Stable Learning for Trustworthy Modeling in Industrial Processes With Small Samples

Liang Cao, *Graduate Student Member, IEEE*, Youqing Wang, *Senior Member, IEEE*, Yankai Cao, *Member, IEEE*, Fan Yang, Jicong Fan, *Senior Member, IEEE*, Yan Qin, Bhushan Gopaluni, and Richard D. Braatz, *Fellow, IEEE*

***Abstract*—Developing trustworthy models for industrial processes is challenging due to changing operating conditions and limited data. In this work, we present an innovative and effective framework for establishing trustworthy process models from limited data. Our approach introduces the uniform manifold approximation and projection (UMAP) algorithm to uncover the essential low-dimensional structure in real industrial data, which allows high-quality virtual sample generation (VSG) that captures the underlying process dynamics. Furthermore, we propose a new algorithm for stable learning that uses sample reweighting to effectively mitigate spurious correlations that can undermine the stability and reliability of the model. Two case studies, the Tennessee Eastman process (TEP) and a commercial fluid catalytic cracking (FCC) unit, demonstrate the effectiveness of the proposed framework. In the Tennessee Eastman benchmark, the proposed method reduces the average error by 12.5% compared with VSG alone and by 47.8% compared with the OLS baseline. In the FCC case study, it further achieves a 41.5% RMSE reduction compared with the strongest competing UMAP-augmented model. Furthermore, the proposed framework maintains robust performance under changing operating conditions. Our framework enables the development of trustworthy industrial process models from limited data, which offers a powerful method to improve operational safety and efficiency in real-world applications.**

***Index Terms*—Process modeling, stable learning, trustworthy model, uniform manifold approximation and projection (UMAP), virtual sample generation (VSG).**

I. Introduction

ADVANCED machine-learning algorithms have become dominant tools in industrial processes. However, many of these methods exhibit poor stability, especially when faced with changing operating conditions and limited data. As industrial processes become increasingly complex and sensitive to risks, the design of trustworthy models is crucial to ensure safe and efficient operation.

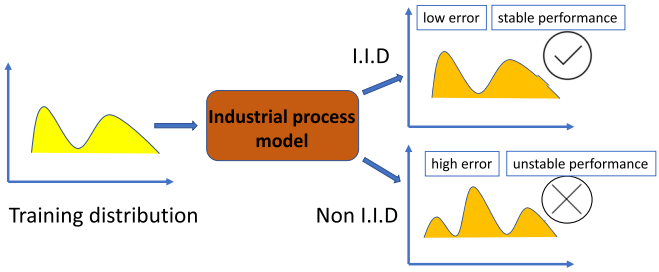

Fig. 1. Performance discrepancy of IID and Non-IID data distributions.

The primary motivation for this work is the urgent need to improve the stability and reliability of industrial process models in the face of these challenges [1], [2], [3], [4]. In this work, we use the term "stability" to refer specifically to a model's ability to maintain robust and consistent predictive performance when confronted with changing or unknown operating conditions. One major factor contributing to the stability issue is the discrepancy between the data distributions used for training and testing of the models. As shown in Fig. 1, when the test data distribution differs from the training data distribution, the model's performance may become very poor. This is common in actual industrial processes, where working conditions change frequently, and unknown working conditions may exist. The industrial production environment is also often subject to various disturbances and uncertainties. As a result, industrial process models often fail to achieve reliable predictions and face the risk of performance degradation when exposed to these changing conditions.

To address this challenge, stable learning has been proposed to guarantee prediction consistency [5]. Unlike traditional machine-learning methods that often operate under the assumption of independent and identically distributed (IID) data, stable learning does not rely on this assumption. Furthermore, stable learning distinguishes itself from transfer learning, as shown in Fig. 2. The goal of stable learning is to optimize overall performance by maximizing the average accuracy across all distributions while minimizing the variance of accuracy. On the contrary, transfer learning focuses on adapting a model trained on one distribution to perform well

Received 31 December 2025; accepted 10 May 2026. Recommended by Associate Editor D. Hoelzle. This work was supported in part by Mitacs under Grant IT49951; and in part by the National Natural Science Foundation of China under Grant 62225303, Grant 62433004, and Grant 62373361. *(Corresponding authors: Liang Cao; Bhushan Gopaluni.)*

Liang Cao and Richard D. Braatz are with the Department of Chemical Engineering, Massachusetts Institute of Technology, Cambridge, MA 02139 USA (e-mail: liangcao@mit.edu; braatz@mit.edu).

Youqing Wang is with the College of Information Science and Technology, Beijing University of Chemical Technology, Beijing 100029, China (e-mail: wang.youqing@ieee.org).

Yankai Cao and Bhushan Gopaluni are with the Department of Chemical and Biological Engineering, University of British Columbia, Vancouver, BC V6T 1Z3, Canada (e-mail: yankai.cao@ubc.ca; bhushan.gopaluni@ubc.ca).

Fan Yang is with the Department of Automation, Tsinghua University, Beijing 100084, China (e-mail: yangfan@tsinghua.edu.cn).

Jicong Fan is with the School of Data Science, The Chinese University of Hong Kong, Shenzhen 518172, China (e-mail: fanjicong@cuhk.edu.cn).

Yan Qin is with the School of Automation, Chongqing University, Chongqing 401331, China (e-mail: yan.qin@cqu.edu.cn).

Digital Object Identifier 10.1109/TCST.2026.3695112

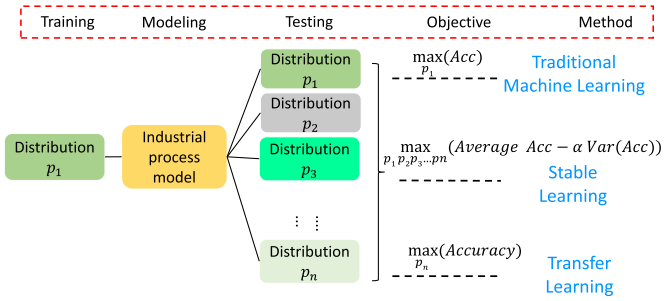

Fig. 2. Comparison of stable learning and transfer learning.

on a different distribution [6]. This fundamental difference makes stable learning particularly suitable for industrial processes where data from multiple operating conditions are available during training, but future operating conditions may differ from all previously seen conditions.

Recent advances have demonstrated the importance of model stability and reliability in industrial applications. One major focus is robust fault management, where contributions include Fravolini et al. [7] data-driven methods for robust sensor fault detection, Zhou et al. [8] system-specific fault detection and isolation methods, and Chen and Huang's development of fault-tolerant soft sensors for dynamic systems [4]. Another significant theme addresses core model stability and interpretability directly; for example, Lou et al. [9] enhanced monitoring reliability using orthonormal subspace analysis, Zhang et al. [10] introduced a causality-inspired stable long short-term memory framework for soft sensors facing distribution shifts, and Gao et al. [11] proposed identifying invariant features through latent causal representation.

In parallel with these developments, researchers have explored various algorithmic approaches to stable learning. Several efficient frameworks have emerged, including methods to find invariant features by minimizing the covariance matrix [12] and approaches using binary probabilistic classifiers [13]. StableNet [14] extends linear stable learning frameworks to nonlinear frameworks using random Fourier features.

However, the performance of these stable learning methods is generally affected by the sample size, typically performing well with big data but failing when data are limited. This limitation is particularly acute in industrial settings. While various methodologies have been proposed to address this "small sample problem" [15], [16], [17], each has inherent limitations. Transfer learning approaches [6] struggle when source and target processes differ significantly. Data augmentation techniques using generative adversarial networks (GANs) [16] or variational autoencoders (VAEs) [18] may fail to capture complex process dynamics. Incremental learning methods [19] require careful handling of concept drift, while semi-supervised strategies [20] assume consistent distributions between labeled and unlabeled samples. Consequently, there is an urgent need for innovative approaches that can support stable learning under limited sample conditions.

Virtual sample generation (VSG) [16], [21], [22] has emerged as a potential solution to address the small sample problem. Of the VSG techniques, dimensionality reduction stands out for its ability to distill complex distributions into more manageable forms, which makes it particularly suitable for generating virtual samples. Uniform manifold approximation and projection (UMAP), a state-of-the-art dimensionality reduction method, is known for its efficacy in extracting features from high-dimensional data [22].

UMAP was originally developed as a visualization tool. Its applicability to chemical and refining processes has been explored in other studies; for example, Webb and Romagnoli [23] show how UMAP could be used for real-time process monitoring. They focused on using UMAP to visualize high-dimensional data streams and detect anomalies in operational conditions. While their work showed the promise of UMAP in industrial analytics, it primarily addressed process monitoring and visualization.

On the contrary, the focus of our article is on using UMAP to tackle a different challenge: VSG under small-data scenarios. Rather than limiting UMAP to exploratory data analysis or anomaly detection, we integrate it into a broader framework for constructing trustworthy soft sensors using stable learning. To effectively recreate physically meaningful virtual samples from their low-dimensional counterparts, regression models are used in combination with UMAP to achieve precise data representation and analysis. Our work on addressing the "small sample problem" differs from existing approaches by combining VSG with stable learning to address both data scarcity and distribution-shift challenges. The contributions of this article are as follows.

1) Developing a framework that integrates UMAP dimensionality reduction with regression models to generate high-quality virtual samples, which allows accurate reconstruction of high-dimensional process data from limited samples.
2) Introducing a new algorithm for stable learning that effectively identifies invariant process features and mitigates spurious correlations, thereby significantly enhancing model stability under changing operating conditions.
3) Validating the methodology using comprehensive case studies on both the Tennessee Eastman benchmark and real-world refinery processes. The proposed approach showed substantial improvements in prediction accuracy and robustness compared to existing methods.

II. Stable Learning

A. Background

In industrial processes, model performance can fluctuate significantly when operating conditions vary, especially under limited data and non-IDD distributions. Stable learning can be defined as follows: given the target y and the input X, the objective is to find a robust model that can achieve consistent predictions on different distributions. The theoretical motivation of stable learning is closely related to the identification of invariant predictive features, which may approximate the Markov blanket of the target variable under ideal conditions. The Markov blanket is a minimal set of variables that can shield the target variable from the rest of the variables. In other

words, the Markov blanket of a variable is the set of nodes consisting of its parents, its children, and any other parents of its children. Here, we define the variables in the Markov blanket as causal variables.

However, it is important to note that stable learning methods do not guarantee exact recovery of the true Markov blanket, particularly under finite sample sizes, imperfect reweighting, and potential model misspecification [5]. Instead, stable learning aims to identify an invariant predictive feature set, features whose relationship with the target remains stable across different distributions. This set may closely approximate the Markov blanket or form a practical superset that maintains predictive stability.

Assume that s represents the causal variables of the dependent variable y, β_s denotes the regression coefficients associated with the causal variables, $g(s)$ represents the nonlinear part of the function, and ϵ is noise that is independent of the process variables, then we have the following relationship:

$$y = s^T\beta_s + g(s) + \epsilon. \tag{1}$$

In actual industrial processes, although a large number of observational variables are collected, the causal relationships between the variables are often unknown. Therefore, regression models tend to introduce noncausal parts as inputs to the model

$$y = s^T\beta_s + v^T\beta_v + g(s) + \epsilon \tag{2}$$

where v denotes a collection of noncausal features that do not have a direct causal relationship with the target variable y but may still exhibit correlations. These features, if included indiscriminately in traditional regression, can introduce spurious correlations that undermine model stability. In industrial processes, v might represent noise factors or variables that become coupled with y only under certain operating conditions, rather than through fundamental causal mechanisms. β_v represents the regression coefficients of the noncausal part, which ideally should be zero. By minimizing the sum of the squared residuals as the loss function, we obtain the following parameter estimates:

$$\hat{\beta}_v = \beta_v + \left(V^TV\right)^{-1}\left(V^Tg\right) + \left(V^TV\right)^{-1}\left(V^TS\right)\left(\beta_s - \hat{\beta}_s\right) \tag{3}$$

$$\hat{\beta}_s = \beta_s + \left(S^TS\right)^{-1}\left(S^Tg\right) + \left(S^TS\right)^{-1}\left(S^TV\right)\left(\beta_v - \hat{\beta}_v\right). \tag{4}$$

The estimated values of β_s and β_v are denoted by $\hat{\beta}_s$ and $\hat{\beta}_v$, respectively. The sample matrices of s and v are represented by S and V, respectively. Equations (3) and (4) indicate that strong correlations between causal variables S and noncausal variables V negatively impact accurate identification of model parameters. These terms can significantly deviate from zero if there is a strong correlation between S and V. As a result, the estimated coefficients $\hat{\beta}_s$ and $\hat{\beta}_v$ can be biased, which leads to incorrect and unstable model predictions. When the data distribution changes, these biases can cause significant drops in model predictive performance, which results in unstable predictive models.

To address this problem, modern algorithms for stable learning use sample reweighting techniques to adjust the data distribution and decorrelate the input variables. By reweighting the samples, the method seeks to reduce the influence of noncausal variables and better reflect the causal relationship between the input variables and the target variable. This approach can be expressed as

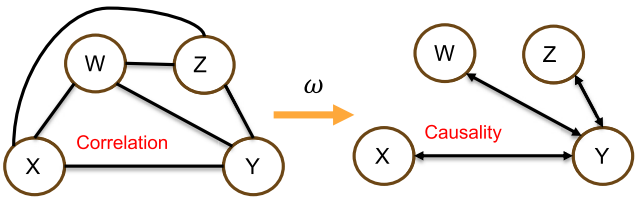

Fig. 3. Graphical representation of the algorithm for stable learning.

$$\min_{\mathbf{W},\beta} \sum_{i=1}^{n} w_i \left(y_i - \mathbf{x}_i^T\beta\right)^2 \tag{5}$$

where $\mathbf{W} = [w_1, w_2, \ldots, w_n]^T$ is the vector of sample weights, $\mathbf{x}_i$ is the ith row of matrix $\mathbf{X}$, which represents the observed values of the ith sample and β is the vector of regression coefficients. β encompasses both the causal (β_s) and noncausal (β_v) parts of the regression coefficients. By optimizing the weights $\mathbf{W}$, this method can, to some extent, weaken or eliminate the correlation between causal and noncausal variables. This drives the regression coefficients of the noncausal part $\hat{\beta}_v$ to approach zero, which improves the stability of model prediction.

Although various robust or weighted regression methods exist, they typically do not explicitly address the risk of spurious correlations between causal and noncausal variables. While these strategies can improve the tolerance to heavy-tailed distributions, they do not enforce explicit decorrelation between the features, and thus may still fail when noncausal variables are strongly correlated with the target.

On the contrary, our approach combines a decorrelation constraint to limit the effect of noncausal variables. Another key difference is the integration of VSG (using UMAP) into the stable learning framework. By synthesizing additional samples representative of the underlying manifold, we ensure that the reweighting strategy has sufficient data coverage to estimate the true causal relationship.

B. Sample Reweighting in Stable Learning

The framework of a typical algorithm for stable learning is shown in Fig. 3. The idea of stable learning is to make all inputs decorrelated by sample reweighting. To learn the sample weights w, we build on the sample reweighted decorrelation operator (SRDO) to ensure statistical independence between the features [13], and propose a new natural gradient boosting (NGBoost)-based scheme to learn these weights.

Previous stable learning approaches typically employed traditional classification methods such as logistic regression and decision trees for weight learning [12], [13]. While these methods can identify basic decorrelation patterns, they have limitations when applied to industrial process data characterized by limited samples, non-Gaussian distributions, and complex nonlinear relationships. We propose to use NGBoost [24] for sample weight learning. NGBoost offers unique

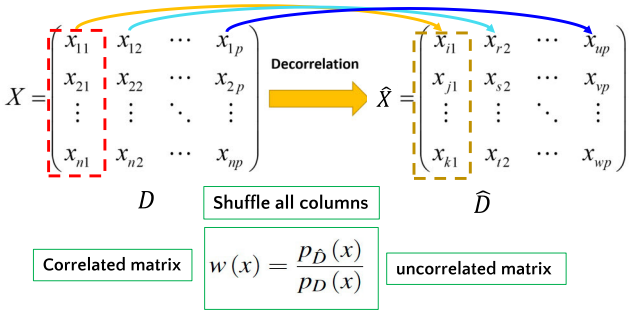

Fig. 4. Graphical representation of the SRDO.

advantages that make it particularly suitable for our stable learning framework. Unlike traditional gradient boosting methods, such as extreme gradient boosting (XGBoost), which focus solely on point predictions, NGBoost provides probabilistic predictions by modeling the complete conditional distribution of the target variable.

Fig. 4 is an example of SRDO with NGBoost. First, we use matrix X to generate a column-decorrelated $\hat{X}$ by performing random resampling columnwise, where $i, j, k, r, s, t, u, v, w$ are drawn from $1, 2, \ldots, n$ at random. Random resampling can break down the joint distribution D of X into p independent marginal distributions $\hat{D}$ of $\hat{X}$. Since $\hat{X}$ has completely independent columns, which means that we can transfer the original X to the decorrelated $\hat{X}$ by SRDO.

In particular, we designate the samples in $\hat{X}$ as positive samples ($Z = 1$) because they are completely decorrelated, while the samples in X are set as negative samples ($Z = 0$) since they represent the original data. We then fit a binary probabilistic classifier to learn the classification and obtain the weights, which represent the degree of decorrelation. The decorrelated weight can be given as follows:

$$w(x) = \frac{p_{\hat{D}}(x)}{p_D(x)} = \frac{p(Z = 1 \mid x)}{p(Z = 0 \mid x)} \tag{6}$$

where $p(Z = 1|x)$ is the estimated probability that the sample x is drawn from $\hat{D}$ and $p(Z = 0|x)$ is the estimated probability of sample x being drawn from D.

C. Theoretical Explanation of Stable Learning

A key theoretical premise of stable learning is that the true causal features remain invariant under different environments, whereas noncausal features tend to vary. Stable learning essentially seeks to discover invariant features that remain predictive across different environments or distributions. By learning a reweighting function on the training data, stable learning highlights the key process relationships, which ensure predictive accuracy even when the test distribution diverges from the training distribution. Under suitable assumptions, stable learning can provide theoretical guarantees for regression and classification tasks with commonly used loss functions, such as mean-squared loss and binary cross-entropy loss [5]. The optimization objective can be expressed as finding model parameters θ that minimize the expected loss across all potential test distributions

$$\min_{\theta} E_{P_{test}}\left[L(Y, f_\theta(X))\right] \tag{7}$$

where $L(\cdot)$ is the loss function, $f_\theta(\cdot)$ is the model's prediction function, and P_{test} denotes the unknown test distribution. Unlike classical robust regression, which typically accounts for outliers within a single distribution, stable learning addresses broader challenges arising from distribution shifts across multiple environments. While the theoretical foundation of stable learning is motivated by the concept of Markov blankets, in practice, stable learning identifies a set of invariant predictive features S that may approximate or form a superset of the true Markov blanket. Under ideal conditions, including infinite samples, perfect weight learning, and correct model specification, the identified feature set S satisfies

$$E[Y|S] = E[Y|X]. \tag{8}$$

Equation (8) formalizes the key property that the selected features S contain all necessary information for prediction, which makes additional variables redundant. The theoretical guarantees of stable learning rely on two critical assumptions. First, there must exist a subset of features that maintains stable relationships with the target variable across different distributions. Second, the training data must provide adequate coverage of the feature space to learn meaningful weights.

However, in practice, these assumptions may not hold, particularly in industrial settings where the available data is often limited, and the underlying data distribution is complex. The performance of stable learning can be significantly influenced by the quality and quantity of available data, as well as the effectiveness of the sample reweighting scheme.

To address these limitations and bridge the gap between theory and practice, we propose the use of VSG techniques. By generating additional samples that follow the underlying data distribution, we can effectively augment the available dataset and improve the performance of stable learning, even in the presence of limited data.

In this work, we focus on the UMAP algorithm as a powerful tool for VSG. In Sections III and IV, we will delve into the details of the UMAP algorithm and its application in VSG. We will also discuss how the generated virtual samples can be seamlessly integrated into the stable learning framework to enhance its performance and applicability in industrial settings.

III. Virtual Sample Generation

A critical limitation of stable learning algorithms is their requirement for substantial training data to accurately estimate sample weights for decorrelation. As demonstrated in Section II, the SRDO must distinguish between correlated and decorrelated feature distributions. This creates a fundamental paradox: stable learning is most needed in small-sample scenarios, yet performs poorly precisely in these settings.

To address this challenge, we propose a novel framework that integrates dimensionality reduction with regression-based reconstruction for high-quality VSG. Our key innovation lies

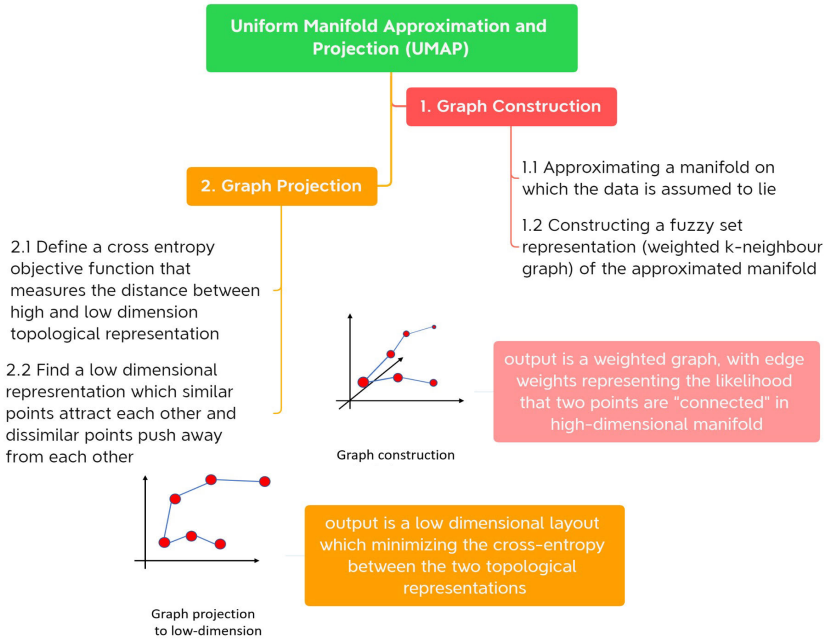

Fig. 5. Main flowchart of UMAP algorithm.

not in applying UMAP as a standalone tool, but in developing a systematic pipeline that first uses UMAP to uncover the intrinsic low-dimensional manifold structure of industrial process data. Within this manifold space, we generate virtual samples through informed interpolation. These samples are then accurately reconstructed to high dimensions, ensuring the preservation of both local geometric relationships and global process dynamics.

A. Overview of the UMAP Algorithm

The VSG task can be defined as follows: given the original data with input X and label y, the task is to generate virtual sample input $\tilde{X}$ and label $\tilde{y}$ according to the distribution of the original data.

UMAP seeks to learn a low-dimensional representation of high-dimensional data while preserving its intrinsic structure. Fig. 5 shows the flowchart of the UMAP algorithm. The algorithm consists of two steps: constructing a weighted k-neighbor graph in a high-dimensional space and optimizing a low-dimensional layout to obtain a faithful representation.

1) Step 1: Graph Construction: In the first step, UMAP constructs a weighted k-nearest neighbor graph H in the high-dimensional space. Given an input dataset X with a distance metric d and a hyperparameter k, the algorithm identifies the k nearest neighbors set $\{x_{i_1}, \ldots, x_{i_k}\}$ for each data point x_i. The parameter k is chosen to balance the preservation of local and global data structures. For each data point x_i, the distance to its nearest neighbor is defined as ρ_i

$$\rho_i = \min\left\{ d\left(x_i, x_{i_j}\right) \middle| \, 1 \le j \le k \right\}. \tag{9}$$

This distance ρ_i varies for each point to ensure that local connectivity within the manifold is maintained. Next, to map distances from high-dimensional space to low-dimensional space, UMAP introduces a normalization factor σ_i. This factor is determined so that

$$\sum_{j=1}^{k} \exp\left(\frac{-\max\left(0, d\left(x_i, x_{i_j}\right) - \rho_i\right)}{\sigma_i} \right) = \log_2(k). \tag{10}$$

The weighted k-nearest neighbor graph is represented by a matrix $H = (V, E, w)$, where V denotes the set of vertices, E represents the set of directed edges between each data point x_i and its k nearest neighbors, and w is the weight function defined as

$$w\left(x_i, x_{i_j}\right) = \exp\left(\frac{-\max\left(0, d\left(x_i, x_{i_j}\right) - \rho_i\right)}{\sigma_i} \right). \tag{11}$$

To ensure symmetry, since the directed edges' probabilities between x_i and x_{ij} might not be equal, we define $w_h(i, j)$ as the probability that at least one of the two directed edges exists

$$w_h(i, j) = w\left(x_i, x_{i_j}\right) + w\left(x_j, x_{j_i}\right) - w\left(x_i, x_{i_j}\right) w\left(x_j, x_{j_i}\right). \tag{12}$$

The resulting UMAP graph H is therefore an undirected weighted graph, with the adjacency matrix elements given by $w_h(i, j)$. This graph construction step sets the stage for the subsequent dimensionality reduction process.

2) Step 2: Graph Projection: In the second step, UMAP projects the graph H from high dimensions to a graph L in low dimensions. The goal is to find low-dimensional positions l_i for each data point x_i such that the low-dimensional graph L closely approximates the high-dimensional graph H. UMAP models the probability of an edge existing between two points in the low-dimensional space using

$$w_l(i, j) = \left(1 + a \left\| l_i - l_j \right\|^2\right)^{-b} \tag{13}$$

where a and b are UMAP hyperparameters. UMAP optimizes the low-dimensional layout by minimizing the cross-entropy between the high-dimensional and low-dimensional edge probabilities

$$C = \sum_i \sum_j w_h(i,j) \log\left(\frac{w_h(i,j)}{w_l(i,j)}\right) + (1 - w_h(i,j)) \log\left(\frac{1 - w_h(i,j)}{1 - w_l(i,j)}\right). \quad (14)$$

Overall, UMAP constructs a high-dimensional graph representation of the data and optimizes a low-dimensional graph to be as structurally similar as possible. By mapping high-dimensional data to low dimensions, UMAP effectively extracts rich information.

B. Generation of Virtual Samples Based on Regression

Although UMAP effectively reduces the dimensionality of data and reveals its intrinsic low-dimensional structure, the mapping between the low-dimensional and high-dimensional spaces is not straightforward. The distances in the low-dimensional space do not directly reflect the distances in the high-dimensional space because of the warping effect of UMAP. To accurately generate high-dimensional virtual samples from their low-dimensional representations, we use regression models.

First, we project the high-dimensional data onto a low-dimensional space using UMAP. The dimensionality of this space is treated as a hyperparameter that is optimized based on the specific dataset characteristics. This low-dimensional space captures the essential structure of the original data. Next, we establish two types of regression models: input regression and output regression. For input regression, we create models using the low-dimensional UMAP co-ordinates (l) as inputs and the high-dimensional original data (X) as outputs. These models allow us to predict high-dimensional data points ($\tilde{X}$) from new low-dimensional points ($\tilde{l}$) generated using k-nearest neighbors (KNNs) interpolation. For output regression, we build a model that uses the original high-dimensional data (X) as the input and the target variable (y) as the output. This model allows us to predict the target variable ($\tilde{y}$) for newly generated high-dimensional virtual samples ($\tilde{X}$). In this work, we use random forest regression for both input and output regression due to its robustness and the ability to capture complex relationships [25]. Using this two-step regression process, we ensure that the virtual samples are representative of the original data distribution and maintain the intricate relationships present in the high-dimensional space.

C. Theoretical Explanation of Virtual Samples

The effectiveness of VSG in improving model performance can be understood through the lens of the "No Free Lunch" theorem and the concept of prior knowledge incorporation. The "No Free Lunch" theorem states that, in the absence of any prior knowledge about the problem at hand, no single model or algorithm can consistently outperform others across all possible datasets [26]. VSG is a way to incorporate prior knowledge into the learning process. By generating virtual samples that adhere to the underlying data distribution, we essentially inject domain-specific information into the model. This prior knowledge acts as a regularizer that constrains the model's hypothesis space and guides it toward more plausible and generalizable solutions.

Mathematically, the incorporation of prior knowledge through VSG can be formulated as a regularization term in the model's objective function. Let $L(X, y; \theta)$ denote the loss function of a model with parameters θ on a dataset X with corresponding labels y. The regularized objective function with virtual samples can be expressed as

$$\min_\theta L(X, y; \theta) + \lambda R\left(\tilde{X}, \tilde{y}; \theta\right) \quad (15)$$

where $\tilde{X}$ and $\tilde{y}$ represent the virtual samples and their corresponding labels, $R(\tilde{X}, \tilde{y}; \theta)$ is the regularization term that captures the prior knowledge encoded by the virtual samples, and λ is a hyperparameter that controls the strength of the regularization.

By minimizing this regularized objective function, the model learns to fit the original data while simultaneously conforming to the prior knowledge embedded in the virtual samples. This regularization effect helps the model to generalize better to unseen data.

IV. Trustworthy Modeling With Small Samples

The challenge of building trustworthy models from limited data arises due to the inherent difficulty in decorrelating all variables with a finite sample size. As established in Section II, stable learning relies on sample reweighting to achieve statistical independence among input features, thereby isolating causal variables from spurious correlations. The framework for trustworthy modeling with small samples is shown in Fig. 6 and Algorithm 1. The framework consists of three integrated phases that systematically address both data scarcity and distribution-shift challenges.

In Phase 1, the UMAP algorithm projects the original high-dimensional data onto a low-dimensional manifold while preserving the intrinsic geometric structure. Virtual samples are then generated through KNN interpolation in this low-dimensional space, followed by reconstruction to the original feature space using trained regression models. This approach ensures that the generated samples faithfully represent the underlying data distribution rather than introducing artificial patterns. The quality of virtual samples is validated by computing the Kullback–Leibler divergence between the distributions of original and virtual samples.

Phase 2 uses the proposed NGBoost-based SRDO to learn appropriate weights for the augmented dataset. By constructing a binary classification problem, the NGBoost classifier learns to estimate the degree of correlation present in each sample. The resulting weights better represent the causal relationships. This probabilistic approach to weight learning offers advantages over traditional methods by providing calibrated uncertainty estimates and robust convergence behavior.

In Phase 3, the learned weights are incorporated into a weighted least squares regression framework to obtain the final stable model. By combining the expanded sample coverage

Algorithm 1 Trustworthy Modeling With VSG and Stable Learning

Require: Training dataset $\mathcal{D} = \{(X_i, y_i)\}_{i=1}^n$, number of virtual samples N_v, UMAP dimension d, number of neighbors k
Ensure: Stable predictive model f^* with parameters β^*

// Phase 1: UMAP-Based Virtual Sample Generation
Construct k-nearest neighbor graph H in original input space
for each data point x_i **do**
  Compute local connectivity ρ_i and normalization factor σ_i
  Calculate edge weights: $w(x_i, x_{i_j}) = \exp\left(\frac{-\max(0, d(x_i, x_{i_j}) - \rho_i)}{\sigma_i}\right)$
end for
Symmetrize: $w_h(i,j) = w(x_i, x_{i_j}) + w(x_j, x_{j_i}) - w(x_i, x_{i_j}) \cdot w(x_j, x_{j_i})$
Optimize low-dimensional embedding $L = \{l_i\}_{i=1}^n \in \mathbb{R}^{n \times d}$ by minimizing cross-entropy
Train input regression model $f_X : \mathbb{R}^d \to \mathbb{R}^p$ using (L, X)
Train output regression model $f_y : \mathbb{R}^p \to \mathbb{R}$ using (X, y)
for $j = 1$ to N_v **do**
  Generate $\tilde{l}_j$ via KNN interpolation in low-dimensional space
  Reconstruct: $\tilde{X}_j = f_X(\tilde{l}_j)$, $\tilde{y}_j = f_y(\tilde{X}_j)$
end for
Form augmented dataset: $\mathcal{D}_{aug} = \mathcal{D} \cup \{(\tilde{X}_j, \tilde{y}_j)\}_{j=1}^{N_v}$
// Phase 2: NGBoost-Based Sample Reweighting
Generate decorrelated reference $\hat{X}$ via column-wise random permutation of X^{aug}
Construct binary dataset: positive samples $(\hat{X}, Z = 1)$, negative samples $(X^{aug}, Z = 0)$
Train NGBoost classifier g to predict $P(Z = 1|x)$
for $i = 1$ to $n + N_v$ **do**
  Compute decorrelation weight: $w_i = \frac{P(Z=1|X_i^{aug})}{P(Z=0|X_i^{aug})}$
end for
Normalize weights: $W \leftarrow W / \left(\sum_i w_i\right) \cdot (n + N_v)$
// Phase 3: Weighted Stable Regression
Solve: $\beta^* = \arg\min_\beta \sum_{i=1}^{n+N_v} w_i (y_i^{aug} - (X_i^{aug})^T \beta)^2$
Construct stable model: $f^*(x) = x^T \beta^*$
return: Stable model f^* with parameters β^*

from virtual sample generation with the decorrelation capability of stable learning, the UMAP-SL framework achieves robust predictive performance. This integrated approach bridges the gap between the theoretical requirements of stable learning and the practical constraints of industrial applications where data collection is limited.

V. Case Studies

In this section, we present two case studies to demonstrate the effectiveness of the proposed methodology. The experimental settings deliberately employ limited training data to reflect realistic industrial scenarios where our method provides the greatest value.

The first case study demonstrates the performance of stable learning using the Tennessee Eastman process (TEP). The second case study applies the methodology to the fluid catalytic cracking (FCC) process from a commercial refinery. To address concerns about generalizability, we conduct comprehensive ablation studies examining the effect of virtual sample size and compare performance across multiple regression methods. The consistency of these findings across two industrial processes underscores the practical utility and generalizability of our framework.

A. Tennessee Eastman Process

The TEP is a widely recognized benchmark for process control and monitoring [27]. The process simulates a realistic chemical plant with five major unit operations: a reactor, condenser, compressor, separator, and stripper. The benchmark includes one normal operating condition and 21 predefined fault conditions that simulate various process disturbances such as step changes, random variations, and slow drifts in process parameters. In this case study, we simulate a scenario with restricted data availability and fluctuating operating conditions. We selected 33 variables as input variables. The target variable for prediction is the concentration of component C in the purge gas. We limited the training data size to 300 samples. This constraint allows us to test the effectiveness of the proposed method under data scarcity. The KL divergence threshold is established at 0.5.

To comprehensively evaluate the stability and generalization performance of the proposed method across different operating conditions, we introduce four aggregated metrics: average error (AE), standard error (SE) of root-mean-squared error, average R^2 (AR^2), and SE of R^2 (SR^2). Let M denote the total number of test scenarios (including normal and various fault conditions). Let $RMSE_k$ and R_k^2 represent the root-mean-square error and the coefficient of determination calculated for the kth test scenario, respectively. The metrics are defined as follows:

$$AE = \frac{1}{M}\sum_{k=1}^{M} RMSE_k \quad AR^2 = \frac{1}{M}\sum_{k=1}^{M} R_k^2$$
$$SE = \sqrt{\frac{1}{M(M-1)}\sum_{k=1}^{M}(RMSE_k - AE)^2}$$
$$SR^2 = \sqrt{\frac{1}{M(M-1)}\sum_{k=1}^{M}\left(R_k^2 - AR^2\right)^2}. \tag{16}$$

1) Comparison of Virtual Sample Generation Methods: We evaluated the quality of virtual samples generated by different dimensionality reduction techniques. UMAP was compared with t-distributed stochastic neighbor embedding (t-SNE) [28] and VAE [18], as these methods also seek to reduce dimensionality.

To ensure fair comparison, we used a consistent strategy for selecting latent dimensionality and key hyperparameters. For UMAP, we evaluated different numbers of components (2–10) using the validation dataset. For t-SNE, we tuned the perplexity parameter using a grid search over the range of 5–50. The VAE architecture was optimized by testing various encoder/decoder configurations with different layer

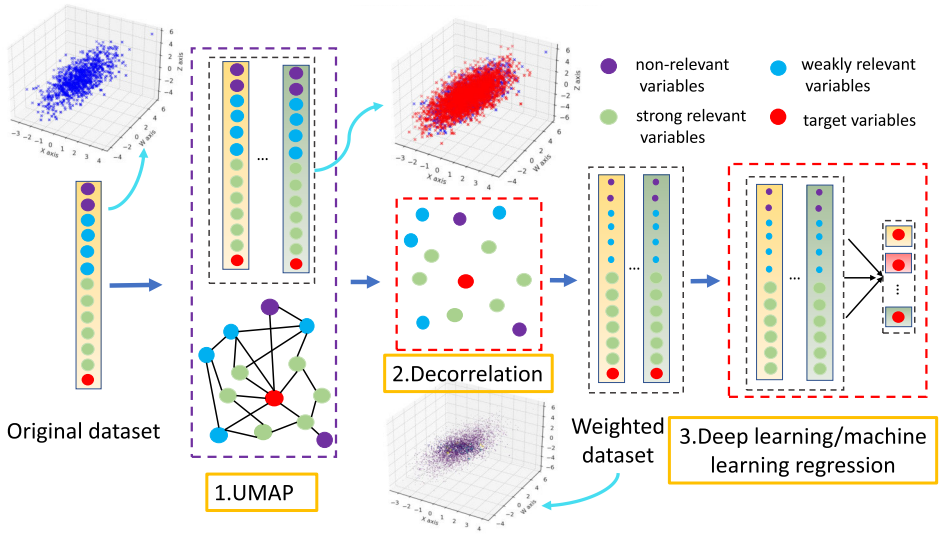

Fig. 6. Framework of trustworthy modeling with small samples.

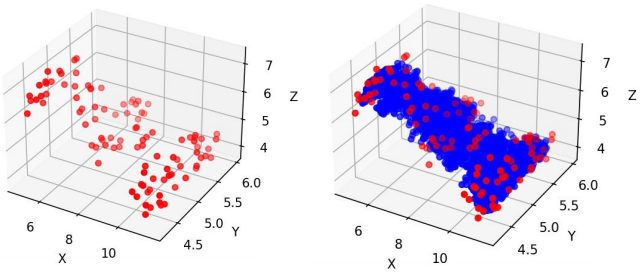

Fig. 7. Low-dimension representation of original data (red dots) and virtual generated data (blue dots).

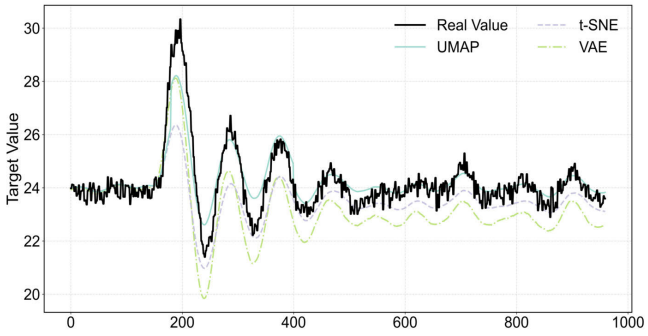

Fig. 8. Virtual sample generation method comparison for fault case 1 with 2000 virtual samples.

TABLE I

PERFORMANCE COMPARISON OF DIMENSIONALITY REDUCTION METHODS

Method	Samples	AE	SE	AR^2	SR^2
UMAP	500	2.1089	1.1644	**0.4949**	0.0470
	2000	**1.9697**	**1.0263**	**0.5324**	**0.0448**
	5000	**1.5886**	**0.7662**	**0.5595**	**0.0441**
t-SNE	500	**1.7300**	**0.7533**	0.2151	0.0361
	2000	2.8844	1.4834	0.4299	0.0449
	5000	2.9614	1.7385	0.5173	0.0445
VAE	500	2.2643	1.0022	0.2973	**0.0360**
	2000	2.4021	1.3373	0.5162	0.0544
	5000	2.7690	1.5034	0.4019	0.0453

TABLE II

ABLATION STUDY: PERFORMANCE COMPARISON OF DIFFERENT METHODS

Method	Samples	AE	SE	AR^2	SR^2
Baseline	/	2.6607	1.5288	0.4541	0.0494
SL	/	3.5203	2.0276	0.3504	0.0412
UMAP	500	2.1089	1.1644	0.4949	0.0470
	2000	1.9697	1.0263	0.5324	0.0448
	5000	1.5886	0.7662	0.5595	0.0441
UMAP-SL	500	2.8931	1.9125	0.3857	**0.0359**
	2000	5.1160	2.9153	0.2697	0.0396
	5000	**1.3900**	**0.6513**	**0.5669**	0.0453

sizes and depths. Default settings were used for less critical hyperparameters.

Fig. 7 shows the UMAP-based virtual sample generation in 3-D space. The original 300 training samples are projected from the high-dimensional input space to a 3-D representation using UMAP. Virtual samples are then generated via KNN interpolation in this low-dimensional space. The left panel shows the original data distribution, while the right panel shows the augmented dataset with virtual samples. The virtual samples closely follow the manifold structure of the original data, which indicates that UMAP effectively preserves the intrinsic geometric relationships during dimensionality reduction.

Table I summarizes the performance comparison for different virtual sample sizes ($N_v = 500, 2000, 5000$). Bold values indicate the best performance. The results reveal that the optimal dimensionality reduction method depends on the number of virtual samples generated. With only 500 virtual

TABLE III
PERFORMANCE COMPARISON OF DIFFERENT REGRESSION METHODS AND VIRTUAL SAMPLE GENERATION METHODS

Method	Original Data		UMAP		t-SNE		VAE	
	R^2	RMSE	R^2	RMSE	R^2	RMSE	R^2	RMSE
Lasso	0.661	1274	0.698	1184	0.671	1241	0.682	1209
Least Squares	0.791	1007	0.836	854	0.813	985	0.817	981
Elastic Net	0.804	985	0.855	799	0.831	896	0.841	872
Huber	0.810	945	0.845	813	0.816	936	0.783	1081
LSTM(3-layer)	0.826	858	0.861	785	0.825	854	0.856	809
LightGBM	0.839	847	0.902	718	0.886	750	0.893	740
SVR	0.847	832	0.894	736	0.889	768	0.891	739
PLS	0.849	836	0.892	739	0.887	771	0.894	734
Decision tree	0.851	819	0.917	687	0.904	700	0.901	696
XGBoost	0.860	801	0.925	670	0.912	688	0.919	682
SL	**0.955**	**494**	**0.983**	**392**	**0.971**	**425**	**0.962**	**451**

samples, t-SNE achieves the lowest AE of 1.7300, while UMAP obtains the highest AR^2 of 0.4949. As the number of virtual samples increases to 2000 and 5000, UMAP consistently outperforms both t-SNE and VAE across all metrics. With 5000 virtual samples, UMAP achieves the lowest AE of 1.5886, the lowest SE of 0.7662, and the highest AR^2 of 0.5595. On the contrary, t-SNE yields an AE of 2.9614, and VAE yields an AE of 2.7690 with the same sample size. These results demonstrate that UMAP better preserves the intrinsic manifold structure of the process data, particularly when sufficient virtual samples are generated to adequately represent the underlying distribution.

Fig. 8 shows a direct comparison of model predictions for fault case 1 using 2000 virtual samples. The UMAP-based model tracks the true values most accurately, particularly in regions with rapid value changes. The t-SNE and VAE-based models show larger deviations from the true values, especially during transient periods.

2) Ablation Study: We performed a comprehensive ablation study to validate the contributions of each component in our proposed framework. Four scenarios were designed to isolate the effects of stable learning and virtual sample generation. The baseline scenario uses ordinary least squares regression without any enhancement. The SL scenario applies stable learning directly on the original small dataset of 300 samples. The UMAP scenario employs UMAP-based virtual sample generation without incorporating stable learning. The proposed UMAP-SL scenario combines both stable learning and UMAP-based virtual sample generation.

Table II shows the performance metrics for all scenarios. The baseline method yields an AE of 2.6607 and an AR^2 of 0.4541. Applying SL actually degrades performance, increasing the AE to 3.5203 and decreasing the AR^2 to 0.3504. This confirms that stable learning struggles to estimate accurate decorrelation weights when sample sizes are insufficient. The limited training data prevents the algorithm from learning meaningful sample weights for effective decorrelation.

The UMAP scenario demonstrates the value of virtual sample generation. With 5000 virtual samples, this approach reduces the AE to 1.5886 and improves the AR^2 to 0.5595. The performance improves progressively as the number of virtual samples increases from 500 to 5000, indicating that larger augmented datasets provide better coverage of the feature space.

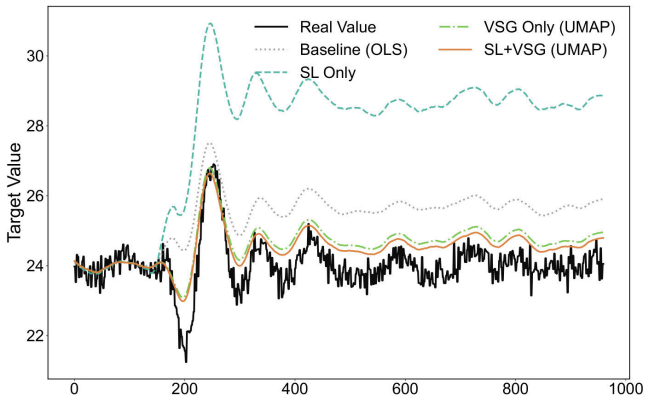

Fig. 9. Ablation study for fault case 5 with 5000 virtual samples.

The proposed UMAP-SL method achieves the best overall performance when sufficient virtual samples are generated. With 5000 virtual samples, UMAP-SL reaches the lowest AE of 1.3900 and the highest AR^2 of 0.5669. The SE decreases to 0.6513, indicating more consistent predictions across different operating conditions. However, an interesting observation is that UMAP-SL with fewer virtual samples (500 or 2000) performs worse than UMAP. This occurs because stable learning requires sufficient samples to accurately estimate the decorrelation weights.

Fig. 9 shows the prediction results for fault case 5 with 5000 virtual samples. The baseline and SL predictions show significant deviations from the true values, particularly during process transitions. The UMAP approach improves tracking accuracy considerably. The UMAP-SL method achieves the closest alignment with the true values throughout the entire test period. This confirms that combining high-quality virtual samples with stable learning effectively mitigates the small-sample problem and ensures robust predictions under changing operating conditions.

B. FCC Process

In the second case study, we present a comprehensive analysis of the methods implemented for the FCC process CO_2

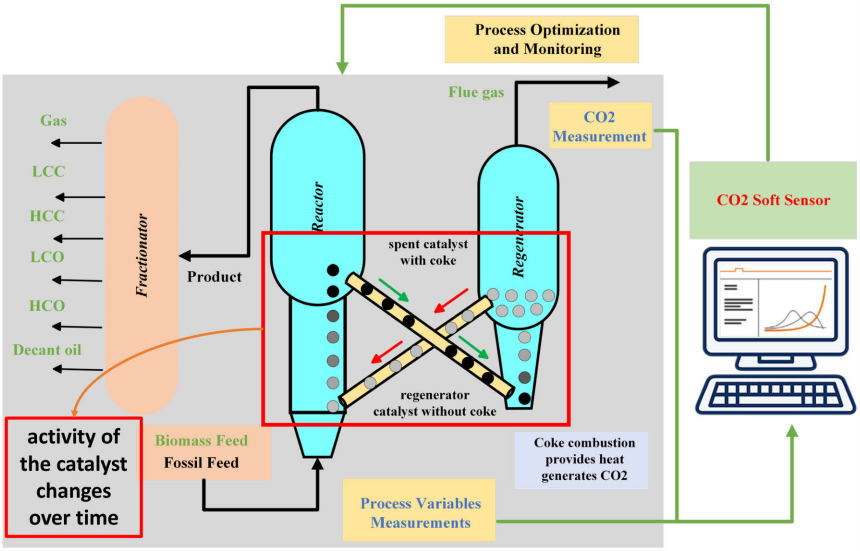

Fig. 10. Flowchart of FCC process.

online monitoring at the Parkland Burnaby Refinery in Canada [29]. The FCC unit is a critical component that converts heavy hydrocarbon fractions into lighter, more valuable products such as gasoline. In the FCC process, shown in Fig. 10, the catalyst activity changes over time due to coke deposition and regeneration cycles, which leads to variations in the data distribution. These variations violate the IID assumption, which poses challenges for traditional FCC CO_2 online monitoring.

We selected 1000 samples. Each sample contains 24 process variables and the target variable CO_2 emission. We used 300 samples for training and 700 samples for testing. The validation set, comprising 20% of the training data, was used for hyperparameter optimization. To evaluate the performance of the proposed algorithm for stable learning, we compared it with several machine-learning methods. We selected a comprehensive set of regression methods, including advanced algorithms like LSTM, lightgbm, XGBoost, and support vector regression (SVR), alongside traditional methods like lasso, elastic net, Huber, and decision trees [3], [29], [30].

We also compared UMAP, t-SNE, and VAE under the stable learning framework. After determining the key hyperparameters, we generated 5000 virtual samples using UMAP, t-SNE, and VAE, and observed performance improvements across all models. Table III summarizes the precision of the prediction in terms of R^2 and RMSE for each method. Compared to traditional methods, our UMAP-SL approach shows substantial improvements, with RMSE reductions of 66.9%, 54.1%, and 50.9%. Even when compared with complex advanced ensemble methods like XGBoost, which achieved the second-best performance, UMAP-SL still demonstrates significant advantages with a 41.5% lower RMSE and 6.3% higher R^2.

To further show the performance of stable learning, we compared its CO_2 prediction results with the XGBoost model. Fig. 11 shows the CO_2 prediction using the XGBoost model (top) and the prediction using the stable learning method

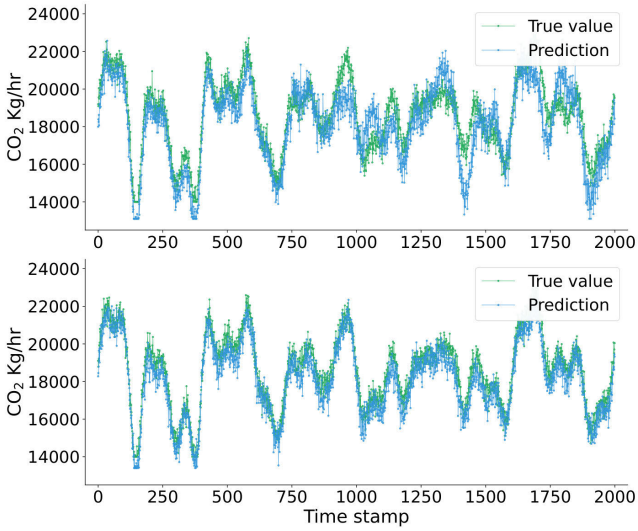

Fig. 11. CO_2 prediction of XGBoost (top) and stable learning (bottom) with changing catalyst activity.

(bottom). The green dots represent the true values, while the blue dots show the predicted values. For the XGBoost model, the model predictions deviate from the actual values, particularly in the second half of the data, where the predictions are consistently higher than the true values. This discrepancy suggests that the model does not fully capture the underlying dynamics or changes in the process over time. The stable learning model shows significantly better alignment with the true values, which shows its robustness and accuracy despite the changing catalyst activity.

VI. Conclusion

In this study, we addressed the challenge of building trustworthy models for industrial processes when only limited

data is available, and the test data distribution is unknown. To tackle this problem, we proposed a new methodology, UMAP-SL, to combine virtual sample generation using the UMAP algorithm with stable learning techniques. This model addresses the assumption in stable learning that a large number of samples are required to learn a stable model with UMAP-based virtual sample generation. In addition, we introduced a new NGBoost-based stable learning to learn sample weights for trustworthy modeling of industrial data. We validated the effectiveness of UMAP-SL in the TEP and an FCC unit. In the Tennessee Eastman benchmark, UMAP-SL reduced the average error by 12.5% compared with UMAP-based virtual sample generation alone and by 47.8% compared with the OLS baseline. In the FCC case study, UMAP-SL achieved the lowest RMSE and reduced the error by 41.5% compared with the UMAP-augmented regression baseline. These results demonstrate the ability of the proposed framework to improve prediction accuracy and robustness under small-sample and distribution-shift conditions. Future research will focus on improving the efficiency of virtual sample generation and exploring applications of UMAP-SL in other industrial processes.

REFERENCES

[1] X. Ma, Y. Si, Y. Qin, and Y. Wang, "Fault detection for dynamic processes based on recursive innovational component statistical analysis," *IEEE Trans. Autom. Sci. Eng.*, vol. 20, no. 1, pp. 310–319, Jan. 2023.

[2] L. Wu and R. D. Braatz, "A direct optimization algorithm for input-constrained MPC," *IEEE Trans. Autom. Control*, vol. 70, no. 2, pp. 1366–1373, Feb. 2025.

[3] L. Cao, F. Yu, F. Yang, Y. Cao, and R. B. Gopaluni, "Data-driven dynamic inferential sensors based on causality analysis," *Control Eng. Pract.*, vol. 104, Nov. 2020, Art. no. 104626.

[4] H. Chen and B. Huang, "Fault-tolerant soft sensors for dynamic systems," *IEEE Trans. Control Syst. Technol.*, vol. 31, no. 6, pp. 2805–2818, Nov. 2023.

[5] R. Xu, X. Zhang, Z. Shen, T. Zhang, and P. Cui, "A theoretical analysis on independence-driven importance weighting for covariate-shift generalization," in *Proc. 39th Int. Conf. Mach. Learn.*, 2021, pp. 24803–24829.

[6] S. J. Pan and Q. Yang, "A survey on transfer learning," *IEEE Trans. Knowl. Data Eng.*, vol. 22, no. 10, pp. 1345–1359, Oct. 2010.

[7] M. L. Fravolini, G. Del Core, U. Papa, P. Valigi, and M. R. Napolitano, "Data-driven schemes for robust fault detection of air data system sensors," *IEEE Trans. Control Syst. Technol.*, vol. 27, no. 1, pp. 234–248, Jan. 2019.

[8] D. Zhou, H. Ji, X. He, and J. Shang, "Fault detection and isolation of the brake cylinder system for electric multiple units," *IEEE Trans. Control Syst. Technol.*, vol. 26, no. 5, pp. 1744–1757, Sep. 2018.

[9] Z. Lou, Y. Wang, Y. Si, and S. Lu, "A novel multivariate statistical process monitoring algorithm: Orthonormal subspace analysis," *Automatica*, vol. 138, Apr. 2022, Art. no. 110148.

[10] X. Zhang, C. Song, B. Huang, and J. Zhao, "Bayesian-based causal structure inference with a domain knowledge prior for stable and interpretable soft sensing," *IEEE Trans. Cybern.*, vol. 54, no. 10, pp. 6081–6094, Oct. 2024.

[11] X. Gao, Y. Huang, and Y. A. W. Shardt, "Discovering latent causal variables using a trade-off between compression and causality," *IFAC-PapersOnLine*, vol. 58, no. 14, pp. 1–6, 2024.

[12] K. Kuang, P. Cui, S. Athey, R. Xiong, and B. Li, "Stable prediction across unknown environments," in *Proc. 24th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, Jul. 2018, pp. 1617–1626.

[13] Z. Shen, P. Cui, T. Zhang, and K. Kunag, "Stable learning via sample reweighting," in *Proc. AAAI Conf. Artif. Intell.*, 2020, vol. 34, no. 4, pp. 5692–5699.

[14] X. Zhang, P. Cui, R. Xu, L. Zhou, Y. He, and Z. Shen, "Deep stable learning for out-of-distribution generalization," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2021, pp. 5368–5378.

[15] M. Wasikowski and X.-W. Chen, "Combating the small sample class imbalance problem using feature selection," *IEEE Trans. Knowl. Data Eng.*, vol. 22, no. 10, pp. 1388–1400, Oct. 2010.

[16] Q. Zhu, K.-R. Hou, Z.-S. Chen, Z. Gao, Y. Xu, and Y. He, "Novel virtual sample generation using conditional GAN for developing soft sensor with small data," *Eng. Appl. Artif. Intell.*, vol. 106, Nov. 2021, Art. no. 104497.

[17] Y.-L. He, Q. Hua, Q. Zhu, and S. Lu, "Enhanced virtual sample generation based on manifold features: Applications to developing soft sensor using small data," *ISA Trans.*, vol. 126, pp. 398–406, Jul. 2022.

[18] Z. Wan, Y. Zhang, and H. He, "Variational autoencoder based synthetic data generation for imbalanced learning," in *Proc. IEEE Symp. Ser. Comput. Intell. (SSCI)*, Nov. 2017, pp. 1–7.

[19] Z. Yang, J. Long, Y. Zi, S. Zhang, and C. Li, "Incremental novelty identification from initially one-class learning to unknown abnormality classification," *IEEE Trans. Ind. Electron.*, vol. 69, no. 7, pp. 7394–7404, Jul. 2022.

[20] L. Yao and Z. Ge, "Deep learning of semisupervised process data with hierarchical extreme learning machine and soft sensor application," *IEEE Trans. Ind. Electron.*, vol. 65, no. 2, pp. 1490–1498, Feb. 2018.

[21] A. Kulesa, M. Krzywinski, P. Blainey, and N. Altman, "Sampling distributions and the bootstrap," *Nature Methods*, vol. 12, no. 6, pp. 477–478, Jun. 2015.

[22] J. Healy and L. McInnes, "Uniform manifold approximation and projection," *Nature Rev. Methods Primers*, vol. 4, no. 1, p. 82, 2024.

[23] Z. Webb and J. A. Romagnoli, "Real-time chemical process monitoring with UMAP," in *Computer Aided Chemical Engineering*. Amsterdam, The Netherlands: Elsevier, 2021, pp. 2077–2082.

[24] T. Duan et al., "NGBoost: Natural gradient boosting for probabilistic prediction," in *Proc. 37th Int. Conf. Mach. Learn.*, 2019, pp. 2690–2700.

[25] M. S. Hossain Lipu et al., "Real-time state of charge estimation of lithium-ion batteries using optimized random forest regression algorithm," *IEEE Trans. Intell. Vehicles*, vol. 8, no. 1, pp. 639–648, Jan. 2023.

[26] M. Kuhn and K. Johnson, *Applied Predictive Modeling*. New York, NY, USA: Springer, 2013.

[27] E. L. Russell, L. H. Chiang, and R. D. Braatz, *Data-Driven Methods for Fault Detection and Diagnosis in Chemical Processes*. London, U.K.: Springer, 2000.

[28] L. V. D. Maaten and G. E. Hinton, "Visualizing data using t-SNE," *J. Mach. Learn. Res.*, vol. 9, no. 86, pp. 2579–2605, 2008.

[29] L. Cao et al., "Real-time tracking of renewable carbon content with AI-aided approaches during co-processing of biofeedstocks," *Appl. Energy*, vol. 360, Apr. 2024, Art. no. 122815.

[30] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, no. 7553, pp. 436–444, 2015.