learn any functional form, based on the training data provided. Typically, the training data in our analyses are *tabular*, as shown in Table 2 for N data points and four input variables. Among various nonparametric algorithms, tree-based learning algorithms are preferred for tabular data, as clarified in several benchmark studies comparing ML algorithms for multiple datasets (Grinsztajn et al., 2022; Shwartz-Ziv and Armon, 2022). Most notably, despite their revolutionary advances in image (e.g., computer vision) and text [e.g., NLP (Natural Language Processing), LLMs (Large Language Models)] processing applications, deep neural networks appear to be outperformed by tree-based models for tabular data (Borisov et al., 2022; Grinsztajn et al., 2022).

Table 2. Schematic for Tabular Training Data in our Analyses

$\boldsymbol{p_B}$	$\boldsymbol{X_1}$	$\boldsymbol{X_2}$	$\boldsymbol{X_3}$	$\boldsymbol{X_4}$
$p_{B,1}$	$X_{1,1}$	$X_{2,1}$	$X_{3,1}$	$X_{4,1}$
$p_{B,2}$	$X_{1,2}$	$X_{2,2}$	$X_{3,2}$	$X_{4,2}$
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
$p_{B,N_{samples}}$	$X_{1,N_{samples}}$	$X_{2,N_{samples}}$	$X_{3,N_{samples}}$	$X_{4,N_{samples}}$

XGBoost is a tree-based algorithm that has become popular since its development in 2014. It is very effective for classification and regression tasks (Cerna et al., 2020; Li et al., 2019; Ma et al., 2021; Ogunleye and Wang, 2020), including classification of rare events (Ashraf et al., 2023; Wang et al., 2023). XGBoost, named for "eXtreme Gradient Boosting", as developed by Chen and Guestrin (2016), is an *ensemble* learning algorithm, developed to provide high-predictive performance with efficient computational speed, capable of CPU (central processing unit) and GPU (graphical processing unit) parallelization. Typically, XGBoost trains an ensemble of decision trees *sequentially*, wherein, each newly trained decision tree attempts to reduce the errors, w.r.t training data, in the previous tree, involving computations of *gradients* for a loss function (e.g., the RMSE in Eq.(28)). Stated differently, the first decision tree (i.e., the *weak learner*) in the ensemble generates inadequate predictions, but through steady and sequential *boosting* (i.e., achieving improvements), XGBoost results in a powerful predictive model that represents complex relationships in the data, generating highly accurate predictions. Hence, herein, for ML algorithm selection in Figure 4, step III, XGBoost is selected as the preferred algorithm.

Figure 5 shows a schematic of a decision tree involving an example dataset created for demonstration purposes only, containing just 10 samples ($N_{samples}$ = 10), to predict p_B as a function of temperature, T. For this dataset, Table 3 shows p_B and T for each of the samples. At each iteration, the optimum split threshold for T is computed using the variance reduction method described in Breiman et al. (2017). On this basis, the data is split further, with the splitting process terminated when insufficient data remain, after which the average p_B is returned. For instance, at the first split, the optimum split threshold for T is computed as T = 480 K – then, the data are divided into two sets; i.e., 5 samples for which, T <= 480 K, and 5 samples for which, T > 480 K. The splitting process continues for both sets until the number of samples remaining in a set is <= $N_{samples, min}$ ($N_{samples, min}$ is the minimum number of samples required in a set to continue splitting; e.g., in Figure 5, $N_{samples, min}$ = 3) thereby, returning the average p_B for that set. To check for consistency with the data in Table 3, note the average p_B values returned at the end of the decision tree in Figure 5 – for instance, there are 2 samples for which, T <= 360 (i.e., T = {300, 340}; p_B = {0.1, 0.2}), with the average p_B for these samples = 0.15, consistent with Figure 5. Similarly, there are 3 samples for which, 360 < T <= 480 (i.e., T = {380, 420, 460}; p_B = {0.3, 0.4, 0.5}), with their average p_B = 0.4. Note that XGBoost involves training several decision trees sequentially, where each decision tree follows a splitting process similar to the one described for the example decision tree in Figure 5. For a detailed description of the XGBoost algorithm, refer to the XGBoost Algorithm: Steps Subsection 7.2 in the Appendix, and to the official document for the algorithm (Chen and Guestrin, 2016).

Note that prior to model development, the data are divided into *training* and *testing* data using *randomized* splits. Additionally, note that most ML models consist of two entities: hyperparameters to be set/optimized before training; and, internal model parameters that are optimized during training. In general, the XGBoost model consists of several hyperparameters (e.g., one of which is the number of decision trees; note – Figure 5 could be the first decision tree) that need to be chosen carefully, as model performance is extremely sensitive to their choice. Hence, for the XGBoost algorithm, model development consists of two key steps:

i) Hyperparameter Optimization with Cross-Validation (Figure 4, step IV): This involves computing the optimum set of hyperparameters for the ML model across multiple subsets of training data to ensure

generalization and robustness. There are several open-source software packages available for hyperparameter optimization, including: Hyperopt (Bergstra et al., 2015), Optuna (Akiba et al., 2019), Ray tune (Liaw et al., 2018), Optunity (Claesen et al., 2014), and the like. For the XGBoost model herein, the Optuna framework is chosen, utilizing a Bayesian optimization technique called a tree-structured parzen estimator; i.e., TPE (Bergstra et al., 2011; Watanabe, 2023) to determine the optimum set of hyperparameters. Additionally, using detailed benchmark studies comparing various optimization techniques and open-source frameworks, Optuna-TPE provided the most favorable performance and computation times (Motz et al., 2022; Shekhar et al., 2022). Typically, the hyperparameter optimization process is carried out with *k*-folds cross validation:

A) Divide training data into *k* sets (i.e., "folds") randomly. Typically, $k = 3$, 4, or 5.
B) Sample a combination of hyperparameters (e.g., the number of decision trees. For more information regarding hyperparameters, please refer to the Appendix, Section 7.4.)
C) Set $i = 1$.
D) Place set *i* aside, and train the model using the remaining $k - 1$ sets. (e.g., when $k = 3$, these are sets 2 and 3.)
E) Evaluate the performance of the trained model using set *i* as the *validation* set and compute the *validation score* (e.g., RMSE – root-mean-squared-error; in Section 3.3, see Eq. (28)).
F) When $i < k$, set $i = i + 1$. Return to D).
G) When $i = k$, compute the average validation score.
H) Return to B). (e.g., sample a different value for number of decision trees)
I) Return the combination of hyperparameters that resulted in the *maximum*/*minimum* average validation score, depending on the chosen metric (e.g., return the number of decision trees that resulted in the *minimum* average RMSE).

ii) Model training with the Optimum Hyperparameters (Figure 4, step V): Post optimization i), the ML model, with its optimum hyperparameters, is trained using the entire training data.

Table 3. Example Dataset for the Decision Tree Trained in Figure 5

Temperature, T (K)	**Committer Probability, p_B**
300	0.1
340	0.2
380	0.3
420	0.4
460	0.5
500	0.6
540	0.7
580	0.8
620	0.9
660	1.0

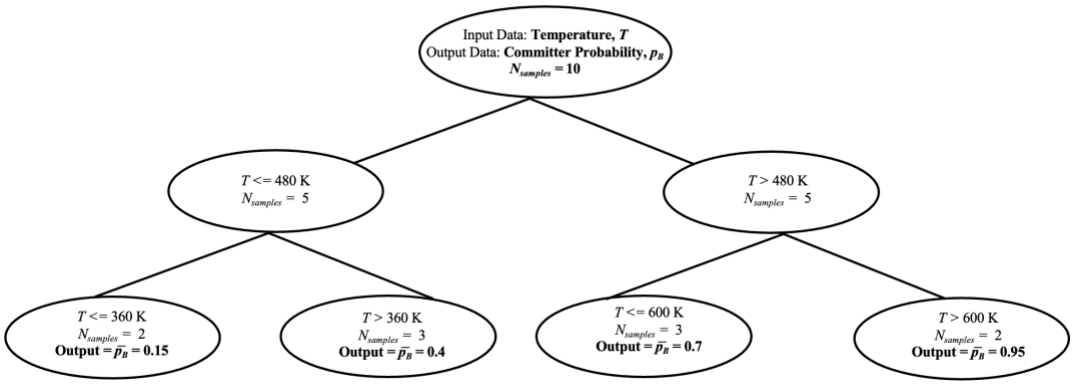


Figure 5. Schematic of a decision tree, a flowchart-like model that helps make decisions by answering a series of questions based on the *input variables* (e.g., temperature) of the data, ultimately leading to a decision or prediction.

The trained ML model is then evaluated using the testing data, returning the evaluation score (Figure 4, step VI). When the performance is satisfactory, the model is ready for deployment; i.e., fresh predictions are generated using the trained model as new input data are received (Figure 4, step VII). Eventually, as output data become available, they are compared with the model predictions, following which, the output data and their corresponding input data are sent back to the data collection step as *feedback* for *model refinement* (Figure 4, step VIII). Clearly, in practice, the ML pipeline is an infinite cycle, with continuous monitoring and model updates as data are received.

Herein, as dynamic simulations yield process variable values, real-time predictions for p_B are estimated by deploying the trained XGBoost model. Clearly, these predictions are utilized to sound *dynamic* multivariate alarms when $p_B >= p_{B,target}$ for defined $p_{B,target}$ values. Consequently, automated response actions are activated that attempt to decrease p_B in real-time. These yield *initial* multivariate alarm systems, consisting of initial guesses for the alarm thresholds and response actions.

2.6. Step 3: Alarm Rationalization and Dynamic Risk Analyses (DRAn)

For the multivariate alarm systems developed in Step 2, key questions are: *Does every alarm indicate an impending rare abnormal event? Are there too many or too few alarms? Is every response action effective in returning the process to normal operation?* To address these issues, rationalization strategies, with improved alarm thresholds and response actions, were demonstrated for an exothermic CSTR, resulting in a reduction of nuisance alarms, by Sudarshan et al. (2024). Note that the key aim of alarm rationalization is to reduce significantly both the number of false alarms and the total alarms annunciated, to ensure that every alarm is a quality alarm. It is claimed that through rationalization, the total number of alarms can be reduced by 50%, coupled with reductions in the nuisance alarms (Timms, 2009).

Next, post rationalization, the risk associated with the multivariate alarm systems is evaluated using dynamic risk analysis (DRAn) — a technique developed over the past two decades to help achieve real-time proactive risk management against rare abnormal events. Typically, assuming availability of raw alarm data from process historians, DRAn involves: i) Tracking all possible postulated abnormal events listed as part of HAZOP studies using raw alarm data; ii) Creating event trees showing all possible paths taken by an abnormal event when propagating through the alarm systems; iii) Compacting the data into a concise and structured representation (Pariyani et al., 2012a); iv) Using the compacted data, to perform Bayesian computations to estimate failure probabilities of alarm systems, probability of trips, and probability of accidents (Pariyani et al., 2012b). However, for our case (with random statistical noise, η, used to model un-postulated abnormal events – see, e.g., Eq. (20)), given the unavailability of raw alarm data, the DRAn methodology is modified:

i) Conducting multiple batches of dynamic simulations of the process, inclusive of control and the rationalized multivariate alarms systems. For each simulation, recording key alarm statistics (e.g., number of alarms, number of alarms at the current level that were active when the next alarm level is reached, and the like). Then, computing alarm system failure probabilities (e.g., 100 batches of dynamic simulations yield 100 failure probabilities for each alarm system).
ii) Using these failure probability results, constructing an informed prior distribution (IPD)
iii) When no alarm system failure data are available, constructing the likelihood distribution using assumed alarm failure data.
iv) Using Bayes' Rule, constructing the final posterior distribution for the failure probabilities as:

$$f\left(p_{\text{failure},i} \mid D\right) = \frac{f\left(p_{\text{failure},i}\right) \times f\left(D \middle| p_{\text{failure},i}\right)}{\int_0^1 f\left(p_{\text{failure},i}\right) \times f(D|x_i)\, dp_{\text{failure},i}} \tag{18}$$

where $p_{\text{failure},i}$ represents the failure probabilities for alarm system i; $f\left(p_{\text{failure},i}\right)$ is the IPD; $f\left(D \middle| p_{\text{failure},i}\right)$ is the likelihood distribution constructed from likelihood data, D; $f\left(p_{\text{failure},i} \middle| D\right)$ is the posterior distribution. Hence, the low-variance IPD constructed based on multiple dynamic simulations results in a low-variance and reliable posterior distribution as compared to a non-informative flat prior that typically leads to an unreliable posterior, depending entirely on the high-variance likelihood distribution (Sudarshan et al., 2024).

2.7. Step 4: Sensitivity Analyses

Note that response actions are key components of multivariate alarm systems – to ensure that the alarm systems are effective, it is expected that the process parameter selected as the response-action variable in Section 2.4 has a reasonably strong influence on the committer probabilities. Hence, in sensitivity analyses, the above steps, to estimate failure probabilities, are repeated for other process parameters that potentially could have stronger impacts on the committer probabilities (e.g., q_m, q_s, etc.). Then, the results and statistics computed using DRAn are compared to identify the most effective response-action variable.

2.8. Computational Specifications and Software Utilized

Note that all simulations, analyses, and results presented herein were conducted on a Windows 11 Desktop computer, having specifications:

i) CPU: 12^{th} - generation Intel i7-12700K with 12 cores (8 performance + 4 efficiency), 32 GB DDR5 RAM
ii) GPU: NVIDIA RTX 3060 Ti, 8 GB RAM

For bifurcation analyses conducted in step 0, the MATCONT toolbox, based on MATLAB, is utilized (Dhooge et al., 2003). For all other steps, the Python programming language (versions 3.9 and 3.11) is utilized, leveraging several powerful open-source software packages, including: NumPy (Harris et al., 2020), Pandas (McKinney, 2010) , SciPy (Virtanen et al., 2020), Matplotlib (Hunter, 2007), Scikit-Learn (Pedregosa et al., 2018), XGBoost (Chen and Guestrin, 2016), Numba (Lam et al., 2015), Optuna (Akiba et al., 2019), to name a few. Also, for efficient GPU-parallelization during model development, the NVIDIA Compute Unified Device Architecture (CUDA) toolkit (NVIDIA et al., 2022) is utilized.

3. RESULTS AND DISCUSSIONS

For each step in Section 2, this section presents results with discussions.

3.1. Step 0: Preliminary Analyses

Figure 6 shows the steady-state solution diagram for the polystyrene CSTR, with q_c being the free bifurcation parameter. The stable branches are shown in solid, and unstable branches in dotted lines. The intermediate unstable region is selected as the desired region, given that, to ensure safe and reliable operation, this is the preferred operating region for most polymerization reactors. Additionally, two stable, undesirable regions are observed; i.e., the unsafe region, with high conversion and high temperature, and the unreliable region, with low conversion and low temperature. Clearly, two rare, abnormal shifts from the unstable operating region are possible: i) Unstable to Unsafe; ii) Unstable to Unreliable. Note the key bifurcation points indicated using red asterisks; i.e., limit points, LP (i.e., the Jacobian matrix has at least one zero eigenvalue), and Hopf bifurcation point, H (i.e., the Jacobian matrix has a pair of complex-conjugate eigenvalues with zero real part), associated with changes in the stability of the process (Kubíček and Marek, 1983).

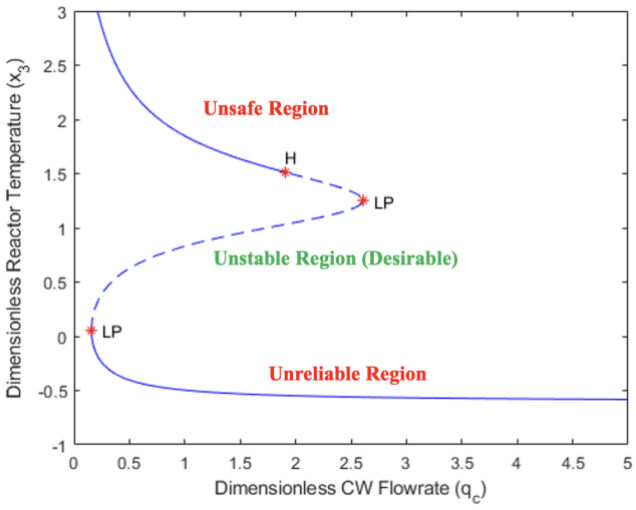


Figure 6. Steady-state solution diagram for the polystyrene CSTR

Next, a PID-control scheme is developed that controls x_3 by manipulating q_c, with constraints imposed on q_c. The controller parameters (i.e., controller gain, K_c, integral time constant, τ_I, and derivative time constant, τ_D) are selected experimentally to ensure both sufficiently tight control as well as to allow rare abnormal transitions to the undesirable regions. Additionally, to model un-postulated abnormal events, random, statistical *noise* is introduced into the monomer feed concentration. With control and noise, the dimensionless modeling equations for the polystyrene CSTR are:

$$\frac{dx_1}{d\tau} = q_i x_{1f} - (q_i + q_m + q_s)x_1 - \phi_d \kappa_d(x_3)x_1 \tag{19}$$

$$\frac{dx_2}{d\tau} = q_m(x_{2f} + \boldsymbol{\eta}) - (q_i + q_m + q_s)x_2 - \phi_p \kappa_p(x_3)x_2 x_5;\ \eta \sim \mathcal{N}\left(0,\ {\sigma_\eta}^2\right) \tag{20}$$

$$\frac{dx_3}{d\tau} = (q_i + q_m + q_s)(x_{3f} - x_3) + \beta\phi_p \kappa_p(x_3)x_2 x_5 - \delta(x_3 - x_4) \tag{21}$$

$$\frac{dx_4}{d\tau} = \delta_1[q_c(x_{4f} - x_4) + \delta\delta_2(x_3 - x_4)] \tag{22}$$

$$q_c = q_{c,0} - K_c\left[(x_{3,sp} - x_3) + \frac{1}{\tau_I}\int_0^t (x_{3,sp} - x_3)dt' + \tau_D \frac{d(x_{3,sp} - x_3)}{dt}\right] \tag{23}$$

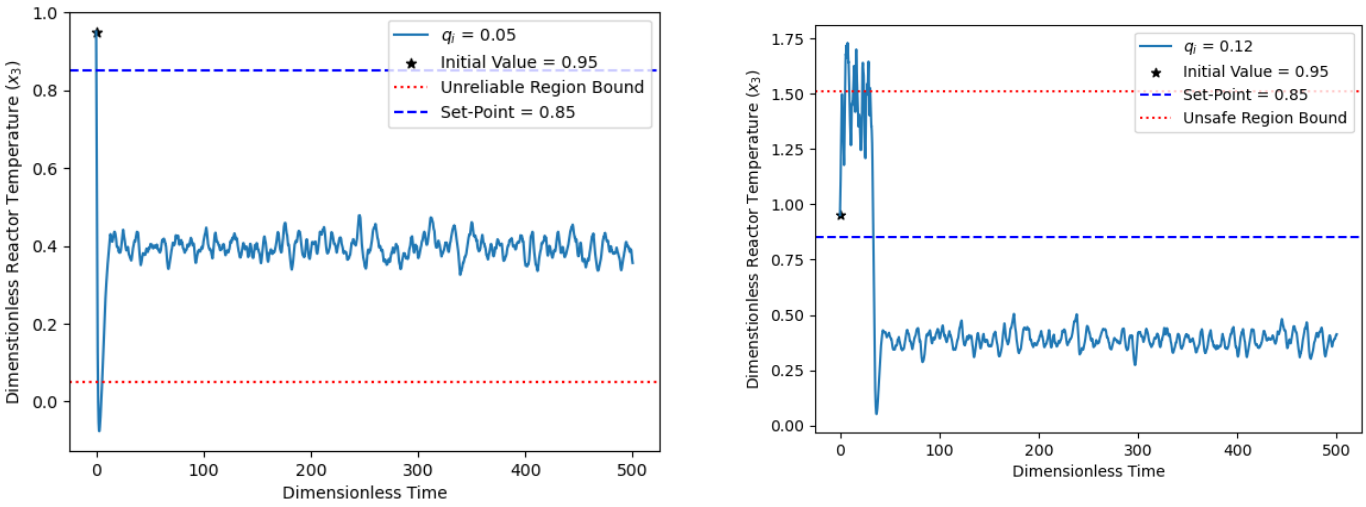
$$0 \le q_c \le 5 \tag{24}$$

$$x_5 = \sqrt{\frac{2f\phi_d \kappa_d(x_3)x_1}{\phi_t \kappa_t(x_3)}} \tag{25}$$

$$x_{1,0} = 0.0041;\ x_{2,0} = 0.2156;\ x_{3,0} = 0.951;\ x_{4,0} = -1.1191;\ q_{c,0} = 1.5 \tag{26}$$

where $x_{1,0}$, $x_{2,0}$, $x_{3,0}$, $x_{4,0}$, $q_{c,0}$ represent initial values. Note that the statistical noise; i.e., η, is sampled at every integration time-step (with integration step-size; i.e., h = 0.001), from a normal distribution,

$\mathcal{N}(\mu, {\sigma_\eta}^2)$, with a mean; i.e., $\mu = 0$, and variance; i.e., ${\sigma_\eta}^2 = 0.0014$. Note that for accurate analyses, identical initial conditions need to be considered for abnormal trajectories generated towards both undesirable regions. Hence, the initial values in Eq. (26), as well the variance of statistical noise, i.e., ${\sigma_\eta}^2$, are selected experimentally, based on trial-and-error simulations, to ensure multiple rare transitions are simulated from the unstable region. Figure 7 shows the PID-controlled polystyrene CSTR (with values for the dimensionless controller tuning parameters in Table 1) under dynamic operation with *noisy* monomer feed concentrations, showing abnormal shifts to: a) Unreliable region and b) Unsafe region. At low initiator flowrates, $q_i = 0.05$, the reactor exhibits strong inverse response, causing rapid shifts in operation from the unstable region to the unreliable operating region; however, the reactor does not remain in the unreliable region and quickly returns to the unstable region. At high initiator flowrates, $q_i = 0.12$, the reactor shifts to the unsafe region, remains for some time, and returns to the unstable region. Additionally, note the significant offset away from the desired set-point (i.e., $x_{3,\,SP} = 0.85$), with potential reasons being: increased sensitivity of the PID controller to statistical noise, integral windup due to the input constraints, causing the controller to undershoot, and the like.



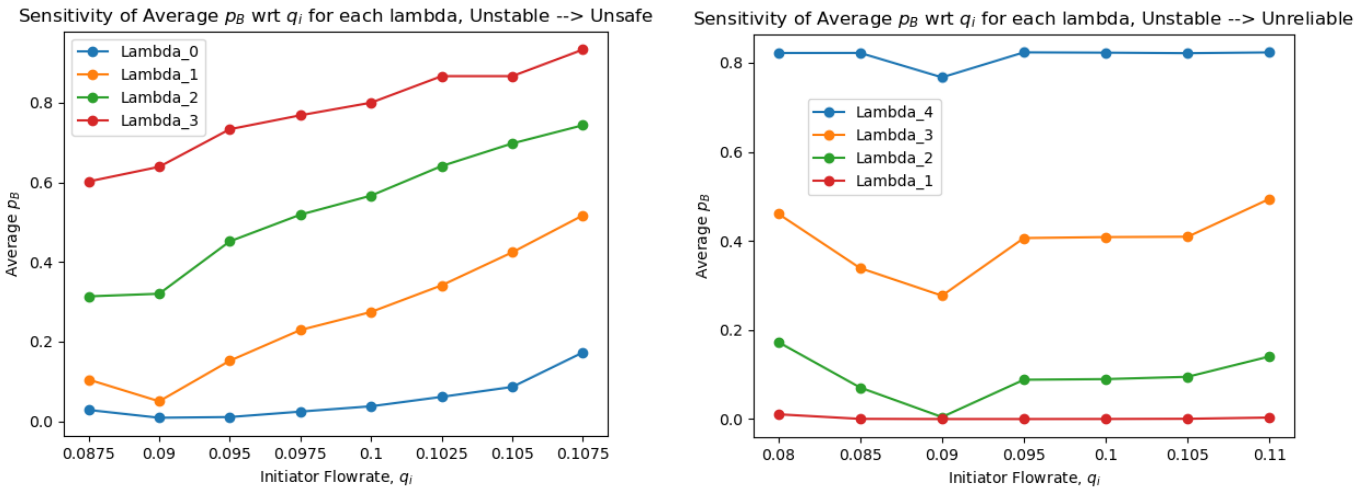
(a) (b)

Figure 7. The polystyrene CSTR under dynamic operation with *noisy* monomer feed concentrations, showing abnormal shifts to: a) Unreliable region at low q_i and b) Unsafe region at high q_i

3.2. Step 1: Data Generation via Forward-flux Sampling

Using BG-FFS, numerous rare abnormal trajectories are simulated efficiently in a piecewise manner – given two undesirable regions, the simulations are conducted *independently* for unsafe and unreliable shifts. Note that the dimensionless reactor temperature; i.e., x_3, is chosen as the order parameter variable. Initially, the initiator flowrate, q_i, is chosen as the response-action variable; hence, for each undesirable region, the BG-FFS simulations are conducted for multiple discrete values of q_i. As part of exploratory data analysis, Figure 8 shows the average p_B as a function of q_i for various λ_i, for the two abnormal shifts. For shifts to

the unsafe region, the average p_B appears to increase with q_i; for shifts to the unreliable region, the trend is not very clear, with decreases in the average p_B observed for intermediate q_i.



(a) (b)

Figure 8. Average p_B as function of q_i for various λ_i, for abnormal shifts to: a) Unsafe region and b) Unreliable region

3.3. Step 2: Predictive Modeling via Machine Learning

Given two categories of abnormal shifts; i.e., unstable to unsafe and unstable to unreliable, two XGBoost models are developed independently by utilizing the steps mentioned in Section 2.5, for the data generated during BG-FFS simulations and p_B calculations. Then, the trained XGBoost models are evaluated on the test data using the following evaluation scores:

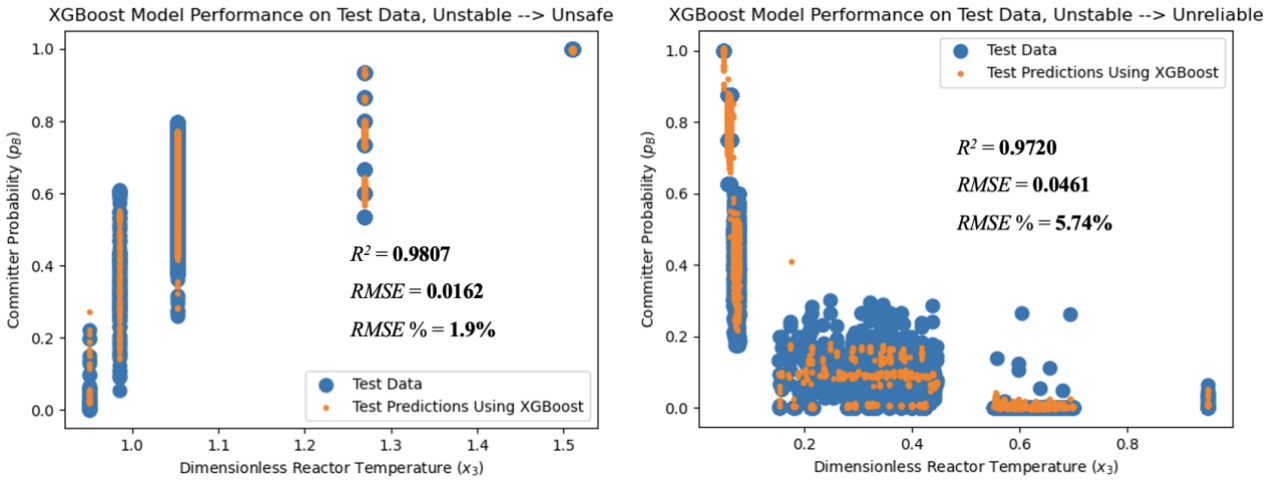
$$R^2 = \text{Coefficient of Determination} = 1 - \frac{\sum_{i=1}^{N_{\text{samples,test}}} \left(p_{\text{B,test}}(i) - \hat{p}_{\text{B,test}}(i)\right)^2}{\sum_{i=1}^{N_{\text{samples,test}}} \left(p_{\text{B,test}}(i) - \bar{p}_{\text{B,test}}\right)^2} \quad (27)$$

$$RMSE = \text{Root Mean Squared Error} = \sqrt{\frac{\sum_{i=1}^{N_{\text{samples,test}}} \left(p_{\text{B,test}}(i) - \hat{p}_{\text{B, test}}(i)\right)^2}{N_{\text{samples,test}}}} \quad (28)$$

$$RMSE\ \% = \frac{RMSE}{\bar{p}_{\text{B,test}}} \times 100 \quad (29)$$

where $p_{\text{B,test}}$ represents the p_B in the test data, $\hat{p}_{\text{B,test}}$ is the p_B predicted using the trained XGBoost model for the test data, $\bar{p}_{\text{B,test}}$ represents the mean p_B for the test data, and $N_{\text{samples,test}}$ represents the number of samples in the test data. Figure 9 shows the performance and evaluation scores for the two XGBoost

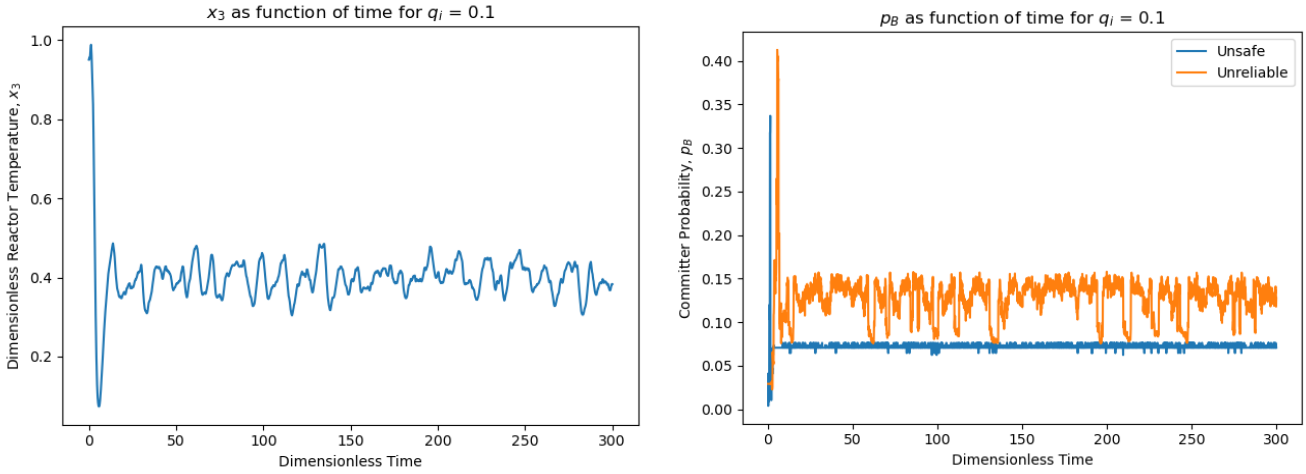
models. For both models, high R^2 and reasonably low *RMSE* and *RMSE%* are observed, indicating strong predictive performance. Note that the clock time recorded for development of each XGBoost model is ~ 7-8 min, with GPU acceleration implemented during hyperparameter optimization. Additionally, note that for reliable cross-validation during the hyperparameter optimization process, it is important to obtain lower standard deviations across folds, for the given evaluation metric selected for cross-validation (i.e., *RMSE*). For both XGBoost models, during cross-validated hyperparameter optimization, the ratio of standard deviation to mean for the *RMSE* is ~ 10^{-2}, confirming the reliability of the cross-validation process.



(a) (b)

Figure 9. Performance of the XGBoost model on test data and evaluation metrics for: a) Unstable to Unsafe; b) Unstable to Unreliable

Given satisfactory performance on test data, both models are adequate to return real-time predictions of p_B, given dynamic simulations not used to obtain the training or testing data. Figure 10a shows x_3 as function of dimensionless time for a new dynamic simulation, with Figure 10b showing real-time p_B predictions by using both XGBoost models. Clearly, these predictions can be leveraged to develop *dynamic*, multivariate alarm systems; e.g., if the first alarm threshold for the unsafe region is set at 0.20, then, when real-time $p_{B,Unsafe}$ >= 0.20, the alarm goes off and its corresponding response action (e.g., slight decrease in q_i) is activated instantaneously, attempting to decrease $p_{B,Unsafe}$ in real-time.



(a) (b)

Figure 10. a) A new dynamic simulation showing x_3 as function of time; b) Real-time p_B predictions obtained by deploying the XGBoost models for the dynamic simulation

3.4. Step 3: Alarm Rationalization and Dynamic Risk Analyses (DRAn)

With real-time tracking of p_B available from the XGBoost models, six-level initial multivariate alarm systems, consisting of alarm thresholds (defined using p_B limits, where basin B is an unsafe or unreliable region) and response actions (defined using discrete q_i values) are:

1. **High Alarm Systems:** Alarm thresholds and response actions corresponding to the unsafe operating region. As initial guesses, these are:
   i) H (High): $p_{B,Unsafe,H}$= 0.20; $q_{i,H}$ = 0.0975
   ii) HH (High-high): $p_{B,Unsafe,HH}$= 0.40; $q_{i,HH}$ = 0.095
   iii) HHH (High-high-high): $p_{B,Unsafe,HHH}$= 0.80; $q_{i,HHH}$ = 0.09

2. **Low Alarm Systems:** Alarm thresholds and response actions corresponding to the unreliable operating region. As initial guesses, these are:
   i) L (Low): $p_{B,Unreliable,L}$= 0.20; $q_{i,L}$ = 0.0975
   ii) LL (Low-low): $p_{B,Unreliable,LL}$= 0.40; $q_{i,LL}$ = 0.095
   iii) LLL (Low-low-low): $p_{B,Unreliable,LLL}$= 0.80; $q_{i,LLL}$ = 0.09

Note the decreasing trend for q_i-based response actions selected for both high and low alarm systems, as the undesirable region is approached – from Figure 8, while the trend for $p_{B,Unreliable}$ is unclear, an increasing trend is observed for $p_{B,Unsafe}$ (i.e., as q_i increases, $p_{B,Unsafe}$ increases); hence, a decreasing trend of response actions is chosen for both high- and low-alarm systems to ensure that safety is not compromised. Additionally, note that response actions corresponding to each alarm level are assumed to

be completely automated, comparable to an "ideal" operator having negligible response times. Figure 11 represents a schematic of multivariate alarm systems in action. For instance, consider a process trajectory (shown in black) initially operating in the normal operating region; as soon as it crosses the H-threshold limit defined for the primary alarm variable, the H-alarm goes off, and its corresponding response action (i.e., changes made to the response action variable, given the XGBoost models' p_B predictions) is activated, attempting to return the process to normal operation.

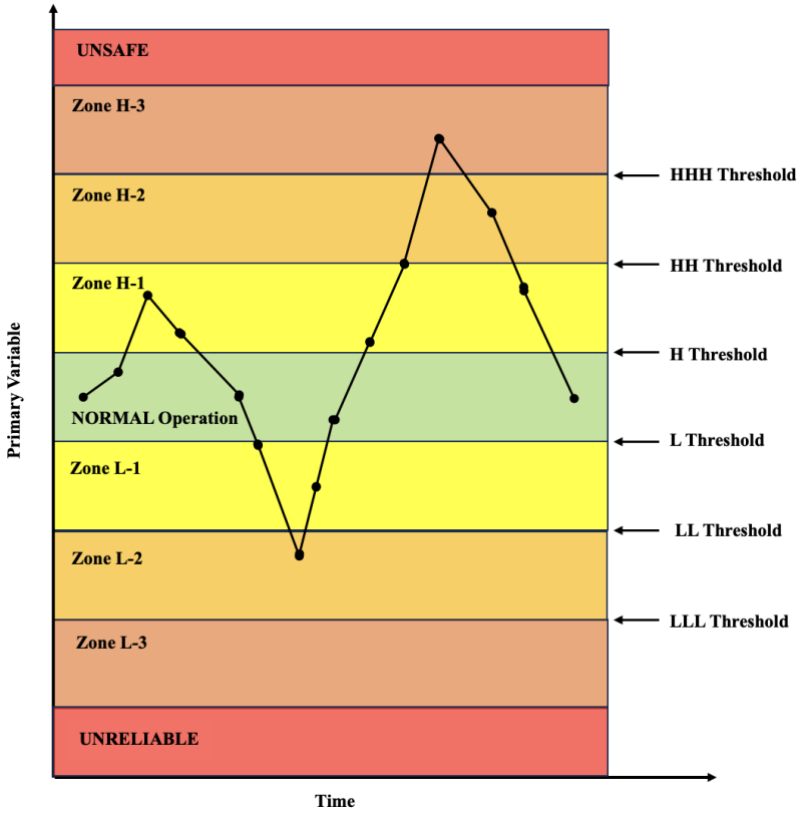


Figure 11. Schematic showing multivariate alarm systems in action, given measurements available for a primary alarm variable.

Note that the initial initiator flowrate; i.e., $q_{i,init}$ = 0.1. Next, as described in Section 2.6, the algorithm developed for rationalization strategies is utilized to evaluate the initial multivariate alarm systems – specifically, the alarm statistics for the initial six-level alarm system (see Table 4) is compared to that of four-level alarm system (defined in Table 5), using 100 dynamic simulations. Figure 12 compares the alarm statistics for the two systems w.r.t high alarms. Overall, the number of alarms that go off for the four-level system is higher than for the six-level system. The number of instances of reaching the unsafe region is lower for the six-level system (i.e., 0 for six-level, but 2 for four-level). Additionally, the percentage of unique high alarms that remain active when the next threshold is reached is observed to be much lower for the six-level system, implying greater effectiveness and success in reducing the real-time committer probabilities to below the limit. Similarly, Figure 13 compares the alarm statistics w.r.t low alarms. Despite more alarms in the four-level system, the number of instances of reaching the unreliable region is observed to be identical for both systems (i.e., 2). Additionally, the percentage of unique low

alarms remaining active when the next threshold is reached is lower for the four-level system. It is likely that when $p_{B,Unreliable}$ reaches intermediate values (i.e., ~ 0.4-0.6), the process is not very sensitive to q_i changes , thereby, resulting in 100% failure of the LL alarm in the six-level system, as observed in Figure 13b. A potential reason for this is the inverse response behavior observed previously in Figure 7a – a phenomenon very challenging to mitigate.

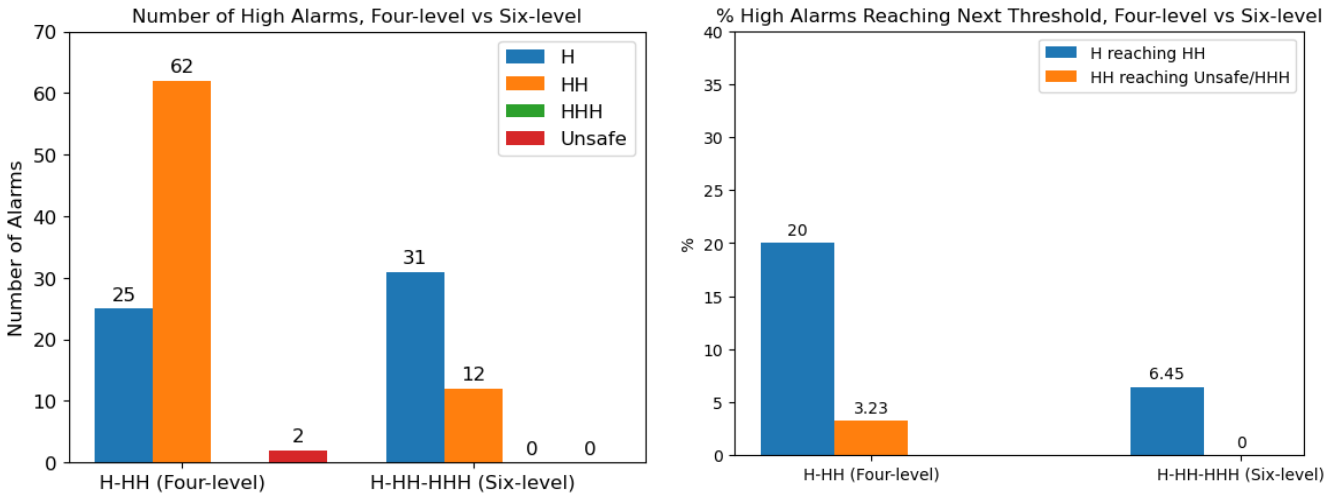
Hence, based on these rationalization studies, a five-level alarm system; i.e., H-HH-HHH-L-LL (summarized in Table 6), is preferable, to ensure effectiveness against shifts to both unsafe and unreliable regions. To validate this, Figure 14 a-d shows the alarm statistics for the five-level system, based on 100 dynamic simulations. The number and percentage of high alarms reaching the next threshold are comparable to the six-level system, with zero instances of reaching the unsafe region. For the low alarms, the number of alarms is lower compared to the four-level system, with comparable percentages of unique low alarms reaching the next threshold.

Table 4. Six-level Multivariate Alarm System, to be Evaluated during Rationalization

Alarm Threshold	**p_B at Threshold (initial)**	**Response Action at Threshold (initial)**
H	$p_{B,Unsafe,H} = 0.2$	$q_{i,H} = 0.0975$
HH	$p_{B,Unsafe,HH} = 0.4$	$q_{i,HH} = 0.095$
HHH	$p_{B,Unsafe,HHH} = 0.8$	$q_{i,HHH} = 0.09$
L	$p_{B,Unreliable,L} = 0.2$	$q_{i,L} = 0.0975$
LL	$p_{B,Unreliable,LL} = 0.4$	$q_{i,LL} = 0.095$
LLL	$p_{B,Unreliable,LLL} = 0.8$	$q_{i,LLL} = 0.09$

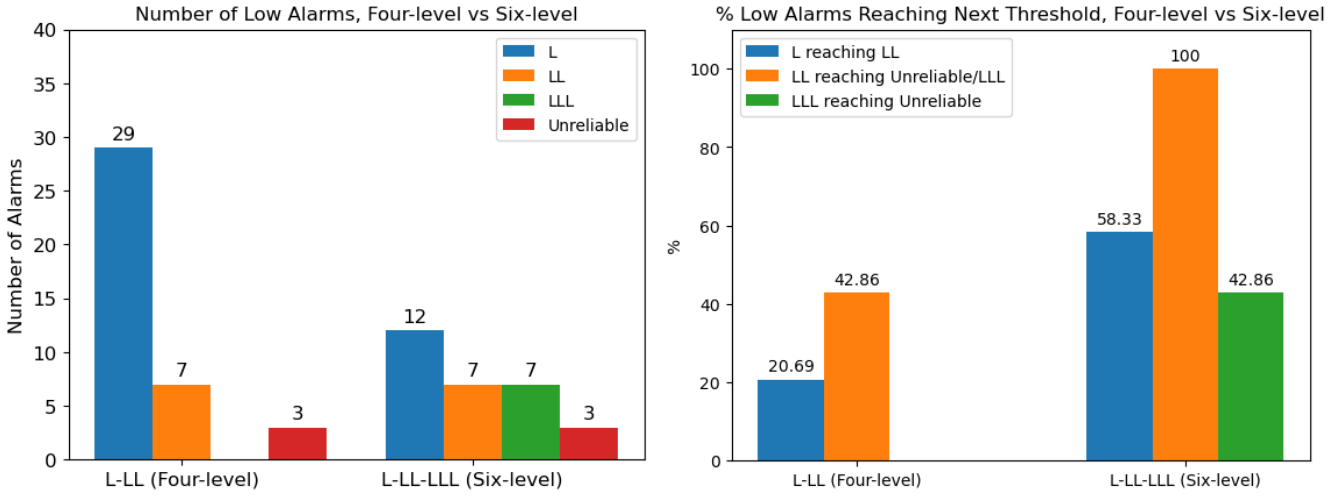
Table 5. Four-level Multivariate Alarm System, to be Evaluated during Rationalization

Alarm Threshold	**p_B at Threshold (initial)**	**Response Action at Threshold (initial)**
H	$p_{B,Unsafe,H} = 0.4$	$q_{i,H} = 0.095$
HH	$p_{B,Unsafe,HH} = 0.8$	$q_{i,HH} = 0.09$
L	$p_{B,Unreliable,L} = 0.35$	$q_{i,L} = 0.095$
LL	$p_{B,Unreliable,LL} = 0.8$	$q_{i,LL} = 0.09$



(a) (b)

Figure 12. Alarm statistics comparison between four-level and six-level alarm systems w.r.t high-alarm systems, showing: a) Number of high alarms; b) Percentage of unique high alarms that are active when next level is reached.



(a) (b)

Figure 13. Alarm statistics comparison between four-level and six-level alarm systems w.r.t low-alarm systems, showing: a) Number of low alarms; b) Percentage of unique low alarms that are active when next level is reached.

Table 6. Rationalized Five-level Alarm System, i.e., H-HH-HH-L-LL

Alarm Threshold	p_B at Threshold (final)	Response Action at Threshold (final)
H	$p_{B,Unsafe,H} = 0.2$	$q_{i,H} = 0.0975$
HH	$p_{B,Unsafe,HH} = 0.4$	$q_{i,HH} = 0.095$
HHH	$p_{B,Unsafe,HHH} = 0.8$	$q_{i,HHH} = 0.09$
L	$p_{B,Unreliable,L} = 0.35$	$q_{i,L} = 0.095$
LL	$p_{B,Unreliable,LL} = 0.8$	$q_{i,LL} = 0.09$

Note that while rationalization studies are important to evaluate the placement of alarm thresholds and choice of response actions, the alarm statistics are generated from just one batch of ~ 100 dynamic simulations; for more comprehensive analyses regarding the failure probabilities of the multivariate alarm systems, it is important to generate alarm statistics based on multiple such batches of dynamic simulations. Hence, for risk assessment, as described in Section 2.6., DRAn studies are conducted based on multiple dynamic simulations of the polystyrene CSTR, inclusive of control and the rationalized five-level multivariate alarm systems. ~75 dynamic simulation batches are conducted, with each batch consisting of 200 dynamic simulations. For each of the five alarm systems, the failure probability is computed as:

$$p_{\text{failure, H}} = \frac{N_{\text{H}\to\text{HH}}}{N_{\text{H}}} \tag{30}$$

$$p_{\text{failure, HH}} = \frac{N_{\text{HH}\to\text{HHH}}}{N_{\text{HH}}} \tag{31}$$

$$p_{\text{failure, HHH}} = \frac{N_{\text{HHH}\to\text{Unsafe}}}{N_{\text{HHH}}} \tag{32}$$

$$p_{\text{failure, L}} = \frac{N_{\text{L}\to\text{LL}}}{N_{\text{L}}} \tag{33}$$

$$p_{\text{failure, LL}} = \frac{N_{\text{LL}\to\text{Unreliable}}}{N_{\text{LL}}} \tag{34}$$

where N_{H}, N_{HH}, N_{HHH}, N_{L}, N_{LL} are the number of alarms for the H, HH, HHH, L and LL systems; $N_{\text{H}\to\text{HH}}$ is the number of unique H alarms that are active when HH is reached; $N_{\text{HH}\to\text{HHH}}$ is the number of unique HH alarms that are active when HHH is reached; $N_{\text{HHH}\to\text{Unsafe}}$ is the number of unique HHH alarms that are active when the unsafe region is reached; $N_{\text{L}\to\text{LL}}$ is the number of unique L alarms that are active when LL is reached; $N_{\text{LL}\to\text{Unreliable}}$ is the number of unique LL alarms that are active when the unreliable region is reached.

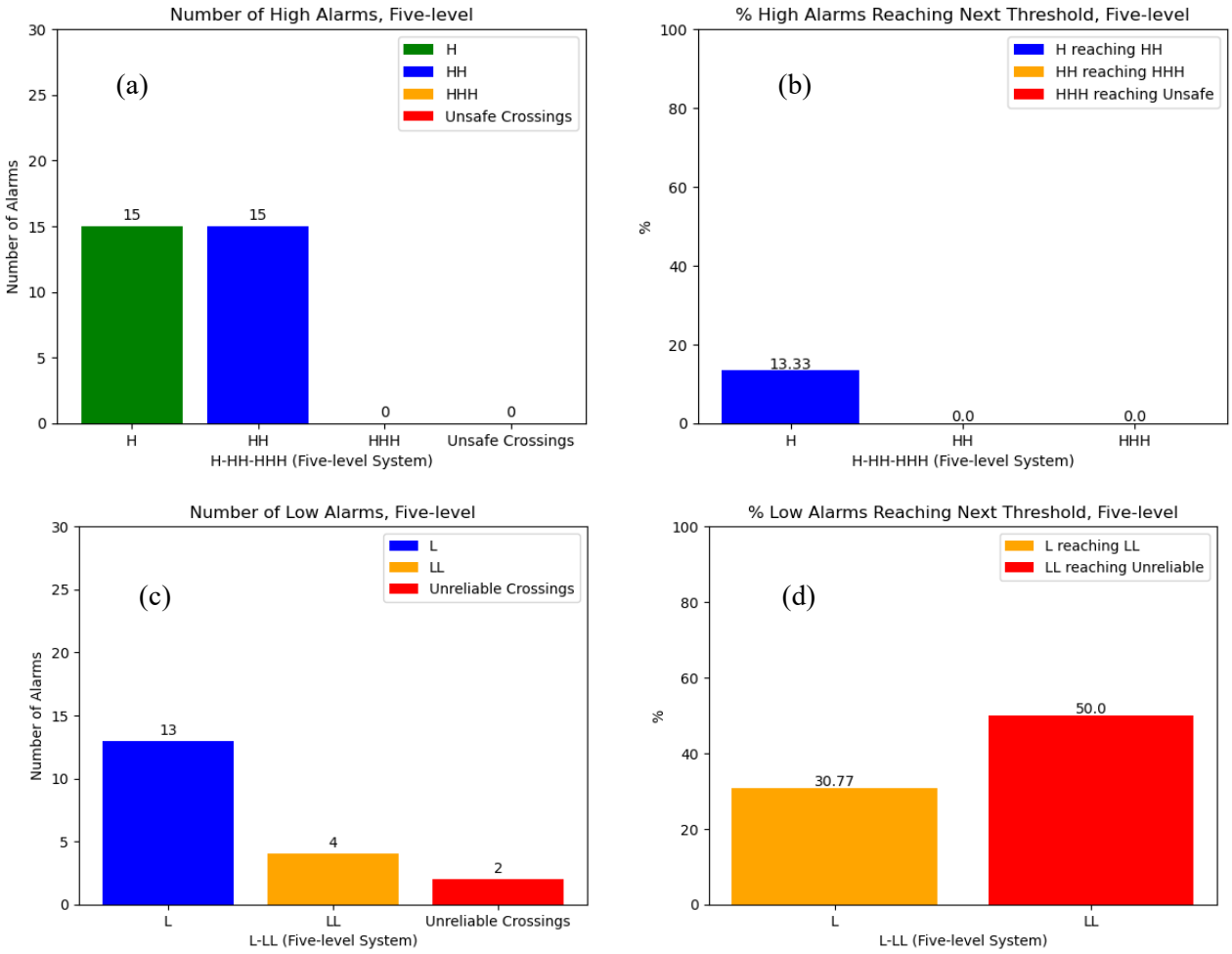


Figure 14. Alarm statistics for the proposed five-level alarm system, showing: a) Number of high alarms; b) Percentage of unique high alarms remaining active when next threshold is reached; c) Number of low alarms; d) Percentage of unique low alarms remaining active when next threshold is reached.

Figures 15 a-e represent the results generated from the DRAn studies, consisting of the histogram for the estimated failure-probabilities data, the Beta IPD developed for the data, and posterior distribution constructed using Bayes' rule, using the IPD and the likelihood distribution. Given the unavailability of failure data, for each of the five alarm systems, a binomial likelihood was assumed (Sudarshan et al., 2024). For more details regarding the computed parameters for the Beta IPDs and assumptions for likelihood distributions, please refer to the Appendix, Section 7.3. Given that the Beta distribution is a conjugate prior of the binomial distribution, the resulting posterior also is a Beta distribution (Gelman et al., 2013). The high-alarm systems appear to be reasonably effective, with a decreasing trend observed for the average failure probabilities as the unsafe region is approached. For the low-alarm systems, the average failure probabilities appear to follow an increasing trend as the unreliable region is approached, with possible improvements to be investigated in the sensitivity analyses of Section 3.5.

3.5. Sensitivity Analyses

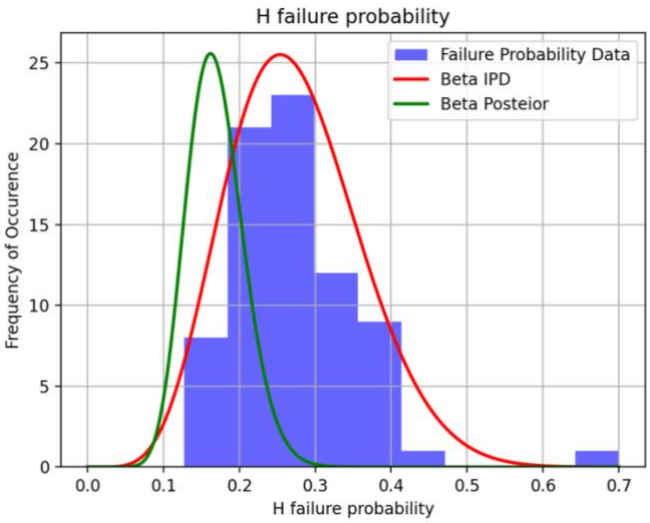
Thus far, the initiator flowrate, q_i, has been selected as the response-action variable for the multivariate alarm systems. In this Sensitivity Analyses subsection, other potential response-action variables are considered; particularly, the monomer flowrate; i.e., q_m, and the solvent flowrate; i.e., q_s. Initially, BG-FFS simulations (i.e., Section 2.4.) are conducted for multiple discrete values of each response-action variable. With several crossing points and committer probabilities generated for each response-action variable, average committer probabilities are computed. Then, Pearson correlation coefficients are computed between the average committer probabilities and response-action variables. These measure the strength and direction of the linear association between the two averages (Kirch, 2008), and are computed as:

$$Pearson_{u,v} = \frac{\sum_{i=1}^{n_{\text{discrete}}}(u_i - \bar{u})(v_i - v)}{\sqrt{\sum_{i=1}^{n_{\text{discrete}}}(u_i - u)^2 \sum_{i=1}^{n_{\text{discrete}}}(v_i - \bar{v})^2}} \tag{35}$$

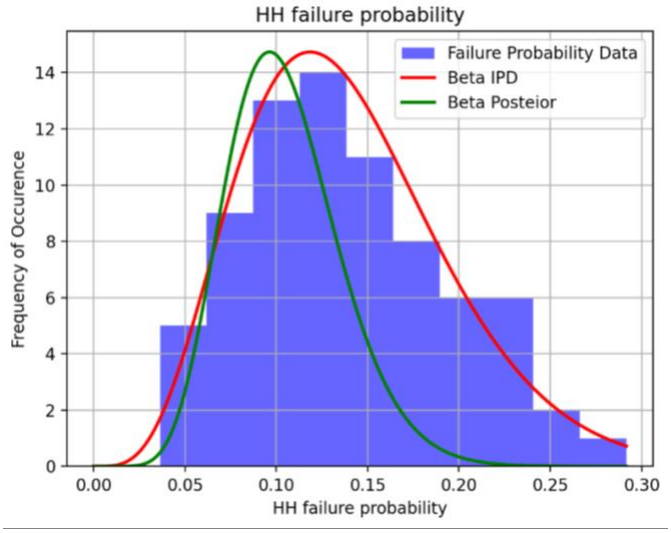
where $Pearson_{u,v}$ is the Pearson correlation coefficient between variables, u and v. In this case, u is the response-action variable (i.e., q_i, q_m, or q_s); v is the average committer probability, $\bar{p}_B$; $\bar{u}$ and $\bar{v}$ are the mean values for u and v, with $\bar{v}$ being the twice-averaged committer probability, and n_{discrete} is the number of discrete values considered for the response-action variable.

Table 7 shows the Pearson coefficients between average committer probabilities and the response-action variables for both unsafe and unreliable regions. For the unsafe operating region, all three response-action variables show strong correlation, with q_s showing strong negative correlation, while the others show strong positive correlations. For the unreliable region, all three response-action variables show positive correlations, with q_i showing the weakest and q_s showing the strongest correlation. Despite q_s showing relatively stronger positive correlation for the unreliable region, it is negatively correlated w.r.t to the unsafe region – this implies that a decrease in q_s will result in decreases in $p_{B,Unreliable}$, but at the cost of increased $p_{B,Unsafe}$. Given that safety should not be compromised, further sensitivity analyses are restricted to only q_i and q_m; q_s is not considered in this study. Also, note that sensitivity analyses conducted, herein, do not consider the effort required to vary the response-action variables.

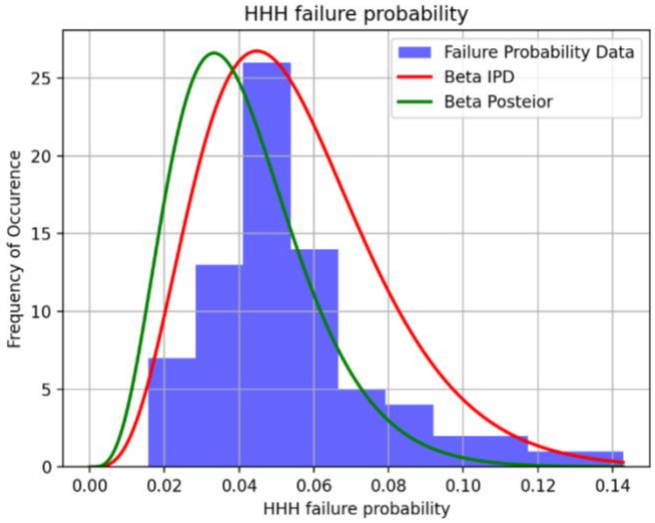
Next, steps 1 - 3 described in Section 2 are repeated to develop dynamic multivariate alarm systems with q_m as the response-action variable (similar to those developed for q_i in Sections 3.2-3.4) – note that based on the rationalization algorithm, a four-level multivariate alarm system (i.e., H-HH-L-LL, with q_m response actions) is preferred. Then, DRAn studies are conducted using the q_m-based alarm systems, with their failure probability statistics compared with q_i-based alarm systems developed in Section 3. Table 8 shows the average alarm statistics per batch of dynamic simulations (i.e., each batch consists of 200 dynamic simulations), generated using DRAn studies for both q_i- and q_m-based alarm systems. On average, high-alarm failure probabilities are higher for q_m; for the L alarm system, q_m has lower failure probabilities, with LL-failure probabilities being slightly lower for q_i. However, the average number of unsafe crossings observed for q_m are *significantly* lower than q_i, i.e., by ~ 77%; additionally, the average number of unreliable crossings are observed to be slightly lower for q_m, i.e., by ~ 8%, with comparable average growing polymer concentrations (i.e., a measure of product quality). Hence, despite relatively higher failure probabilities, the q_m-based multivariate alarm systems are more effective than the q_i-based systems, resulting in fewer instances of reaching the undesirable operating regions, with negligible loss in product quality.



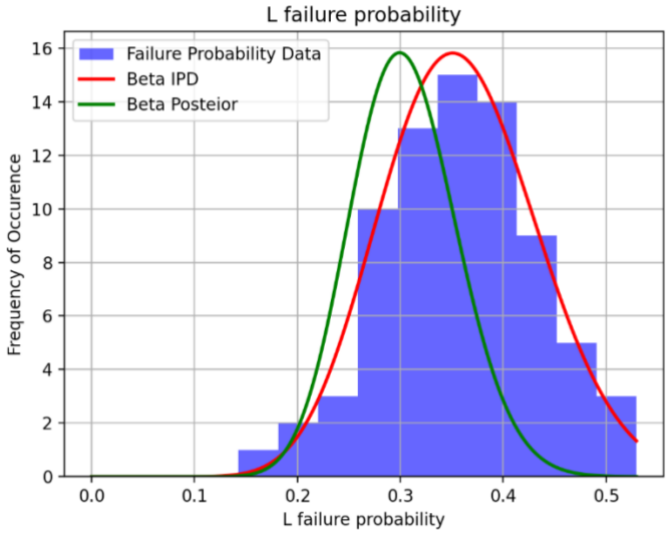
(a)



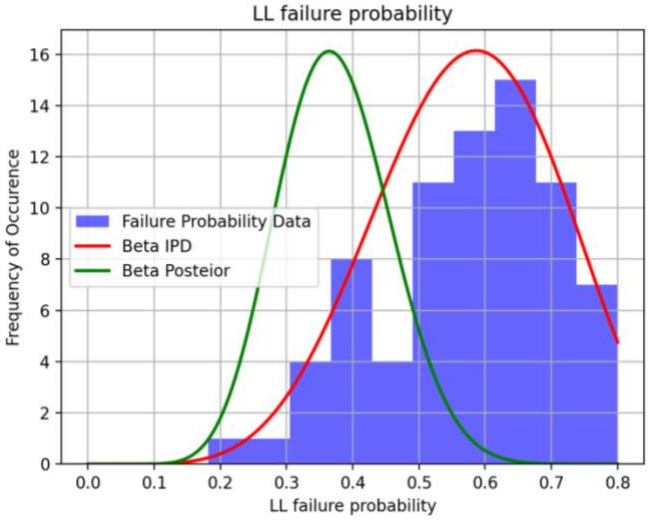
(b)



(c)



(d)



(e)

Figure 15. Results generated from DRAn studies, consisting of histogram for failure probability data, Beta IPD, and Beta Posterior for the five alarm systems: a) H; b) HH; c) HHH; d) L; e) LL.

Table 7. Pearson Correlation Coefficients between Average Committer Probabilities and Response-Action Variables

Response-Action Variable	Pearson Coefficient for average $p_{B,Unsafe}$	Pearson Coefficient for average $p_{B,Unreliable}$
q_i	0.9952	0.1464
q_m	0.9897	0.5937
q_s	-0.9891	0.6951

Table 8. Average Alarm Statistics generated based on DRAn studies for q_i and q_m as Response-Action Variables

DRAn Alarm Statistic (Average)	Response-Action Variable: q_i	Response-Action Variable: q_m
H Failure Probability	0.2732	0.4616
HH Failure Probability	0.1391	0.2022
HHH Failure Probability	0.055	-
L Failure Probability	0.3583	0.0754
LL Failure Probability	0.5724	0.6019
Number of Unsafe Crossings	7.0133	1.5857
Number of Unreliable Crossings	7.76	7.1571
Growing Polymer Concentration (x_5)	2.338×10^{-6}	2.336×10^{-6}

4. CONCLUSIONS

In our previous research, novel, improved, multivariate alarm systems were developed for rare un-postulated abnormal events, using random statistical noise in one or more process variables – demonstrated for an exothermic CSTR (Sudarshan et al., 2023). But, their alarm thresholds were based on static process variable limits, often leading to increased nuisance alarms and/or missed alarms.

Herein, our methods are extended for a more complex and realistic PID-controlled polystyrene CSTR model, involving abnormal shifts from a desirable unstable region towards two undesirable unsafe and unreliable operating regions. As a BG-FFS algorithm locates multiple rare abnormal trajectories towards these two undesirable regions, key process variables and committer probabilities, p_B, are obtained. Then, the XGBoost ML algorithm is utilized, resulting in accurate and reliable predictive models for p_B as a function of the key process variables. The XGBoost models are deployed in real-time to develop initial dynamic bidirectional multivariate alarm systems based on p_B predictions, a major improvement as compared with the static unidirectional systems developed previously for the exothermic CSTR. Additionally, note the direct influence of the XGBoost models' predictions on decision-making, i.e., varying the response action variables based on the predicted p_B, addressing the decision science aspect of machine learning and risk assessment. Then, rationalization strategies developed previously are utilized to evaluate and modify the initial alarm systems – with q_i selected as the response-action variable, a five-level

alarm system; i.e., H-HH-HHH-L-LL, is preferred. Next, DRAn studies are performed, based on which, the high-alarm systems appear to be reasonably effective, with decreasing trends closer to the unsafe region; whereas, due to increasing failure probabilities, the low-alarm systems are shown to require improvements. Then, sensitivity analyses are shown to be helpful – as they investigate the impact of other potential response-action variables; i.e., q_m and q_s.

5. FUTURE WORK

In spite of improved findings herein, limitations remain to be addressed. This section addresses two possibilities for future research.

5.1. Exploring Further Improvements to the Multivariate Alarm Systems

In this paper, the multivariate alarm systems developed consisted of single response-action variables for abnormal transitions to both unsafe as well as unreliable regions. However, it is likely that considering multiple variables is beneficial – e.g., when variable c_1 has greater influence on committer probabilities towards the unsafe region, and variable c_2 has greater influence on committer probabilities towards the unreliable regions. However, the increased computational costs, as well as the impact of this approach on safety, need to be considered (e.g., as observed in Section 3.5., decreasing q_s increases the reliability of the process, but is detrimental to safety) i.e., the FFS simulations need to be repeated for multiple discrete combinations of the response action variables.

Additionally, note that the alarm rationalization framework introduced previously in Sudarshan et al. (2024), and demonstrated in Section 3.4., herein, involved experimenting with empirically-chosen combinations of alarm thresholds and response actions – a process that can be very cumbersome and inconvenient. Moreover, it is important to enhance the rationalization process by exploring automated or semi-automated techniques, capable of selecting optimum alarm thresholds and response actions more-intelligently.

Moreover, as more-advanced control schemes are becoming popular (e.g., MPC, reinforcement-learning control, and the like), attempts should be made to analyze the impact of statistical noise sensitivity on controller offset (as observed in Figure 7), as well as impact of more-robust control schemes on generating sufficient un-postulated abnormal events using FFS.

5.2. Hybrid Modeling to Incorporate Plant Data and Regular Model Refinements

As observed in Figure 4, the ML pipeline is an infinite cycle, with regular model refinements enabled by feedback, as more data are recorded. However, herein, a feedback loop to refine the alarm systems by updating XGBoost model predictions does not exist due to lack of plant data. In future research, hybrid computational models (e.g., physics-informed neural networks; i.e., PINNs) involving underlying physics (i.e., material and energy balances, kinetics, transport phenomena, and the like), coupled with plant data from open-source databases and/or industrial collaborations, should be developed. Such hybrid models can simulate abnormal trajectories via FFS, followed by ML-based predictive-models that facilitate regular updates of the alarm systems as data are received.

5.3. More Advanced Models for Unpostulated Abnormal Events

To model un-postulated abnormal events herein, statistical noise is utilized, with random samples drawn from the normal distribution, with specified means and variances. Additionally, for the polystyrene CSTR operating in the unstable region, given adequate statistical noise, rare un-postulated abnormal shifts can lead to only two regions; i.e., unsafe and unreliable. But, for some abnormal events (or combinations/sequences of events), other distributions are preferable. In future research, more advanced abnormal event models should be considered; e.g., using mixed probability distributions, trained using random samples from distributions involving multiple parameters.

6. ACKNOWLEDGEMENTS

The NSF CBET funding for our EAGER GOALI grant, 1839535, and the NSF CBET funding for our grant, 2220276, are greatly appreciated. Also, the suggestions and advice of Prof. Talid Sinno, Prof. Masoud Soroush, and Prof. Manfred Morari, were very helpful.

7. APPENDIX

This Appendix contains three subsections.

7.1. Abbreviations (Acronyms)

AI	Artificial Intelligence
AIBN	Azobisisobutyronitrile
ASU	Air Separation Unit
BG-FFS	Branched-Growth Forward-Flux Sampling
CPU	Central Processing Unit
CSTR	Continuous Stirred Tank Reactor
DFFS	Direct Forward-Flux Sampling
DRAn	Dynamic Risk Analyses
FFS	Forward-Flux Sampling
GPU	Graphical Processing Unit
HAZOP	Hazard and Operability Study
IPD	Informed Prior Distribution
IoT	Internet of Things
LLM	Large Language Model
MCMC	Markov-Chain Monte-Carlo
MD	Molecular Dynamics
ML	Machine Learning
NMPC	Nonlinear Model Predictive Control
OLS	Ordinary Least Squares
PID	Proportional Integral Derivative
RMSE	Root Mean Squared Error
SIS	Safety Instrumented Systems
TPE	Tree-structured Parzen Estimator
TPS	Transition-Path Sampling

TST	Time Series Transformer
XGBoost	eXtreme Gradient Boosting

7.2. XGBoost Algorithm: Steps

Given the training data; i.e., $\{(x_i,\ y_i)\}_{i=1}^{N_{\text{samples}}}$; number of decision trees in the ensemble; i.e., *M*; the learning rate; i.e., α (i.e., a measure of the contribution of the previous tree to the newly trained tree); and a differentiable loss function; i.e., $L\left(y, \hat{f}(x)\right)$ (e.g., RMSE in Eq. (28)); the XGBoost algorithm proceeds as (Chen and Guestrin, 2016):

1. Develop a *weak* initial model:

$$\hat{f}_0(x) = \arg\min_{\hat{f}} \sum_{i=1}^{N} L\left(y_i, \hat{f}\right) + \ \Omega\left(\hat{f}\right) \tag{36}$$

where $\Omega\left(\hat{f}\right)$ is the *regularization* term, typically utilized to reduce overfitting (i.e., low errors on in-sample training data, but high errors on out-of-sample testing data, leading to poor predictive models).

2. Repeat following steps for *m* = 1 to *M:*

2.1. Compute Gradients and Hessians:

$$\hat{g}_m(x_i) = \frac{\partial L\left(y_i, \hat{f}_{(m-1)}(x_i)\right)}{\partial \hat{f}_{(m-1)}(x_i)} \tag{37}$$

$$\hat{h}_m(x_i) = \frac{\partial^2 L\left(y_i, \hat{f}_{(m-1)}(x_i)\right)}{\partial \hat{f}^2{}_{(m-1)}(x_i)} \tag{38}$$

2.2. Train decision tree *m* for the training data $\left\{x_i, -\frac{\hat{g}_m(x_i)}{\hat{h}_m(x_i)}\right\}_{i=1}^{N}$ by solving the following optimization problem:

$$\hat{\phi}_m(x) = \arg\min_{\phi} \left[\sum_{i=1}^{N} \frac{1}{2} h_m(x) \left(\phi(x_i) - \frac{\hat{g}_m(x_i)}{\hat{h}_m(x_i)}\right)^2 + \ \Omega(\phi)\right] \tag{39}$$

$$\hat{f}_m(x) = \ \alpha\hat{\phi}_m(x) \tag{40}$$

2.3. Update the model:

$$\hat{f}_{(m)}(x) = f_{(m-1)}(x_i) + \ \alpha\hat{\phi}_m(x) \tag{41}$$

3. Compute final model:

$$\hat{f}(x) = \hat{f}_{(M)}(x) = \sum_{m=0}^{M} \hat{f}_m(x) \tag{42}$$

7.3. Computed Parameters for Beta IPDs and Assumptions for Likelihood Distributions for q_i-based Alarm Systems

Consider the following $Beta\ (a_i,\ b_i)$ IPD for the failure probability of alarm system i, $p_{\text{failure},i}$:

$$f\left(p_{\text{failure},i}\right) = Beta\ (a_i,\ b_i) = \frac{\Gamma(a_i + b_i)}{\Gamma(a_i)\Gamma(b_i)}\ {p_{\text{failure},i}}^{a_i - 1}\left(1 - \ p_{\text{failure},i}\right)^{b_i - 1};\ \ a_i > 0; b_i > \ 0 \tag{43}$$

where the gamma function, $\Gamma(a_i)$, is:

$$\Gamma(a_i) = \int_0^\infty t^{a_i\,-\,1} e^{-t}\, dt \tag{44}$$

The a_i and b_i parameters for the Beta distribution are:

$$a_i = \mu_i\left(\frac{\mu_i(1-\mu_i)}{{\sigma_i}^2} - 1\right),\ \ if\ {\sigma_i}^2\ < \mu_i(1-\mu_i) \tag{45}$$

$$b_i\ = (1-\mu_i)\left(\frac{\mu_i(1-\mu_i)}{{\sigma_i}^2} - 1\right),\ \ if\ {\sigma_i}^2\ < \mu_i(1-\mu_i) \tag{46}$$

where μ_i and ${\sigma_i}^2$ represent the mean and variance of the failure probabilities for alarm system i. Note that i corresponds to the alarm system level; i.e., i = H, HH, L, LL, and the like. Table 9 shows the Beta distribution parameters computed for each of the five q_i – based alarm systems.

Typically, the likelihood distribution; i.e., $f\left(D\middle|p_{\text{failure},i}\right)$, is developed utilizing available alarm data; i.e., D. However, given the unavailability of alarm data, likelihood distributions need to be assumed for each of the five q_i-based multivariate alarm systems; i.e., H-HH-HHH-L-LL. Hence, for each, a binomial likelihood distribution is assumed:

$$\begin{aligned} f\left(D\middle|p_{\text{failure},i}\right) &= Binomial\left(n_{\text{binom},i}, k_{\text{binom},i}\right) \\ &= \frac{n_{\text{binom},i}!}{\left(n_{\text{binom},i} - k_{\text{binom},i}\right)!}\ {p_{\text{failure},i}}^{k_{\text{binom},i}}\ \left(1 - p_{\text{failure},i}\right)^{n_{\text{binom},i}\,-\,k_{\text{binom},i}} \end{aligned} \tag{47}$$

where $n_{\text{binom},i}$ is the number of instances an alarm went off at a level i (stated differently, the number of instances a particular alarm system was activated; e.g., the H alarm system); $k_{\text{binom},i}$ is the number of failures corresponding to alarm system i. Table 10 shows the parameters for the binomial likelihood distributions assumed for each of the five alarm q_i – based systems. Note the general decreasing trend assumed for the parameters as the undesirable region is approached.

Table 9. Parameters for Beta IPDs Computed for q_i-based Five-level (i.e., H-HH-HH-L-LL) Alarm Systems

Alarm System Level	a_i (Beta IPD Parameter)	b_i (Beta IPD Parameter)
H	6.94	18.47
HH	5.16	31.91
HHH	4.90	84.13
L	14.44	25.87
LL	7.05	5.27

Table 10. Parameters for Binomial Likelihood Distributions Assumed for q_i-based Five-level (i.e., H-HH-HH-L-LL) Alarm Systems

Alarm System Level	$n_{binom,i}$ (Binomial Distribution Parameter)	$k_{binom,i}$ (Binomial Distribution Parameter)
H	75	10
HH	60	5
HHH	30	0
L	40	10
LL	20	5

7.4. Hyperparameters for XGBoost Model

Hyperparameters of ML models are parameters that are external to the model training process – stated differently, these parameters are required to be set/optimized before the training process, unlike internal model parameters optimized during training. Appropriate selection of hyperparameters is crucial for the predictive performance of ML models; poor choice can lead to unsatisfactory performance, such as overfitting, i.e., poor performance on test data, despite the model performing well on the train data.

Note that XGBoost consists of several hyperparameters – optimizing every hyperparameter is computationally expensive, and possibly inefficient as well. Hence, in this paper, only a few key hyperparameters crucial to prevent overfitting were optimized using Bayesian Optimization through the Optuna package in Python, while the rest remained at their default values:

i) *n_estimators*: Number of decision trees in the ensemble
ii) *learning_rate*: Scaling factor that determines contribution of each decision tree in the ensemble.
iii) *subsample*: Fraction of data used in training each decision tree – minimizes overfitting by introducing randomness
iv) *reg_alpha*: Parameter for L1 regularization – minimizes overfitting by penalizing large coefficients.
v) *reg_lambda*: Parameter for L2 regularization – minimizes overfitting by penalizing sum of squared coefficients.

vi) *max_delta_step*: Maximum step-size permitted for a leaf output in each decision tree.

For more information regarding these and other hyperparameters, the readers are encouraged to refer to the official documentation for XGBoost (Chen and Guestrin, 2016; xgboost developers, 2023).

8. REFERENCES

Ahooyi Mohseni, T., Soroush, M., Arbogast, J.E., Seider, W.D., Oktem, U.G., 2014. Maximum-likelihood maximum-entropy constrained probability density function estimation for prediction of rare events. AIChE J. 60, 1013–1026. https://doi.org/10.1002/aic.14330

Akiba, T., Sano, S., Yanase, T., Ohta, T., Koyama, M., 2019. Optuna: A Next-generation Hyperparameter Optimization Framework. https://doi.org/10.48550/arXiv.1907.10902

Allen, R.J., Frenkel, D., ten Wolde, P.R., 2006. Simulating rare events in equilibrium or nonequilibrium stochastic systems. J. Chem. Phys. 124, 024102. https://doi.org/10.1063/1.2140273

Allen, R.J., Valeriani, C., Rein ten Wolde, P., 2009. Forward flux sampling for rare event simulations. J. Phys. Condens. Matter 21, 463102. https://doi.org/10.1088/0953-8984/21/46/463102

Angelopoulos, A., Michailidis, E.T., Nomikos, N., Trakadas, P., Hatziefremidis, A., Voliotis, S., Zahariadis, T., 2020. Tackling Faults in the Industry 4.0 Era—A Survey of Machine-Learning Solutions and Key Aspects. Sensors 20, 109. https://doi.org/10.3390/s20010109

Arjun, A., Bolhuis, P.G., 2023. Homogeneous nucleation of crystalline methane hydrate in molecular dynamics transition paths sampled under realistic conditions. J. Chem. Phys. 158, 044504. https://doi.org/10.1063/5.0124852

Arunthavanathan, R., Ahmed, S., Khan, F., Imtiaz, S., 2022. Machine Learning for Process Fault Detection and Diagnosis, in: Machine Learning in Chemical Safety and Health. John Wiley & Sons, Ltd, pp. 113–137. https://doi.org/10.1002/9781119817512.ch6

Ashraf, M.T., Dey, K., Mishra, S., 2023. Identification of high-risk roadway segments for wrong-way driving crash using rare event modeling and data augmentation techniques. Accid. Anal. Prev. 181, 106933. https://doi.org/10.1016/j.aap.2022.106933

Aven, T., 2020. Chapter Six - Rare event risk assessments, in: Khan, F.I., Amyotte, P.R. (Eds.), Methods in Chemical Process Safety, Advanced Methods of Risk Assessment and Management. Elsevier, pp. 205–237. https://doi.org/10.1016/bs.mcps.2020.02.003

Barata, J., Kayser, I., 2023. Industry 5.0 – Past, Present, and Near Future. Procedia Comput. Sci., CENTERIS – International Conference on ENTERprise Information Systems / ProjMAN – International Conference on Project MANagement / HCist – International Conference on Health and Social Care Information Systems and Technologies 2022 219, 778–788. https://doi.org/10.1016/j.procs.2023.01.351

Beniston, M., Stephenson, D.B., Christensen, O.B., Ferro, C.A.T., Frei, C., Goyette, S., Halsnaes, K., Holt, T., Jylhä, K., Koffi, B., Palutikof, J., Schöll, R., Semmler, T., Woth, K., 2007. Future extreme events in European climate: an exploration of regional climate model projections. Clim. Change 81, 71–95. https://doi.org/10.1007/s10584-006-9226-z

Bergstra, J., Bardenet, R., Bengio, Y., Kégl, B., 2011. Algorithms for Hyper-Parameter Optimization, in: Advances in Neural Information Processing Systems. Curran Associates, Inc.

https://proceedings.neurips.cc/paper_files/paper/2011/hash/86e8f7ab32cfd12577bc2619bc635690-Abstract.html

Bergstra, J., Komer, B., Eliasmith, C., Yamins, D., Cox, D.D., 2015. Hyperopt: a Python library for model selection and hyperparameter optimization. Comput. Sci. Discov. 8, 014008. https://doi.org/10.1088/1749-4699/8/1/014008

Berne, B.J., 1985. 13 - Molecular Dynamics and Monte Carlo Simulations of Rare Events, in: Brackbill, J.U., Cohen, B.I. (Eds.), Multiple Time Scales. Academic Press, pp. 419–436. https://doi.org/10.1016/B978-0-12-123420-1.50018-5

Bhaumik, D.K., Amatya, A., Normand, S.-L.T., Greenhouse, J., Kaizar, E., Neelon, B., Gibbons, R.D., 2012. Meta-Analysis of Rare Binary Adverse Event Data. J. Am. Stat. Assoc. 107, 555–567. https://doi.org/10.1080/01621459.2012.664484

Bi, Y., Li, T., 2014. Probing Methane Hydrate Nucleation through the Forward Flux Sampling Method. J. Phys. Chem. B 118, 13324–13332. https://doi.org/10.1021/jp503000u

Borisov, V., Leemann, T., Seßler, K., Haug, J., Pawelczyk, M., Kasneci, G., 2022. Deep Neural Networks and Tabular Data: A Survey. IEEE Trans. Neural Netw. Learn. Syst. 1–21. https://doi.org/10.1109/TNNLS.2022.3229161

Borrero, E.E., Escobedo, F.A., 2007. Reaction coordinates and transition pathways of rare events via forward flux sampling. J. Chem. Phys. 127, 164101. https://doi.org/10.1063/1.2776270

Breiman, L., Friedman, J., Olshen, R.A., Stone, C.J., 2017. Chapter 8: Regression Trees, in: Classification and Regression Trees. Chapman and Hall/CRC.

Brooks, B.W., 1981. Dynamic behaviour of a continuous-flow polymerisation reactor. Chem. Eng. Sci. 36, 589–593. https://doi.org/10.1016/0009-2509(81)80147-9

Bzdok, D., 2017. Classical Statistics and Statistical Learning in Imaging Neuroscience. Front. Neurosci. 11. https://doi.org/10.3389/fnins.2017.00543

Bzdok, D., Altman, N., Krzywinski, M., 2018. Statistics versus machine learning. Nat. Methods 15, 233–234. https://doi.org/10.1038/nmeth.4642

Bzdok, D., Krzywinski, M., Altman, N., 2017. Machine learning: A primer. Nat. Methods 14, 1119–1120. https://doi.org/10.1038/nmeth.4526

Cai, T., Parast, L., Ryan, L., 2010. Meta-analysis for rare events. Stat. Med. 29, 2078–2089. https://doi.org/10.1002/sim.3964

Candanedo, I.S., Nieves, E.H., González, S.R., Martín, M.T.S., Briones, A.G., 2018. Machine Learning Predictive Model for Industry 4.0, in: Uden, L., Hadzima, B., Ting, I.-H. (Eds.), Knowledge Management in Organizations, Communications in Computer and Information Science. Springer International Publishing, Cham, pp. 501–510. https://doi.org/10.1007/978-3-319-95204-8_42

Carvalho, D.V., Pereira, E.M., Cardoso, J.S., 2019. Machine Learning Interpretability: A Survey on Methods and Metrics. Electronics 8, 832. https://doi.org/10.3390/electronics8080832

Cerna, S., Guyeux, C., Arcolezi, H.H., Couturier, R., Royer, G., 2020. A Comparison of LSTM and XGBoost for Predicting Firemen Interventions, in: Rocha, Á., Adeli, H., Reis, L.P., Costanzo, S., Orovic, I., Moreira, F. (Eds.), Trends and Innovations in Information Systems and Technologies, Advances in Intelligent Systems and Computing. Springer International Publishing, Cham, pp. 424–434. https://doi.org/10.1007/978-3-030-45691-7_39

Chen, T., Guestrin, C., 2016. XGBoost: A Scalable Tree Boosting System, in: Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. pp. 785–794. https://doi.org/10.1145/2939672.2939785

Ciccotti, G., Ferrario, M., 2000. Rare events by constrained molecular dynamics. J. Mol. Liq. 89, 1–18. https://doi.org/10.1016/S0167-7322(00)90001-1

Çınar, Z.M., Abdussalam Nuhu, A., Zeeshan, Q., Korhan, O., Asmael, M., Safaei, B., 2020. Machine Learning in Predictive Maintenance towards Sustainable Smart Manufacturing in Industry 4.0. Sustainability 12, 8211. https://doi.org/10.3390/su12198211

Claesen, M., Simm, J., Popovic, D., Moreau, Y., De Moor, B., 2014. Easy Hyperparameter Search Using Optunity. https://doi.org/10.48550/arXiv.1412.1114

Culot, G., Fattori, F., Podrecca, M., Sartor, M., 2019. Addressing Industry 4.0 Cybersecurity Challenges. IEEE Eng. Manag. Rev. 47, 79–86. https://doi.org/10.1109/EMR.2019.2927559

Demir, K.A., Döven, G., Sezen, B., 2019. Industry 5.0 and Human-Robot Co-working. Procedia Comput. Sci., 3rd World Conference on Technology, Innovation and Entrepreneurship "Industry 4.0 Focused Innovation, Technology, Entrepreneurship and Manufacture" June 21-23, 2019 158, 688–695. https://doi.org/10.1016/j.procs.2019.09.104

Dhooge, A., Govaerts, W., Kuznetsov, Yu.A., 2003. MATCONT: A MATLAB package for numerical bifurcation analysis of ODEs. ACM Trans. Math. Softw. 29, 141–164. https://doi.org/10.1145/779359.779362

Dingli, A., Haddod, F., Klüver, C. (Eds.), 2021. Artificial Intelligence in Industry 4.0: A Collection of Innovative Research Case-studies that are Reworking the Way We Look at Industry 4.0 Thanks to Artificial Intelligence, Studies in Computational Intelligence. Springer International Publishing, Cham. https://doi.org/10.1007/978-3-030-61045-6

Domova, V., Dagnino, A., 2017. Towards intelligent alarm management in the Age of IIoT, in: 2017 Global Internet of Things Summit (GIoTS). Presented at the 2017 Global Internet of Things Summit (GIoTS), pp. 1–5. https://doi.org/10.1109/GIOTS.2017.8016234

Donnenberg, A.D., Donnenberg, V.S., 2007. Rare-Event Analysis in Flow Cytometry. Clin. Lab. Med., Flow Cytometry 27, 627–652. https://doi.org/10.1016/j.cll.2007.05.013

Ervural, B.C., Ervural, B., 2018. Overview of Cyber Security in the Industry 4.0 Era, in: Ustundag, A., Cevikcan, E. (Eds.), Industry 4.0: Managing The Digital Transformation, Springer Series in Advanced Manufacturing. Springer International Publishing, Cham, pp. 267–284. https://doi.org/10.1007/978-3-319-57870-5_16

Filion, L., Hermes, M., Ni, R., Dijkstra, M., 2010. Crystal nucleation of hard spheres using molecular dynamics, umbrella sampling, and forward flux sampling: A comparison of simulation techniques. J. Chem. Phys. 133, 244115. https://doi.org/10.1063/1.3506838

Gazi, E., Seider, W.D., Ungar, L.H., 1996. Verification of Controllers in the Presence of Uncertainty: Application to Styrene Polymerization. Ind. Eng. Chem. Res. 35, 2277–2287. https://doi.org/10.1021/ie9504361

Gelman, A., Carlin, J.B., Stern, H.S., Dunson, D.B., Vehtari, A., Rubin, D.B., 2013. Bayesian Data Analysis, 0 ed. Chapman and Hall/CRC. https://doi.org/10.1201/b16018

Ghobakhloo, M., 2018. The future of manufacturing industry: a strategic roadmap toward Industry 4.0. J. Manuf. Technol. Manag. 29, 910–936. https://doi.org/10.1108/JMTM-02-2018-0057

Ghobakhloo, M., Iranmanesh, M., Tseng, M.-L., Grybauskas, A., Stefanini, A., Amran, A., 2023. Behind the definition of Industry 5.0: a systematic review of technologies, principles, components, and values. J. Ind. Prod. Eng. 40, 432–447. https://doi.org/10.1080/21681015.2023.2216701

Gokalp, M.O., Kayabay, K., Akyol, M.A., Eren, P.E., Koçyiğit, A., 2016. Big Data for Industry 4.0: A Conceptual Framework, in: 2016 International Conference on Computational Science and

Computational Intelligence (CSCI). Presented at the 2016 International Conference on Computational Science and Computational Intelligence (CSCI), pp. 431–434. https://doi.org/10.1109/CSCI.2016.0088

Grinsztajn, L., Oyallon, E., Varoquaux, G., 2022. Why do tree-based models still outperform deep learning on tabular data? https://doi.org/10.48550/arXiv.2207.08815

Hanna, B.N., Dinh, N.T., Youngblood, R.W., Bolotnov, I.A., 2020. Machine-learning based error prediction approach for coarse-grid Computational Fluid Dynamics (CG-CFD). Prog. Nucl. Energy 118, 103140. https://doi.org/10.1016/j.pnucene.2019.103140

Harris, C.R., Millman, K.J., van der Walt, S.J., Gommers, R., Virtanen, P., Cournapeau, D., Wieser, E., Taylor, J., Berg, S., Smith, N.J., Kern, R., Picus, M., Hoyer, S., van Kerkwijk, M.H., Brett, M., Haldane, A., del Río, J.F., Wiebe, M., Peterson, P., Gérard-Marchant, P., Sheppard, K., Reddy, T., Weckesser, W., Abbasi, H., Gohlke, C., Oliphant, T.E., 2020. Array programming with NumPy. Nature 585, 357–362. https://doi.org/10.1038/s41586-020-2649-2

Hartmann, C., Banisch, R., Sarich, M., Badowski, T., Schütte, C., 2014. Characterization of Rare Events in Molecular Dynamics. Entropy 16, 350–376. https://doi.org/10.3390/e16010350

Hidalgo, P.M., Brosilow, C.B., 1990. Nonlinear model predictive control of styrene polymerization at unstable operating points. Comput. Chem. Eng. 14, 481–494. https://doi.org/10.1016/0098-1354(90)87022-H

Hunter, J.D., 2007. Matplotlib: A 2D Graphics Environment. Comput. Sci. Eng. 9, 90–95. https://doi.org/10.1109/MCSE.2007.55

Jalali, B., Solli, D.R., Goda, K., Tsia, K., Ropers, C., 2010. Real-time measurements, rare events and photon economics. Eur. Phys. J. Spec. Top. 185, 145–157. https://doi.org/10.1140/epjst/e2010-01245-8

Javaid, M., Haleem, A., Singh, R.P., Suman, R., 2022. Artificial Intelligence Applications for Industry 4.0: A Literature-Based Study. J. Ind. Integr. Manag. 07, 83–111. https://doi.org/10.1142/S2424862221300040

Jiang, H., Haji-Akbari, A., Debenedetti, P.G., Panagiotopoulos, A.Z., 2018. Forward flux sampling calculation of homogeneous nucleation rates from aqueous NaCl solutions. J. Chem. Phys. 148, 044505. https://doi.org/10.1063/1.5016554

Kim, K.J., Choi, K.Y., Alexander, J.C., 1991. Dynamics of a cascade of two continuous stirred tank polymerization reactors with a binary initiator mixture. Polym. Eng. Sci. 31, 333–352. https://doi.org/10.1002/pen.760310506

Kirch, W. (Ed.), 2008. Pearson's Correlation Coefficient, in: Encyclopedia of Public Health. Springer Netherlands, Dordrecht, pp. 1090–1091. https://doi.org/10.1007/978-1-4020-5614-7_2569

Kitchin, J.R., 2018. Machine learning in catalysis. Nat. Catal. 1, 230–232. https://doi.org/10.1038/s41929-018-0056-y

Kleinmuntz, B., 1990. Why we still use our heads instead of formulas: Toward an integrative approach. Psychol. Bull. 107, 296–310. https://doi.org/10.1037/0033-2909.107.3.296

Kochkov, D., Smith, J.A., Alieva, A., Wang, Q., Brenner, M.P., Hoyer, S., 2021. Machine learning–accelerated computational fluid dynamics. Proc. Natl. Acad. Sci. 118, e2101784118. https://doi.org/10.1073/pnas.2101784118

Kotsiopoulos, T., Sarigiannidis, P., Ioannidis, D., Tzovaras, D., 2021. Machine Learning and Deep Learning in smart manufacturing: The Smart Grid paradigm. Comput. Sci. Rev. 40, 100341. https://doi.org/10.1016/j.cosrev.2020.100341

Koza, J.R., Bennett, F.H., Andre, D., Keane, M.A., 1996. Automated Design of Both the Topology and Sizing of Analog Electrical Circuits Using Genetic Programming, in: Gero, J.S., Sudweeks, F. (Eds.), Artificial Intelligence in Design '96. Springer Netherlands, Dordrecht, pp. 151–170. https://doi.org/10.1007/978-94-009-0279-4_9

Kubíček, M., Marek, M., 1983. Computational Methods in Bifurcation Theory and Dissipative Structures. Springer, Berlin, Heidelberg. https://doi.org/10.1007/978-3-642-85957-1

Kumari, P., Bhadriraju, B., Wang, Q., Kwon, J.S.-I., 2021. Development of parametric reduced-order model for consequence estimation of rare events. Chem. Eng. Res. Des. 169, 142–152. https://doi.org/10.1016/j.cherd.2021.02.006

Kumari, P., Lee, D., Wang, Q., Karim, M.N., Sang-Il Kwon, J., 2020. Root Cause Analysis of Key Process Variable Deviation for Rare Events in the Chemical Process Industry. Ind. Eng. Chem. Res. 59, 10987–10999. https://doi.org/10.1021/acs.iecr.0c00624

Lam, S.K., Pitrou, A., Seibert, S., 2015. Numba: a LLVM-based Python JIT compiler, in: Proceedings of the Second Workshop on the LLVM Compiler Infrastructure in HPC, LLVM '15. Association for Computing Machinery, New York, NY, USA, pp. 1–6. https://doi.org/10.1145/2833157.2833162

Lee, C., Lim, C., 2021. From technological development to social advance: A review of Industry 4.0 through machine learning. Technol. Forecast. Soc. Change 167, 120653. https://doi.org/10.1016/j.techfore.2021.120653

Lee, J., Davari, H., Singh, J., Pandhare, V., 2018. Industrial Artificial Intelligence for industry 4.0-based manufacturing systems. Manuf. Lett. 18, 20–23. https://doi.org/10.1016/j.mfglet.2018.09.002

Lezzi, M., Lazoi, M., Corallo, A., 2018. Cybersecurity for Industry 4.0 in the current literature: A reference framework. Comput. Ind. 103, 97–110. https://doi.org/10.1016/j.compind.2018.09.004

Li, W., Yin, Y., Quan, X., Zhang, H., 2019. Gene Expression Value Prediction Based on XGBoost Algorithm. Front. Genet. 10. https://doi.org/10.3389/fgene.2019.01077

Liaw, R., Liang, E., Nishihara, R., Moritz, P., Gonzalez, J.E., Stoica, I., 2018. Tune: A Research Platform for Distributed Model Selection and Training. https://doi.org/10.48550/arXiv.1807.05118

Ma, M., Zhao, G., He, B., Li, Q., Dong, H., Wang, S., Wang, Z., 2021. XGBoost-based method for flash flood risk assessment. J. Hydrol. 598, 126382. https://doi.org/10.1016/j.jhydrol.2021.126382

Manavalan, E., Jayakrishna, K., 2019. A review of Internet of Things (IoT) embedded sustainable supply chain for industry 4.0 requirements. Comput. Ind. Eng. 127, 925–953. https://doi.org/10.1016/j.cie.2018.11.030

Maqbool, R., Saiba, M.R., Ashfaq, S., 2023. Emerging industry 4.0 and Internet of Things (IoT) technologies in the Ghanaian construction industry: sustainability, implementation challenges, and benefits. Environ. Sci. Pollut. Res. Int. 30, 37076–37091. https://doi.org/10.1007/s11356-022-24764-1

McKinney, W., 2010. Data Structures for Statistical Computing in Python. Proc. 9th Python Sci. Conf. 56–61. https://doi.org/10.25080/Majora-92bf1922-00a

Moskowitz, I.H., Seider, W.D., Patel, A.J., Arbogast, J.E., Oktem, U.G., 2018. Understanding rare safety and reliability events using transition path sampling. Comput. Chem. Eng. 108, 74–88. https://doi.org/10.1016/j.compchemeng.2017.06.016

Motz, M., Krauß, J., Schmitt, R.H., 2022. Benchmarking of hyperparameter optimization techniques for machine learning applications in production. Adv. Ind. Manuf. Eng. 5, 100099. https://doi.org/10.1016/j.aime.2022.100099

Nguyen, T., Gosine, R.G., Warrian, P., 2020. A Systematic Review of Big Data Analytics for Oil and Gas Industry 4.0. IEEE Access 8, 61183–61201. https://doi.org/10.1109/ACCESS.2020.2979678

NVIDIA, Vingelmann, P., Fitzek, F.H.P., 2022. NVIDIA CUDA Toolkit 11.8.90. https://developer.download.nvidia.com/compute/cuda/11.8.0/docs/sidebar/CUDA_Toolkit_Release_Notes.pdf

Ogunleye, A., Wang, Q.-G., 2020. XGBoost Model for Chronic Kidney Disease Diagnosis. IEEE/ACM Trans. Comput. Biol. Bioinform. 17, 2131–2140. https://doi.org/10.1109/TCBB.2019.2911071

Pariyani, A., Seider, W.D., Oktem, U.G., Soroush, M., 2012a. Dynamic risk analysis using alarm databases to improve process safety and product quality: Part I-Data compaction. AIChE J. 58, 812–825. https://doi.org/10.1002/aic.12643

Pariyani, A., Seider, W.D., Oktem, U.G., Soroush, M., 2012b. Dynamic risk analysis using alarm databases to improve process safety and product quality: Part II-Bayesian analysis. AIChE J. 58, 826–841. https://doi.org/10.1002/aic.12642

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Müller, A., Nothman, J., Louppe, G., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., Duchesnay, É., 2018. Scikit-learn: Machine Learning in Python. https://doi.org/10.48550/arXiv.1201.0490

Peters, B., Trout, B.L., 2006. Obtaining reaction coordinates by likelihood maximization. J. Chem. Phys. 125, 054108. https://doi.org/10.1063/1.2234477

Rai, R., Tiwari, M.K., Ivanov, D., Dolgui, A., 2021. Machine learning in manufacturing and industry 4.0 applications. Int. J. Prod. Res. 59, 4773–4778. https://doi.org/10.1080/00207543.2021.1956675

Russo, L.P., Bequette, B.W., 1998. Operability of chemical reactors: multiplicity behavior of a jacketed styrene polymerization reactor. Chem. Eng. Sci. 53, 27–45. https://doi.org/10.1016/S0009-2509(97)00281-9

Sarich, M., Banisch, R., Hartmann, C., Schütte, C., 2014. Markov State Models for Rare Events in Molecular Dynamics. Entropy 16, 258–286. https://doi.org/10.3390/e16010258

Sarkar, S., Vinay, S., Raj, R., Maiti, J., Mitra, P., 2019. Application of optimized machine learning techniques for prediction of occupational accidents. Comput. Oper. Res. 106, 210–224. https://doi.org/10.1016/j.cor.2018.02.021

Sarker, I.H., 2021. Machine Learning: Algorithms, Real-World Applications and Research Directions. SN Comput. Sci. 2, 160. https://doi.org/10.1007/s42979-021-00592-x

Sharda, R., Delen, D., Turban, E., 2021. Analytics, Data Science, & Artificial Intelligence: Systems for Decision Support, 11th Global Edition. Pearson.

Shekhar, S., Bansode, A., Salim, A., 2022. A Comparative study of Hyper-Parameter Optimization Tools. https://doi.org/10.48550/arXiv.2201.06433

Shivpuje, S., Jaipal, M., Chatterjee, A., 2019. Accelerating rare events using temperature programmed molecular dynamics: a review. Mol. Simul. 45, 1295–1303. https://doi.org/10.1080/08927022.2019.1619929

Shwartz-Ziv, R., Armon, A., 2022. Tabular data: Deep learning is not all you need. Inf. Fusion 81, 84–90. https://doi.org/10.1016/j.inffus.2021.11.011

Sitapure, N., Kwon, J.S.-I., 2023a. Exploring the potential of time-series transformers for process modeling and control in chemical systems: An inevitable paradigm shift? Chem. Eng. Res. Des. 194, 461–477. https://doi.org/10.1016/j.cherd.2023.04.028

Sitapure, N., Kwon, J.S.-I., 2023b. CrystalGPT: Enhancing system-to-system transferability in crystallization prediction and control using time-series-transformers. Comput. Chem. Eng. 177, 108339. https://doi.org/10.1016/j.compchemeng.2023.108339

Soori, M., Arezoo, B., Dastres, R., 2023. Internet of things for smart factories in industry 4.0, a review. Internet Things Cyber-Phys. Syst. 3, 192–204. https://doi.org/10.1016/j.iotcps.2023.04.006

Stanley, H.E., Gabaix, X., Gopikrishnan, P., Plerou, V., 2007. Economic fluctuations and statistical physics: Quantifying extremely rare and less rare events in finance. Phys. Stat. Mech. Its Appl., Applications of Physics in Financial Analysis 382, 286–301. https://doi.org/10.1016/j.physa.2007.02.023

Sudarshan, V., Seider, W.D., Patel, A.J., Arbogast, J.E., 2021. Understanding rare safety and reliability events using forward-flux sampling. Comput. Chem. Eng. 153, 107387. https://doi.org/10.1016/j.compchemeng.2021.107387

Sudarshan, V., Seider, W.D., Patel, A.J., Oktem, U.G., Arbogast, J.E., 2024. Alarm rationalization and dynamic risk analyses for rare abnormal events. Comput. Chem. Eng. 184, 108633. https://doi.org/10.1016/j.compchemeng.2024.108633

Sudarshan, V., Seider, W.D., Patel, A.J., Oktem, U.G., Arbogast, J.E., 2023. Multivariate Alarm Systems To Recognize Rare Unpostulated Abnormal Events. AIChE J. e18284. https://doi.org/10.1002/aic.18284

Suleiman, Z., Shaikholla, S., Dikhanbayeva, D., Shehab, E., Turkyilmaz, A., 2022. Industry 4.0: Clustering of concepts and characteristics. Cogent Eng. 9, 2034264. https://doi.org/10.1080/23311916.2022.2034264

Tamascelli, N., Solini, R., Paltrinieri, N., Cozzani, V., 2022. Learning from major accidents: A machine learning approach. Comput. Chem. Eng. 162, 107786. https://doi.org/10.1016/j.compchemeng.2022.107786

Timms, C., 2009. Hazards equal trips or alarms or both. Process Saf. Environ. Prot. 87, 3–13. https://doi.org/10.1016/j.psep.2008.07.003

Toyao, T., Maeno, Z., Takakusagi, S., Kamachi, T., Takigawa, I., Shimizu, K., 2020. Machine Learning for Catalysis Informatics: Recent Applications and Prospects. ACS Catal. 10, 2260–2297. https://doi.org/10.1021/acscatal.9b04186

Tran, M.-Q., Elsisi, M., Mahmoud, K., Liu, M.-K., Lehtonen, M., Darwish, M.M.F., 2021. Experimental Setup for Online Fault Diagnosis of Induction Machines via Promising IoT and Machine Learning: Towards Industry 4.0 Empowerment. IEEE Access 9, 115429–115441. https://doi.org/10.1109/ACCESS.2021.3105297

Vaidya, S., Ambad, P., Bhosle, S., 2018. Industry 4.0 – A Glimpse. Procedia Manuf., 2nd International Conference on Materials, Manufacturing and Design Engineering (iCMMD2017), 11-12 December 2017, MIT Aurangabad, Maharashtra, INDIA 20, 233–238. https://doi.org/10.1016/j.promfg.2018.02.034

Varshney, K.R., Alemzadeh, H., 2017. On the Safety of Machine Learning: Cyber-Physical Systems, Decision Sciences, and Data Products. Big Data 5, 246–255. https://doi.org/10.1089/big.2016.0051

Virtanen, P., Gommers, R., Oliphant, T.E., Haberland, M., Reddy, T., Cournapeau, D., Burovski, E., Peterson, P., Weckesser, W., Bright, J., van der Walt, S.J., Brett, M., Wilson, J., Millman, K.J., Mayorov, N., Nelson, A.R.J., Jones, E., Kern, R., Larson, E., Carey, C.J., Polat, İ., Feng, Y., Moore, E.W., VanderPlas, J., Laxalde, D., Perktold, J., Cimrman, R., Henriksen, I., Quintero, E.A., Harris, C.R., Archibald, A.M., Ribeiro, A.H., Pedregosa, F., van Mulbregt, P., 2020. SciPy 1.0:

fundamental algorithms for scientific computing in Python. Nat. Methods 17, 261–272. https://doi.org/10.1038/s41592-019-0686-2

Vogel, G., Schulze Balhorn, L., Schweidtmann, A.M., 2023. Learning from flowsheets: A generative transformer model for autocompletion of flowsheets. Comput. Chem. Eng. 171, 108162. https://doi.org/10.1016/j.compchemeng.2023.108162

Wang, H., Ma, C., Zhou, L., 2009. A Brief Review of Machine Learning and Its Application, in: 2009 International Conference on Information Engineering and Computer Science. Presented at the 2009 International Conference on Information Engineering and Computer Science, pp. 1–4. https://doi.org/10.1109/ICIECS.2009.5362936

Wang, T., Bian, Y., Zhang, Y., Hou, X., 2023. Classification of earthquakes, explosions and mining-induced earthquakes based on XGBoost algorithm. Comput. Geosci. 170, 105242. https://doi.org/10.1016/j.cageo.2022.105242

Watanabe, S., 2023. Tree-Structured Parzen Estimator: Understanding Its Algorithm Components and Their Roles for Better Empirical Performance. https://doi.org/10.48550/arXiv.2304.11127

Webber, R.J., Plotkin, D.A., O'Neill, M.E., Abbot, D.S., Weare, J., 2019. Practical rare event sampling for extreme mesoscale weather. Chaos Interdiscip. J. Nonlinear Sci. 29, 053109. https://doi.org/10.1063/1.5081461

xgboost developers, 2023. xgboost Release 1.7.6. https://xgboost.readthedocs.io/_/downloads/en/release_1.7.0/pdf/

Yan, J., Meng, Y., Lu, L., Li, L., 2017. Industrial Big Data in an Industry 4.0 Environment: Challenges, Schemes, and Applications for Predictive Maintenance. IEEE Access 5, 23484–23491. https://doi.org/10.1109/ACCESS.2017.2765544

Yang, F., Gu, S., 2021. Industry 4.0, a revolution that requires technology and national strategies. Complex Intell. Syst. 7, 1311–1325. https://doi.org/10.1007/s40747-020-00267-9

Yang, M., Khan, F., Lye, L., Amyotte, P., 2015. Risk assessment of rare events. Process Saf. Environ. Prot. 98, 102–108. https://doi.org/10.1016/j.psep.2015.07.004

Yang, M., Khan, F.I., Lye, L., 2013. Precursor-based hierarchical Bayesian approach for rare event frequency estimation: A case of oil spill accidents. Process Saf. Environ. Prot. 91, 333–342. https://doi.org/10.1016/j.psep.2012.07.006