PCCP

ROYAL SOCIETY OF CHEMISTRY

PAPER

View Article Online
View Journal | View Issue

Cite this: *Phys. Chem. Chem. Phys.*, 2023, **25**, 15744

Analyzing drop coalescence in microfluidic devices with a deep learning generative model

Kewei Zhu,[a] Sibo Cheng,*[b] Nina Kovalchuk,[c] Mark Simmons,[c] Yi-Ke Guo,[b] Omar K. Matar[d] and Rossella Arcucci[e]

Received 22nd December 2022,
Accepted 21st April 2023

DOI: 10.1039/d2cp05975d

rsc.li/pccp

Predicting drop coalescence based on process parameters is crucial for experimental design in chemical engineering. However, predictive models can suffer from the lack of training data and more importantly, the label imbalance problem. In this study, we propose the use of deep learning generative models to tackle this bottleneck by training the predictive models using generated synthetic data. A novel generative model, named double space conditional variational autoencoder (DSCVAE) is developed for labelled tabular data. By introducing label constraints in both the latent and the original space, DSCVAE is capable of generating consistent and realistic samples compared to the standard conditional variational autoencoder (CVAE). Two predictive models, namely random forest and gradient boosting classifiers, are enhanced on synthetic data and their performances are evaluated based on real experimental data. Numerical results show that a considerable improvement in prediction accuracy can be achieved by using synthetic data and the proposed DSCVAE clearly outperforms the standard CVAE. This research clearly provides more insights into handling imbalanced data for classification problems, especially in chemical engineering.

1 Introduction

Drop coalescence is of high industrial importance as it determines emulsion stability. Extensive effort has been devoted to comprehending and predicting coalescence under diverse hydrodynamic conditions, with many investigations seeking to minimize or maximize the likelihood of coalescence based on a specific application. Typical areas of industrial interest where coalescence should be avoided are the prediction and maximization of foam/emulsion shelf life (mild hydrodynamic conditions) or product stability during transportation (more challenging hydrodynamic conditions, including, for example, shaking). On the other hand, coalescence is desirable in separation processes, such as the elimination of aqueous droplets that contaminate an oil.

Microfluidics provide a unique opportunity to study drop coalescence under well-controlled hydrodynamic conditions and if necessary to account for the fate of each formed doublet. The use of microfluidic platforms allows for the exploration of thousands of coalescence instances with minimal material usage and energy consumption. Drop coalescence is used in microfluidics for triggering/quenching chemical reactions within a drop or for cell screening. Various designs of microfluidic chambers are used depending on the aim of investigation.[1–4]

Considering that the Machine Learning (ML) technique has widely succeeded in chemistry and chemical engineering[5] with applications spanning from quantum chemistry research[6] to molecular reaction kinetics,[7] we believe that an effective ML model can also predict the probability of drop coalescence in a microfluidic device, enabling a considerable reduction of time and cost spent on optimization of microfluidic designs including the cost of energy as well as expensive and hazardous materials used in photo-lithography.

For microfluidic drop reactions or cell screening relying on coalescence, it is obligatory that the rate of drop coalescence is close to 100%. In such scenarios, the composition of coalescing drops is set as a priority, and there are usually very limited possibilities for adjusting the properties of the continuous phase. Therefore, accurate prediction of drop coalescence based on geometrical and hydrodynamic conditions is pivotal for effective experimental design. In this study, the experimental dataset comprises outcomes of either ''coalescence'' or ''non-coalescence'' when two drops interact in a microfluidic coalescence chamber, contingent on process parameters, such as drop size and flow rate.

Here, ML can be used for the optimization of parameters of the coalescence chamber and geometry of microfluidic devices

[a] *Department of Computer Science, University of York, UK*
[b] *Data Science Institute, Department of Computing, Imperial College London, UK. E-mail: sibo.cheng@imperial.ac.uk*
[c] *School of Chemical Engineering, University of Birmingham, UK*
[d] *Department of Chemical Engineering, Imperial College London, UK*
[e] *Department of Earth Science & Engineering, Imperial College London, UK*

in general, as well as flow characteristics maximizing the drop coalescence probability. By construction, different ML methods are able to deal with various types of information in chemistry, such as images for phenomena, videos for processes, texts for descriptions, and tabular data for numerical records. For example, Lasso regression and Random Forest (RF) have been employed to predict drop coalescence based on experimental data.[8] Moreover, the usage of neural networks has helped parameterize the collision merging process[9] and surrogate models based on Deep Learning (DL) models are developed to predict the dynamics of drop interactions based on recorded videos of experiments.[10] Using effective ML protocols providing an accurate prediction of coalescence in combination with microfluidics enables, in this case, a considerable reduction in material and energy consumption during formulation optimization.

Despite the success in a large range of applications, it is widely noticed that imbalanced training data can lead to poor prediction/classification performance in ML.[11] Examples can be found in the assessment of chemical-disease relation[12] and in the classification of drug-induced injury.[13] To tackle the data imbalance problem, multiple advanced tree-based algorithms[14–17] are developed. These approaches mainly consist of building sub-models trained by partial data to alleviate the risk of overfitting caused by data imbalance. However, in the case of extreme data imbalance, the size of the training datasets in each sub-model is limited due to the constraint of label balance.[18]

Generative models are developed as a method of data augmentation.[19] From this extension, they are employed to address the issue of data imbalance. Generative adversarial network (GAN)[20] is a common method to generate artificial data, of which the generator is used to generate data, and then the discriminator judges if the synthetic data are similar to the real. Although GANs reach obvious improvement on imbalance data issue,[21] the implicit posterior distribution is difficult to optimize and requires large-scale data to converge.[22] Another method to create synthetic data is variational autoencoder (VAE).[23] It is proposed with an explicit posterior distribution, whereas VAE cannot generate data with class-level conditions. To enable VAEs to generate class-specific data, a conditional variational autoencoder (CVAE)[24] is developed with an extra classifier in the original space. Thanks to the constraint, CVAE can learn conditional representations explicitly in the original space and implicitly in the latent space simultaneously. Benefiting from the conditional representation, CVAE can generate synthetic data separately and substantially to decrease the impact of imbalanced data issue.[25–27] However, the latent space of CVAE still lacks conditional constraint. Therefore, it is unable to explicitly learn latent conditional representation, which pushes more information stored in the decoder rather than the latent space.[28] The recent work of Chagot *et al.*[29] generates synthetic data to mitigate the issue of imbalanced data but the challenge associated with latent conditional representation remains. To improve the performance of generative models, some studies have begun to focus on latent spaces.[30,31] GET-3D[32] in 3D modeling area does not constrain the outputs only. It is proposed to add an extra constraint in latent space to improve the consistency of outputs. However, all outputs from GET-3D are not according to specific labels due to the target task. For example, GET-3D is capable of generating vivid 3D cars, but there is no label-specific constraint involved. GET-3D just guides the generator to generate similar data without focusing on their class differences.

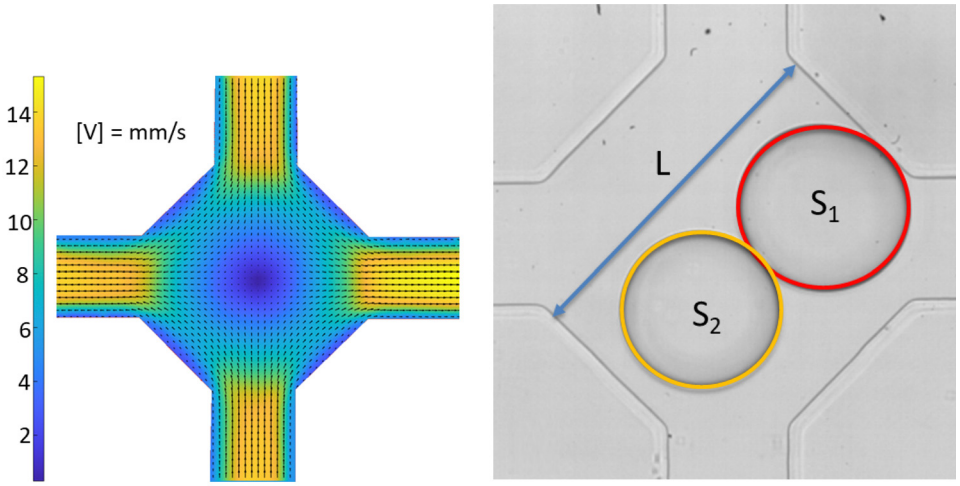
In this work, we aim to handle an experiment-based classification problem using ML methods. This study is conducted based on tabular data, which have a relatively small sample size and the results (*i.e.*, ''coalescence'' or ''non-coalescence'') correspond to a very similar distribution of conditional values. By predicting drop coalescence in the microfluidic device, we explore a new implementation to solve the dilemma of tabular data imbalance. To accomplish the prediction tasks, we use ML methods of RF and XGBoost for their interpretability and strengths in processing small sample-sized datasets. To address the data imbalance for better predictive performances, we use generative models based on variational autoencoder (VAE) to get more synthetic data. In this stage, we propose a double space conditional variational autoencoder (DSCVAE) consisting of a standard VAE and two explicit constraints on both the latent space and the original space. Its novel latent constraint guides the latent space to learn conditional representation and enables the latent space of DSCVAE to become more informative. After generating more training data using DSCVAE, the accuracy of the predictive models improved by 7.5% and 8.5%, respectively. Through sensitivity analysis based on Shapley addictive explanations (SHAP) values, the contribution of these synthetic data to predictive models is more consistent with microfluidics experiments.

The rest of the paper is organized as follows. Section 2 describes the experimental process and the initial data that originated from the experiment. Section 3 introduces the machine learning methods involved in this task, including two tree-based predictive algorithms for predicting coalescence results and our novel DSCVAE model for synthetic data. Section 4 exhibits the results of our numerical experiments. Starting with implementation details, we trace the training process of generative models, discuss the performance of predictors, and analyze the impact of the initial and generative datasets on the predictors *via* SHAP.

2 Experiments and dataset

2.1 Experiments

Coalescence experiments were carried out in a microfluidic device with rectangular channels made of PDMS using standard soft lithography.[33] Aqueous drops (double distilled water from water still Aquatron A 4000 D, Stuart) in a continuous phase of silicone oil (5 cSt, Aldrich) were formed using flow-focusing[34] in two symmetrical cross-junctions and met within a coalescence chamber shown in Fig. 1. Drop movement and coalescence were recorded using a high-speed video camera Photron SA-5 connected to an inverted microscope Nikon Eclipse



(a) (b)

Fig. 1 Drop coalescence experiments in a microfluidic device.

Ti2-U at 1000 frames per second and spatial resolution of 2 μm per pixel. The images were processed using ImageJ[35] to find drop sizes.

The chamber has two entrances or input channels (top and bottom channel in Fig. 1) and two exits or output channels (left and right channel in Fig. 1) of width 315 ± 5 μm and depth 140 ± 5 μm. The chamber size, **L**, shown in Fig. 1b was in the range of 901 ± 11 μm. The flow field in the chamber and adjacent channels is shown in Fig. 1a, with the velocity magnitude shown in mm s^{-1}. It was measured using ghost particle velocimetry, a non-invasive technique using the speckle pattern produced by light scattered by particles smaller than the diffraction limit as a tracer.[36,37] As Fig. 1a shows, there is a stagnation point in the centre of the chamber due to flow symmetry. Drops arrive into the chamber with various time delays between them caused by fluctuations in the flow rate of continuous and dispersed phases supplied by syringe pumps (Al-4000, World Precision Instruments), inevitable deviations in the channel size, and fluctuations in channel resistance due to drop presence.

In an ideal situation, the drops move along the symmetry axis of input channels and the first drop arriving in the chamber is trapped at the stagnation point until the arrival of the second drop. In reality, the flow fluctuations and effect of the second drop on the flow field in the chamber result in drop encounters at various positions around the stagnation point. Following an encounter, the drops form a doublet, *i.e.*, start to move together. If the doublet axis coincides with that of the input channels, which is also one of the chamber symmetry axes, the doublet is trapped in the compression flow provided by the continuous phase. In this case, the coalescence probability is 100% and compositions of continuous and dispersed phases affect only the time span between the doublet formation and coalescence. Because of ever-present fluctuations as well as the inevitable small asymmetries of real devices, the doublet begins to rotate/translate to one of the output channels; a doublet rotated and moved from its initial position as shown in Fig. 1b. Doublet rotation results in its transfer from a region of compression flow to one of extensional flow when its axis coincides with the axis of output channels. The extensional flow leads to drop detachment resulting in a non-coalescence event.

In general, the outcome of the drop encounter depends on the relative time scales of drainage of the continuous phase from the film separating the drops and the transition of the doublet from the region of compression flow to that of extensional flow: if the doublet does not rotate at all, the coalescence probability is 100%; if it rotates very fast, the coalescence probability is zero, *i.e.*, the coalescence probability increases with decreasing flow rate. However, the very slow flows are incompatible with the requirement of high throughput being a measure of the efficiency of microfluidic devices. Moreover, in a certain range of flow rates, the drainage time decreases with an increase of the continuous phase flow rate due to diminishing inertial effects[38], or possibly even reversing, the dependence of the coalescence probability on the flow rate.

Therefore, process optimization is necessary to determine the conditions under which coalescence probability remains large while maintaining a sufficiently high throughput. The optimization parameters include $\mathbf{x}_{\text{flow}}$, the total flow rate; $\mathbf{x}_{\text{drop1}}$ and $\mathbf{x}_{\text{drop2}}$, the drop diameters; and $\mathbf{x}_{\text{dt}}$, the time delay between drops. The total flow rate is calculated as the sum of the flow rates of continuous and dispersed phases, with a syringe pump dispensing accuracy of ±1%. The drop sizes are determined from their area in the plane of observation, **S1** and **S2** in Fig. 1b, with a maximum error of ±1%. The drop diameters are normalized by the chamber width. The time delay, $\mathbf{x}_{dt}$ between drops has an uncertainty of 2 ms.

2.2 Dataset

The experimental dataset consists of 1531 samples in total, including one label **y**, influenced by four features $X^{\text{features}} = [\mathbf{x}_{\text{flow}},\mathbf{x}_{\text{drop1}},\mathbf{x}_{\text{drop2}},\mathbf{x}_{dt}]$ in various ranges. Label **y** has two classes:

"coalescence" and "non-coalescence". The features are scaled for their comparability by min-max normalization according to

$$\mathbf{x}_{\text{scaled}} = \frac{\mathbf{x} - \mathbf{x}_{\min}}{\mathbf{x}_{\max} - \mathbf{x}_{\min}}, \text{ where } \mathbf{x} \in \{\mathbf{x}_{\text{flow}}, \mathbf{x}_{\text{drop1}}, \mathbf{x}_{\text{drop2}}, \mathbf{x}_{\text{dt}}\} \quad (1)$$

here $\mathbf{x}_{\max}$ and $\mathbf{x}_{\min}$ are the maximum value and minimum value of $\mathbf{x}$, and $\mathbf{x}_{\text{scaled}}$ for their normalised outcomes.

The statistical characteristics of binary classifications are given in detail in Fig. 2. Fig. 2a shows the imbalance percentages of "coalescence" and "non-coalescence" samples, where "coalescence" accounts for 76% and "non-coalescence" only accounts for 24%. The value distributions of four features compared between two labels are shown in Fig. 2b. Fig. 2c shows the median, quartiles and whiskers of X^{features}. These values of the two types of labels under each feature are not significantly different. Fig. 2d shows that X^{features} values are not subject to the standard normal distribution, and confirms again that the distributions are similar with different labels. To sum up, X^{features} is imbalanced in volume regarding two opposite classes, though identical distributions. In other words, there are significantly more samples of "coalescence" compared to "non-coalescence" in the training dataset, and it is difficult to separate the two classes of data by simply looking at their input distributions. The latter makes the prediction task even more challenging.

ML models must be evaluated using data that have not been used for training. Moreover, a validation set is needed for tuning the hyperparameters of the predictive models. As there is no dominant experimental result in actuality, the dataset is split into a balanced test set and a balanced validation set ("balanced" here means with the same number of "coalescence" and "non-coalescence"), instead of being divided proportionally. The remaining samples are used as the training set. The imbalanced ratio (IR) here is equal to the number of major results ("coalescence") divided by the number of minor results ("non-coalescence"). Thus, the whole dataset is split into three

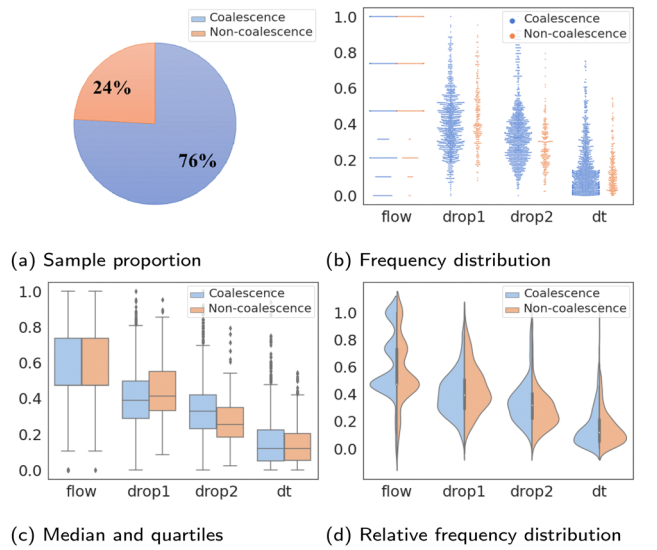


Fig. 2 Experimental dataset characteristics.

Table 1 Dataset split

	Coalescence	Non-coalescence	IR	Total
Total dataset	1162	369	3.15	1531
Training dataset	1012	219	4.62	1231
Balanced training dataset	219	219	1	438
Validation dataset	50	50	1	100
Test dataset	100	100	1	200

mutually-independent sets, namely the balanced training dataset, validation dataset and test dataset, for subsequent procedures (shown in Table 1). It is worth mentioning that all comparative models in this paper are trained on this balanced training set to avoid overfitting.

After dataset splitting, IR becomes larger from 3.15 of the total experimental set to 4.62 of the training dataset which makes the classification task more difficult.

3 Methods

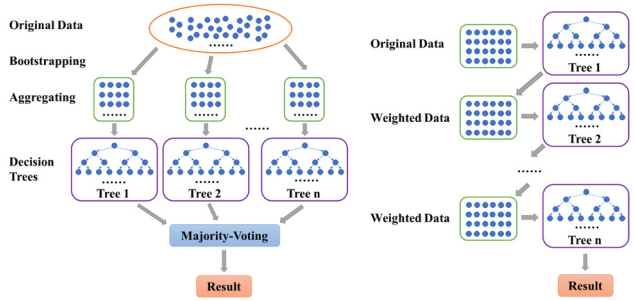
In this section, we introduce two advanced tree-based models, RF and XGBoost, to make predictions for coalescence results. In addition, we design a VAE variant, DSCVAE, to solve the aforementioned insufficiency and imbalance problem by generating synthetic data.

3.1 Predictive models

Tree-based models belong to the class of widely-used algorithms for assessing tabular data, which outperform neural networks for small and intermediate-sized datasets.[39] Therefore, they are applied to process the small-scale data of this study. The Decision Tree (DT) is a classical ML algorithm and has natural interpretability. It has been used in ML models for classification and regression.[40,41] To avoid overfitting, some parameters, such as max depth $\mathbf{d}_{\max}$, should be set to enhance its robustness.[42] Nevertheless, since minor changes may lead to an entirely disparate tree,[43] some variants combined with ensemble learning approaches, such as RF (see Section 3.1.1) and XGBoost (see Section 3.1.2), have been utilized to make predictions. The former and latter adopt "bagging" and "boosting" algorithms, respectively.

3.1.1 Random forest. RF is a widely applied model focusing on imbalanced data thanks to the "bagging" technique[44] (Fig. 3a). Here, the term "Bagging" corresponds to the abbreviation of "Bootstrap aggregation", that is, multiple models trained by bootstrap sampling in a parallel process.[45,46] An RF selects an irregular portion of samples from the training set with put-backs and randomly selects some features for training; the number of sub-models is $\mathbf{n}_{\text{estimators}}$. The results of sub-models vary, because of different sub-datasets. After that, a prediction is made by taking a majority vote on classification trees. The benefit of this approach is that it not only settles the over-fitting problems but also helps to reduce prediction errors in general.[47]

3.1.2 XGBoost. Unlike "bagging" where submodels expand in parallel, "boosting" is a sequential structure. It has been



(a) Random Forest (RF) (b) XGBoost

Fig. 3 Flowchart of the predictive models.

used in Gradient Boosting Decision Tree (GBDT) to decrease variance and bias of tree models.[48] The new tree is built on the basis of the previous one, to decrease the residuals to the target. XGBoost is an efficient implementation of GBDT (Fig. 3b) based on a gradient boosting algorithm.[49] It is also robust to over-fitting. Its important parameters also include $\mathbf{d}_{\max}$ and $\mathbf{n}_{\text{estimators}}$, similar to RF.

3.1.3 Metrics. In order to evaluate the prediction performance from different perspectives, and equip data characteristics with interpretability, several common approaches were chosen in this study.

To introduce model-performance measures for binary classification, we used the following metrics to evaluate the classification performance, including accuracy, precision (eqn (2)) from an actual perspective, recall (eqn (3)) from a predictive perspective, *F*1 score (eqn (4)), and confusion matrix (Table 2).

• **Confusion matrix** includes four categories. Here, true positive (TP) and true negative (TN) indicate correctly classified labels, and false positive (FP) means the amount of incorrectly predicted positive results, and *vice versa*, false negative (FN) shows incorrectly classified negative results.

• Precision or positive predictive value:

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}} \tag{2}$$

• Recall or true positive rate:

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}} \tag{3}$$

• *F*1 score is the harmonic mean of precision and recall:

$$\text{F1}_{\text{score}} = \frac{2}{\frac{1}{\text{Precision}} + \frac{1}{\text{Recall}}} = 2 \times \frac{\text{Precision} \times \text{TPR}}{\text{PPV} + \text{Recall}} \tag{4}$$

Table 2 An example of the confusion matrix

	Prediction	
Real	Negative	Positive
Negative	TN	FP
Positive	FN	TP

SHAP values (SHAP) are based on the classical Shapley value in game theory.[50,51] It is used in this study to interpret feature influences toward predictive results by accounting for how each feature affects the tree model.

3.2 Generative model

The imbalance (Fig. 2a and b) between the two classes of labels caused a preference in the tree model prediction. Moreover, balancing the data (*i.e.*, deleting the excess) reduces the number of majority samples and thus the total number of samples, which could make the training dataset insufficient for predictive models. This is the reason that the number of samples should be increased with the generative model.

Generative models, namely variational autoencoders, are employed in this study to address data imbalance by generating data with different labels separately. These data are then used as inputs for predictive models to enhance performance.

3.2.1 Variational autoencoder (VAE). The Autoencoder (AE) is a feed-forward neural network belonging to unsupervised learning.[52] A typical AE contains symmetrical architectures as shown in Fig. 4a. The input $\mathbf{x}$ is encoded to obtain the latent variable $\mathbf{z}$, from which the synthetic output $\hat{\mathbf{x}}$ is decoded. Therefore, an encoder E and a decoder D are represented by

$$\mathbf{z} = E(\mathbf{x}) \quad \text{and} \quad \hat{\mathbf{x}} = D(\mathbf{z}) \tag{5}$$

This network is trained by minimzing the reconstruction error $L(\mathbf{x},\hat{\mathbf{x}})$ between $\hat{\mathbf{x}}$ and $\mathbf{x}$, usually using Mean Square Error (MSE):

$$\mathscr{L}_{\text{MSE}}(\mathbf{x}, \hat{\mathbf{x}}) = \frac{1}{N}\sum_{i=0}^{N}(\mathbf{x} - \hat{\mathbf{x}}_i)^2 \tag{6}$$

This loss shows the efficiency of the data compression strategy. It is used to compare the difference between model output and input. Thus, AEs are traditionally used for dimensionality reduction[53,54] and feature extraction[55] instead of data generation.

VAE stems from AE, and starts to focus on the distribution of $\mathbf{z}$ in the latent space (Fig. 4b),[23] which equips VAE with generation capability. Unlike AE, VAE constructs the distribution of $\mathbf{z}$ in the latent space and resamples $\mathbf{z}'$ over the distribution. Thus, the whole process of VAE can be expressed as

$$\mathbf{z} = E(\mathbf{x}) \text{ and } \hat{\mathbf{x}} = D(\mathbf{z}') \tag{7}$$

Latent distribution $q(\mathbf{z}|\mathbf{x})$ is trained to be subjected to *a posterior* distribution $p(\mathbf{z}|\mathbf{x})$, usually assumed Gaussian. The approximation between $q(\mathbf{z}|\mathbf{x})$ and $p(\mathbf{z}|\mathbf{x})$ is measured by Kullback–Leibler Divergence (KLD) loss as

$$\mathscr{L}_{\text{KLD}}(q(\mathbf{z}|\mathbf{x}) \parallel p(\mathbf{z}|\mathbf{x})) = -\frac{1}{2}\sum\left(1 + \log(\sigma^2) - \mu^2 - \sigma^2\right) \tag{8}$$

KLD loss is used to regularize the distribution of hidden variables $q(\mathbf{z}|\mathbf{x})$ in the latent space to converge to a Gaussian distribution $p(\mathbf{z}|\mathbf{x})$. Thus, VAE enables the model with interpretability.

Resampled latent variables $\mathbf{z}'$ can be characterized by a normal distribution with mean μ and variance σ denoted as

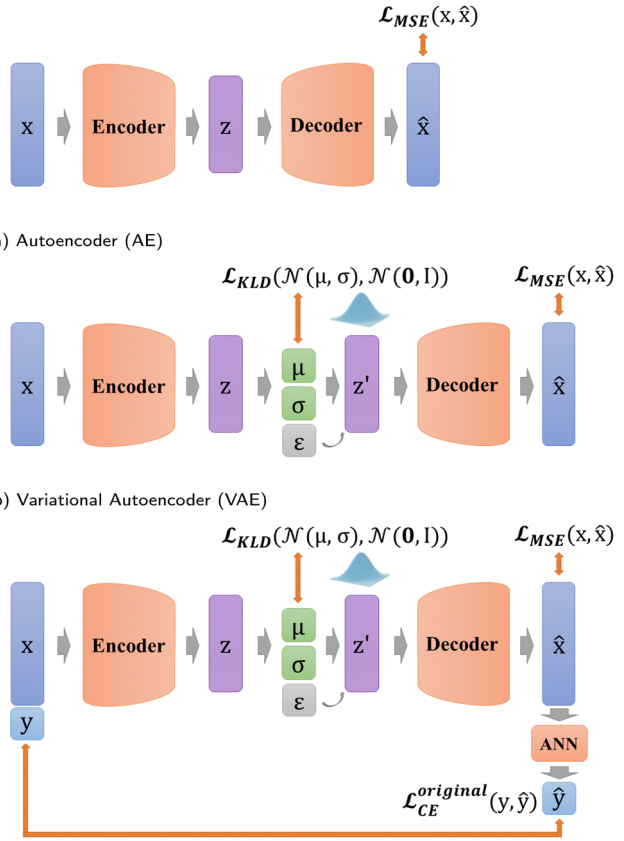


Fig. 4 Existing VAE structures.

$\mathcal{N}(\mu,\sigma^2)$, with ε sampled from a Gaussian distribution as

$$\mathbf{z}' \approx \mu + \varepsilon \cdot \sigma, \quad \text{where } \varepsilon \sim \mathcal{N}(0,\mathbf{I}) \tag{9}$$

To sum up, the total loss function of VAE[56] presents as

$$\mathcal{L} = \mathcal{L}_{\text{MSE}}(\mathbf{x},\hat{\mathbf{x}}) + \mathcal{L}_{\text{KLD}}(q(\mathbf{z}|\mathbf{x})\|p(\mathbf{z}|\mathbf{x})) \tag{10}$$

VAE can be used not only like AEs for dimensionality reduction, but also to generate "consistent" data that are different from inputs because of their latent state representation.[29] Here, by 'consistent', we refer to the generated data whose statistical properties mimic those of the original dataset. However, it is also an unsupervised learning method and cannot handle condition-specific samples[57] (*e.g.*, "coalescence" and "non-coalescence" in this study) because conditional prerequisites are involved neither in the encoding nor decoding processes.

3.2.2 Conditional variational autoencoder (CVAE). To modulate the latent distribution according to prior conditions, a CVAE[24] is devised to generate data with its conditional information. CVAE concatenates feature inputs **x** with auxiliary condition **y**, thus, it becomes a supervised learning model.

In addition, in our study, labels **y** are not put directly into the encoder with X^{features} together to prevent label leakage during training. In other words, the label **y** is not given to the encoder when constructing latent variables. As shown in Fig. 4c, firstly, X^{features} are used as inputs for the encoder only. Then, their corresponding labels **y** are used as the target of artificial neural network (ANN) classifier results $\hat{\mathbf{y}}$, and this ANN classifier is jointly trained with the entire VAE network. The difference is measured by predicted labels $\hat{\mathbf{y}} \in (0,1)$ and probability $\mathscr{V}$ *via* cross entropy (CE) loss according to

$$\mathcal{L}_{\text{CE}}^{\text{original}} = -(\hat{\mathbf{y}}\log\mathscr{V} + (1-\hat{\mathbf{y}})\log(1-\mathscr{V})) \tag{11}$$

Thus, the total loss can be described as

$$\mathcal{L} = \mathcal{L}_{\text{MSE}}(\mathbf{x},\hat{\mathbf{x}}) + \mathcal{L}_{\text{KLD}}(q(\mathbf{z}|\mathbf{x},\mathbf{y})\|p(\mathbf{z}|\mathbf{x},\mathbf{y})) + \mathcal{L}_{\text{CE}}^{\text{original}} \tag{12}$$

CVAE helps generate synthetic data by conditions, and it changes the distribution of the latent space indirectly. However, it is not sufficient to distinguish distributions that are very similar in the original space based on its conditional restrictions.

3.2.3 Double space conditional variational autoencoder (DSCVAE). Inspired by the latent discriminator from GET-3D,[32] we propose our DSCVAE. DSCVAE should be a more robust generative model to finely discriminate the difference between similar distributions of conditions. The constraint only at the original space enforces more information stored in the decoder according to ref. 28. In contrast, DSCVAE ensures the consistency of generated data by simultaneously assigning conditions to both the latent distribution and the original-space distribution, which is a crucial factor in the quality determination of generative models.

In DSCVAE, we first adopt two classifiers at two spaces to classify the latent representation $\mathbf{z}'$ and the original space reconstruction $\hat{\mathbf{x}}$ respectively according to their common label. Double space conditions are implemented by adding another ANN classifier for resampled variables $\mathbf{z}'$ (shown in Fig. 5), of which classification error is still measured by CE loss as

$$\mathcal{L}_{\text{CE}}^{\text{latent}} = -(\hat{\mathbf{y}}'\log\mathscr{U} + (1-\hat{\mathbf{y}}')\log(1-\mathscr{U})) \tag{13}$$

where, the latent-space predicted label is $\hat{\mathbf{y}}'$ and probability is $\mathscr{U}$.

These two classifiers contribute to distinguishing different labels, more than judging the similarity of outputs in conventional generative models. Furthermore, our novel latent classifier promotes DSCVAE to learn a more informative latent space than generative models with only original-space conditions. Thus, DSCVAE should better solve the conditional inputs

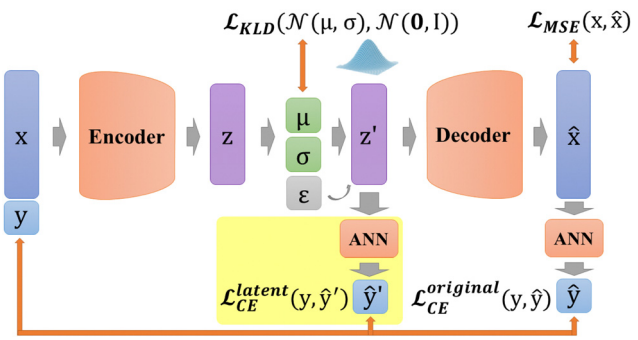


Fig. 5 Double space CVAE (DSCVAE).

("coalescence" or "non-coalescence" here) with considerable noise and similar distributions for different labels.

The total loss function can be written as

$$\mathscr{L} = \mathscr{L}_{\mathrm{MSE}}(\mathbf{x},\hat{\mathbf{x}}) + \mathscr{L}_{\mathrm{KLD}}(q(\mathbf{z}|\mathbf{x},\mathbf{y})\|p(\mathbf{z}|\mathbf{x},\mathbf{y})) + \mathscr{L}_{\mathrm{CE}}^{\mathrm{original}} + \mathscr{L}_{\mathrm{CE}}^{\mathrm{latent}} \quad (14)$$

where $\mathscr{L}_{\mathrm{KLD}}$ and $\mathscr{L}_{\mathrm{CE}}^{\mathrm{latent}}$ both regularise the latent distribution, making the latent space more interpretable.

4 Numerical results and analysis

In this section, we compare and analyze the performance of different generative and predictive methods. Ablation tests are performed to demonstrate the strength of the proposed DSCVAE model. The SHAP value is used as a metric to interpret the model performance.

4.1 Implementation details

In the generation phase, the balanced training set (shown in Table 1, named initial dataset hereafter) is employed in generative models as inputs and also used to inspect and validate these models. When training the generative neural networks, the parameters are set as follows: batch size is 73, the learning rate is 10^{-3}, the optimizer is Adam, the scheduler is CosineAnnealing LR,[58] and the training epoch is 5000. For all generative methods, $\mathbf{z}'$ is resampled from the distribution, and added to a random four-dimensional Gaussian noise for regularizing purposes. The sum is put into the decoder to get output $\hat{\mathbf{x}}$. The samples of different labels are generated separately. Totally, we generate 6570 balanced samples from each generative model.

We then use the predictive models to evaluate the generated synthetic data. The training datasets are set as Table 3, and the validation set and test set are also shown in Table 1.

The algorithms for assessments are Random Forest (mentioned in Section 3.1.1) and XGBoost (mentioned in Section 3.1.2). The key hyperparameters $\mathbf{n}_{\mathrm{estimators}}$ and $\mathbf{d}_{\mathrm{max}}$ are chosen from a grid search. All the numerical experiments are implemented on the Google Colab platform. The DL generative models are performed on a single Tesla K80 GPU.

4.2 Training loss of the generative models

The losses of MSE, KLD and classifiers both in the original space and the latent space are plotted in Fig. 6. The MSE loss and the KLD loss are the basic loss functions used to build the VAE (Fig. 4b). As shown in Fig. 6a, MSE losses of both models tend to converge. Fig. 6b shows that the plots of KLD losses vary significantly between CVAE and DSCVAE. The classifiers are used to discriminate the conditions (*i.e.*, the label "coalescence" or "non-coalescence"). Fig. 6c shows the classifiers of the CVAE and DSCVAE set for decoder outputs at the original space, while Fig. 6d shows the classifier of DSCVAE for the resampled variables in the latent space. The classifier losses all decrease slightly before stabilization since the classifiers are used as regularizers in CVAEs.

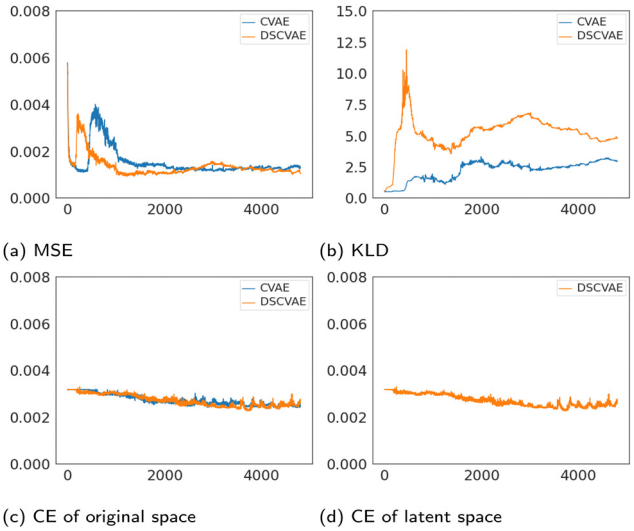


Fig. 6 Losses of the generative models.

The classifier of the latent space in DSCVAE has a direct impact on the construction of the latent distribution and indirectly affects the classification and reconstruction of the original space. Because the latent-space classifier affects the latent distribution, unsurprisingly we find that the KLD loss of DSCVAE is more unstable than CVAE, as shown Fig. 6b. In addition, the double-space classifiers make the reconstruction training converge faster, as Fig. 6a shows that the MSE loss of DSCVAE decreases faster than the MSE loss of CVAE. This may be due to the fact that the classifier of the latent space also guides the data reconstruction.

Table 3 Training dataset

	Initial	Synthetic	Total
Initial dataset	438	—	438
CVAE mixed dataset	438	438 × 15	7008
DSCVAE mixed dataset	438	438 × 15	7008

4.3 Hyperparameter tuning

The parameterization of machine learning algorithms impacts their generalization ability and prediction accuracy. In this study, the balanced validation dataset is used to find the appropriate hyperparameters, namely $\mathbf{n}_{\mathrm{estimators}}$ and $\mathbf{d}_{\mathrm{max}}$. As shown in Fig. 7, the dark blue areas indicate high prediction accuracy in the validation set. For the RF predictors, the heatmap of DSCVAE (Fig. 7e) has the largest dark areas, signifying higher prediction accuracy in this validation dataset. As for XGBoost predictors, CVAE (Fig. 7d) and DSCVAE (Fig. 7f) both lead to significantly more accurate predictions compared to solely using the original dataset (Fig. 7b). Overall, the predictive models training on DSCVAE synthetic data show a more significant improvement in RF compared to XGboost in the validation dataset.

We fix $\mathbf{n}_{\mathrm{estimators}}$ and $\mathbf{d}_{\mathrm{max}}$ that perform the best on the validation dataset (according to Fig. 7) since the test dataset should not be used for tuning hyperparameters. These hyperparameters are used for predicting the test data. The exact values of these tuned hyperparameters are shown in Table 4.

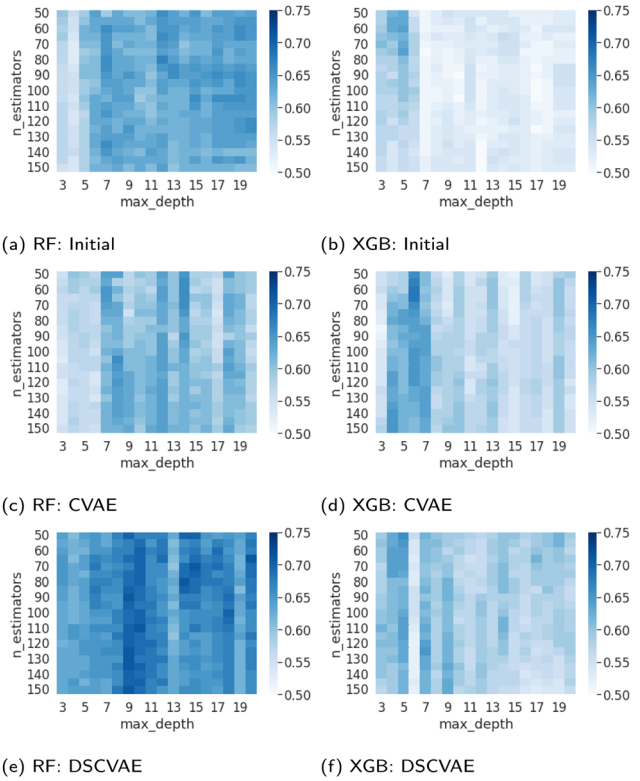


Fig. 7 Validation heatmaps for the tuning hyperparameters of the various generative models.

4.4 Predictive results

Table 5 shows the prediction results based on different synthetic training datasets for RF and XGBoost. To further examine the necessity of training two classifiers jointly in DSCVAE, we set up another CVAE (L) as an ablation experiment. The CVAE (L) mixed dataset combines the initial dataset and the synthetic data generated from the CVAE with only one classifier in the latent space. It has the same hyperparameter tuning process on the predictive model as the other three sets.

As displayed in Table 5, the predictors trained by DSCVAE mixed (synthetic and initial) dataset have the best test results in terms of all metrics investigated. In fact, all generated datasets improve the accuracy of predictive models. Compared to the predictive models only trained on the initial dataset, DSCVAE substantially enhances the prediction accuracy by 7.5% and 8.5% for RF and XGBoost, respectively. The predictive models trained by DSCVAE mixed dataset also have the best performance in precision, recall and $F1$ score. In addition, DSCVAE helps to reduce predictors over-fitting between validation results and test results. As shown in Table 5, DSCVAE synthetic data narrows the gap between validation accuracy and testing accuracy from 8.5% to 5% (RF) and from 6% to −0.5% (XGBoost) compared to the initial dataset.

Table 5 Test results using Initial or mixed datasets

Evaluative method	Generative method	Accuracy (%)	Precision (%)	Recall (%)	$F1$ score (%)
RF	None	58.50	58.57	58.50	58.42
	CVAE	63.00	63.13	63.00	62.91
	CVAE (L)	60.50	60.59	60.50	60.42
	DSCVAE	**66.00**	**66.42**	**66.00**	**65.78**
XGB	None	58.00	58.01	58.00	57.98
	CVAE	63.00	63.01	63.00	63.00
	CVAE (L)	61.50	61.59	61.50	61.42
	DSCVAE	**66.50**	**66.58**	**66.50**	**66.46**

We consider ''coalescence'' as positive samples and ''non-coalescence'' as negative samples. Shown in Table 6 is the number of correct and incorrect predictions. The accurate predictions for both types increase compared to the predictive models trained on the initial dataset. The true positive rates increase by 7.4% and 12.5% respectively, and the true negative rates increase by about 17% for both RF and XGBoost. It is worth mentioning that the prediction accuracy of 66% for DSCVAE might still present some limitations for the practical use of the predictive model. However, increasing the accuracy from 58% to 66% for a binary classification problem is still significant and it clearly demonstrates the strength of the proposed generative model. With this model, we predict the outcome, ''coalescence'' or ''non-coalescence'', of each forming doublet. Due to the dependence on the local process parameters, the probability of drop coalescence varies between 0 and 1, leaving some cases indeterminable. As a result, the predictability of the model cannot be very high. On the other hand, the reliable generation of a considerable amount of synthetic data will enable the prediction of coalescence probability for the prescribed set of data. Furthermore, a more accurate predictive model results in a more meaningful interpretability analysis as detailed in the following section.

4.5 Intrepretability

We use the SHAP value here to reveal the contributions of samples in training datasets. The values can be analyzed both

Table 4 Validation results for hyperparameter tuning

Evaluative method	Generative method	Accuracy (%)	Precision (%)	Recall (%)	$F1$ score (%)	$\mathbf{n}_{\text{estimators}}$	$\mathbf{d}_{\max}$
RF	None	67.00	67.34	67.00	66.84	80	12
	CVAE	66.00	66.10	66.00	65.95	105	8
	DSCVAE	71.00	71.01	71.00	71.00	125	9
XGB	None	64.00	64.09	64.00	63.94	60	5
	CVAE	68.00	68.12	68.00	67.95	60	6
	DSCVAE	66.00	66.03	66.00	65.99	50	5

Table 6 Confusion matrices for RF and XGBoost. Here, 'TP' and 'TN', and 'FP' and 'FN' correspond to true positive and true negative predictions, and false positive and false negative predictions, respectively, where "positive" and "negative", in turn, correspond to coalescence and non-coalescence events, respectively

Evaluative method	Generative method	Validation				Test			
		TP	TN	FP	FN	TP	TN	FP	FN
RF	None	30	37	20	13	54	63	46	37
	CVAE	31	35	19	15	58	68	42	32
	CVAE (L)	30	38	20	12	56	65	44	35
	DSCVAE	35	36	15	13	**58**	**74**	42	26
XGB	None	30	34	20	16	56	60	44	40
	CVAE	32	36	18	14	62	64	38	36
	CVAE (L)	28	38	22	12	57	66	43	34
	DSCVAE	32	34	18	16	**63**	**70**	37	30

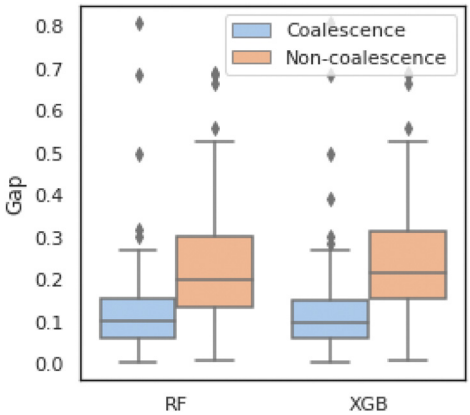


Fig. 9 "Gap" comparison according to predictive labels from DSCVAE enhanced predictors.

in global and local aspects. The bar plots (Fig. 8(a)–(f)) show the mean of SHAP values of each feature. The models trained on the DSCVAE mixed (*i.e.*, synthetic and initial) data have more similar SHAP values for different inputs. In Fig. 8c and f, the SHAP values of all four features are relatively close compared to other approaches, indicating that the contributions of these features to the predictive models are similar. For example, the two features "drop1" and "drop2", in Fig. 8b and e show a large gap between them, which may be caused by overfitting in CVAE. Whereas, as shown in Fig. 8c and f, the SHAP values of "drop1" and "drop2" are more similar, and thus, more realistic.

In addition, scatter plots (Fig. 8(g)–(l)) combine feature importances and feature effects. Each point on the scatter plot is a SHAP value of the feature in a training sample. The position on the *x*-axis is determined by the SHAP value which represents the feature contribution. The colours of the scatter points represent the feature values (red/blue for positive/negative contribution to "coalescence"). Most of the points in Fig. 8i and l are clustered, indicating that their SHAP values are closer. This means that the individual samples in each feature contribute similarly to the model, which improves the model's robustness. Also, it can be seen from Fig. 8(i) and (l) that the contribution of "drop1" and "drop2" are reversed. That is, the higher size of "drop1" contributes negatively to the probability of "coalescence", while the higher size of "drop2" contributes positively to the probability of "coalescence". In fact, it can be noticed from Fig. 2 that in general, the average size of "drop1" is larger than that of "drop2". In addition, when the sizes of "drop1" and "drop2" are close, the drops have a higher probability of coalescence. To further confirm this trend, we plot in Fig. 9 the impact of the drop size difference (*i.e.*, $|\mathbf{x}_{\mathrm{drop1}} - \mathbf{x}_{\mathrm{drop2}}|$) on the model predictions of RF and XGBoost. It can be concluded that smaller differences in drop sizes can lead to a significantly higher probability of drop coalescence. In other words, to obtain a higher drop coalescence in experiments, one should minimize the drop size difference. This explains the opposite contribution of "drop1" size and "drop2" size in Fig. 8 and also confirms the consistency of DSCVAE since both the RF and the XGBoost here are trained using the synthetic data. For the predictive models trained on the initial and DSCVAE datasets, it can be seen that the impact of $\mathbf{x}_{\mathrm{d}\ell}$ on model output is limited. It is worth mentioning that we consider here only the case when two drops form a doublet. For large $\mathbf{x}_{\mathrm{dt}}$, drops can proceed as singlets, without forming a doublet and therefore are not included in the consideration.

In summary, DSCVAE mixed data outperforms the initial dataset and CVAE mixed dataset in training predictors in classification tasks. It not only reduces the overfitting gap between validation accuracy and test accuracy, but also increases overall

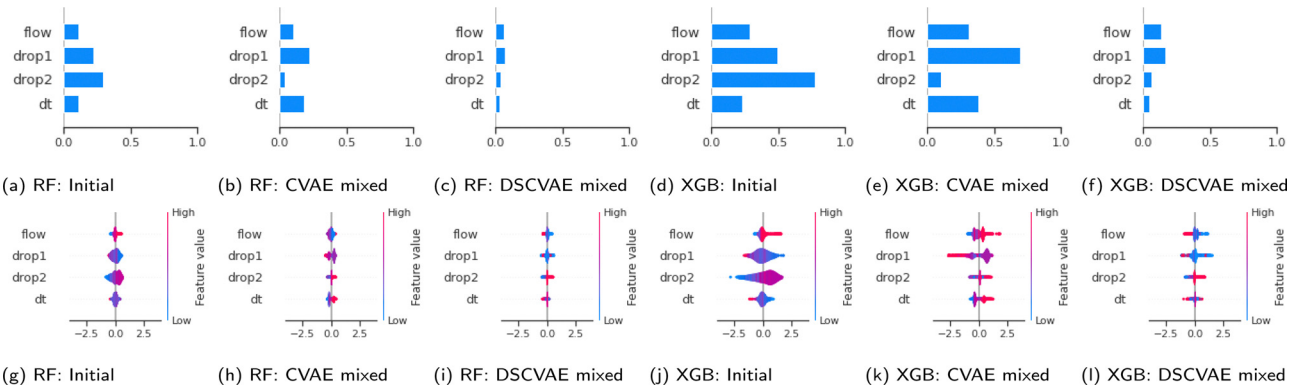


Fig. 8 SHAP values for training dataset.

predictive performance. By analyzing XGBoost values of each predictive model, we find that DSCVAE mixed dataset contributes to features' homogeneity, allowing features to be treated equally.

5 Conclusions

In this study, we propose a novel generative model named the double space conditional variational autoencoder for generating synthetic tabular data. Compared to traditional generative models, the proposed model utilizes two classifiers in the latent and original spaces to regularize the data generation process. The DSCVAE model is applied to generate synthetic data for training predictive models of drop coalescence in a microfluidics device. Numerical results show that the predictive models, namely RF and XGBoost, achieve substantially more accurate and robust predictions than those trained on purely experimental data in terms of several metrics. With the help of DSCVAE synthetic data, both RF and XGBoost manage to obtain a prediction accuracy of about 66%, which is about 8% higher than using only the experimental data. It is worth mentioning that, from a prediction perspective, this is an extremely challenging task due to the following reasons: firstly the imbalanced dataset, and more importantly, the similar input distributions of "coalescence" and "non-coalescence" samples. These predictive models are valuable for optimizing experimental design, and insightful analysis is provided through the computation of Shapley addictive explanation (SHAP) values. The results show that reducing the gap between two drop sizes increases the probability of coalescence. Moreover, ablation tests demonstrate the strength of DSCVAE compared to non-conditional VAE and standard CVAE (with constraints either in the latent or the original space).

This study offers a paradigm for addressing classification problems with limited and imbalanced tabular data. It also highlights the potential of data-driven methods for predicting microfluidic drop coalescence. The proposed methodology can be applied to other materials and devices. Future work may include further improving the interpretability and robustness of DSCVAE, for instance, by imposing physical or chemical constraints. The future work will also aim at extending the model parametric space to include viscosities of continuous and dispersed phases, interfacial tension, dynamic surfactant effects, and the presence of surfactants in different phases.

Code and data availability

The computational part of this study is performed using Python language. The code is available at: **https://github.com/DL-WG/dscvae-for-drop-coalescence**.

Experimental data are available upon reasonable request to Dr Nina Kovalchuk (n.kovalchuk@bham.ac.uk).

Acronyms

AE	Autoencoder
ANN	Artificial neural network
CE	Cross entropy
CVAE	Conditional variational autoencoder
DL	Deep learning
DSCVAE	Double space conditional variational autoencoder
DT	Decision tree
FN	False negative
FP	False positive
GAN	Generative adversarial network
GBDT	Gradient boosting decision tree
IR	Imbalanced ratio
KLD	Kullback–Leibler divergence
ML	Machine learning
MSE	Mean square error
RF	Random forest
SHAP	Shapley addictive explanations
TN	True negative
TP	True positive
VAE	Variational autoencoder
XGB	XGBoost

Conflicts of interest

There are no conflicts to declare.

Acknowledgements

This research is funded by the EP/T000414/1 PREdictive Modelling with Quantification of UncERtainty for MultiphasE Systems (PREMIERE). This work is partially supported by the Leverhulme Centre for Wildfires, Environment and Society through the Leverhulme Trust, grant number RC-2018-023.

References

1 T. Krebs, K. Schroen and R. Boom, A microfluidic method to study demulsification kinetics, *Lab Chip*, 2012, **12**(6), 1060–1070.
2 Z. Liu, X. Wang, R. Cao and Y. Pang, Droplet coalescence at microchannel intersection chambers with different shapes, *Soft Matter*, 2016, **12**(26), 5797–5807.
3 A. Shenoy, C. V. Rao and C. M. Schroeder, Stokes trap for multiplexed particle manipulation and assembly using fluidics, *Proc. Natl. Acad. Sci. U. S. A.*, 2016, **113**(15), 3976–3981.
4 X. Niu, S. Gulati, J. B. Edel and A. J. Demello, Pillar-induced droplet merging in microfluidic circuits, *Lab Chip*, 2008, **8**(11), 1837–1841.
5 J. A. Keith, V. Vassilev-Galindo, B. Cheng, S. Chmiela, M. Gastegger, K.-R. Müller and A. Tkatchenko, Combining machine learning and computational chemistry for predictive insights into chemical systems, *Chem. Rev.*, 2021, **121**(16), 9816–9872.
6 P. O. Dral, Quantum chemistry in the age of machine learning, *J. Phys. Chem. Lett.*, 2020, **11**(6), 2336–2347.

7 M. Meuwly, Machine learning for chemical reactions, *Chem. Rev.*, 2021, **121**(16), 10218–10239.
8 E. Wikramanayake and V. Bahadur, Statistical modeling of electrowetting-induced droplet coalescence for condensation applications, *Colloids Surf., A*, 2020, **599**, 124874.
9 C. F. Rodrguez Genó and L. Alfonso, Parameterization of the collision-coalescence process using series of basis functions: Colnetv1. 0.0 model development using a machine learning approach, *Geosci. Model Dev.*, 2022, **15**(2), 493–507.
10 Y. Zhuang, S. Cheng, N. Kovalchuk, M. Simmons, O. K. Matar, Y.-K. Guo and R. Arcucci, Ensemble latent assimilation with deep learning surrogate model: application to drop interaction in a microfluidics device, *Lab Chip*, 2022, **22**(17), 3187–3202.
11 H. He and E. A. Garcia, Learning from imbalanced data, *IEEE Trans. Knowledge Data Eng.*, 2009, **21**(9), 1263–1284.
12 S. Mitra, S. Saha and M. Hasanuzzaman, A multi-view deep neural network model for chemical-disease relation extraction from imbalanced datasets, *IEEE J. Biomed. Health Inf.*, 2020, **24**(11), 3315–3325.
13 S. Thakkar, M. Chen, H. Fang, Z. Liu, R. Roberts and W. Tong, The liver toxicity knowledge base (lktb) and drug-induced liver injury (dili) classification for assessment of human liver injury, *Expert Rev. Gastroenterol. Hepatol.*, 2018, **12**(1), 31–38.
14 C. Su, S. Ju, Y. Liu and Z. Yu, Improving random forest and rotation forest for highly imbalanced datasets, *Intell. Data Anal.*, 2015, **19**(6), 1409–1432.
15 J. Błaszczyński and J. Stefanowski, Neighbourhood sampling in bagging for imbalanced data, *Neurocomputing*, 2015, **150**, 529–542.
16 M. Bader-El-Den, E. Teitei and T. Perry, Biased random forest for dealing with the class imbalance problem, *IEEE Trans. Neural Networks Learn. Syst.*, 2018, **30**(7), 2163–2172.
17 L. E. B. Ferreira, H. M. Gomes, A. Bifet and L. S. Oliveira, Adaptive random forests with resampling for imbalanced data streams, in *2019 International Joint Conference on Neural Networks (IJCNN)*, IEEE, 2019, pp. 1–6.
18 G. E. Batista, R. C. Prati and M. C. Monard, A study of the behavior of several methods for balancing machine learning training data, *ACM SIGKDD Explor. Newsl.*, 2004, **6**(1), 20–29.
19 J. Wei and K. Zou, EDA: Easy Data Augmentation Techniques for Boosting Performance on Text Classification Tasks, *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, Association for Computational Linguistics, 2019, Hong Kong, China, pp. 6382–6388.
20 I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville and Y. Bengio, Generative adversarial networks, *Commun. ACM*, 2020, **63**(11), 139–144.
21 J. Moon, S. Jung, S. Park and E. Hwang, Conditional tabular gan-based two-stage data generation scheme for short-term load forecasting, *IEEE Access*, 2020, **8**, 205327.
22 R. Burks, K. A. Islam, Y. Lu and J. Li, Data augmentation with generative models for improved malware detection: A comparative study, in *2019 IEEE 10th Annual Ubiquitous Computing, Electronics & Mobile Communication Conference (UEMCON)*. IEEE, 2019, pp. 0660–0665.
23 D. P. Kingma and M. Welling, Auto-encoding variational bayes, *Stat*, 2014, **1050**, 1.
24 K. Sohn, H. Lee and X. Yan, Learning structured output representation using deep conditional generative models, in *Advances in Neural Information Processing Systems*, ed. C. Cortes, N. Lawrence, D. Lee, M. Sugiyama and R. Garnett. Curran Associates, Inc., 2015, vol. 28. Available: **https://proceedings.neurips.cc/paper/2015/file/8d55a249e6baa5c06772297520da2051-Paper.pdf=0pt**.
25 K. Huang and X. Wang, Ada-incvae: Improved data generation using variational autoencoder for imbalanced classification, *Appl. Intell.*, 2022, **52**(3), 2838–2853.
26 X. Chen, J. Xu, R. Zhou, W. Chen, J. Fang and C. Liu, Trajvae: A variational autoencoder model for trajectory generation, *Neurocomputing*, 2021, **428**, 332–339.
27 Y. Yang, K. Zheng, C. Wu and Y. Yang, Improving the classification effectiveness of intrusion detection by using improved conditional variational autoencoder and deep neural network, *Sensors*, 2019, **19**(11), 2528.
28 K. He, X. Chen, S. Xie, Y. Li, P. Dollár and R. Girshick, Masked autoencoders are scalable vision learners, in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 16000–16009.
29 L. Chagot, C. Quilodrán-Casas, M. Kalli, N. M. Kovalchuk, M. J. Simmons, O. K. Matar, R. Arcucci and P. Angeli, Surfactant-laden droplet size prediction in a flow-focusing microchannel: a data-driven approach, *Lab Chip*, 2022, **22**(20), 3848–3859.
30 C. Gelada, S. Kumar, J. Buckman, O. Nachum and M. G. Bellemare, Deepmdp: Learning continuous latent space models for representation learning, *International Conference on Machine Learning*. PMLR, 2019, pp. 2170–2179.
31 P. Bojanowski, A. Joulin, D. Lopez-Pas and A. Szlam, Optimizing the latent space of generative networks, in *International Conference on Machine Learning*, PMLR, 2018, pp. 600–609.
32 J. Gao, T. Shen, Z. Wang, W. Chen, K. Yin, D. Li, O. Litany, Z. Gojcic and S. Fidler, GET3D: A Generative Model of High Quality 3D Textured Shapes Learned from Images, in *Advances in Neural Information Processing Systems*, ed. S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho and A. Oh, Curran Associates, Inc., 2022, vol. 35, pp. 31841–31854.
33 P. Kim, K. W. Kwon, M. C. Park, S. H. Lee, S. M. Kim and K. Y. Suh, Soft lithography for microfluidics: a review, *Biochip J.*, 2008, **2**(1), 1–11.
34 N. M. Kovalchuk, M. Sagisaka, K. Steponavicius, D. Vigolo and M. J. Simmons, Drop formation in microfluidic cross-junction: jetting to dripping to jetting transition, *Microfluid. Nanofluid.*, 2019, **23**(8), 1–14.
35 C. A. Schneider, W. S. Rasband and K. W. Eliceiri, Nih image to imagej: 25 years of image analysis, *Nat. Methods*, 2012, **9**(7), 671–675.

36 S. Buzzaccaro, E. Secchi and R. Piazza, Ghost particle velocimetry: accurate 3d flow visualization using standard lab equipment, *Phys. Rev. Lett.*, 2013, **111**(4), 048101.
37 N. Kovalchuk, J. Chowdhury, Z. Schofield, D. Vigolo and M. Simmons, Study of drop coalescence and mixing in microchannel using ghost particle velocimetry, *Chem. Eng. Res. Des.*, 2018, **132**, 881–889.
38 H. Yi, C. Zhu, T. Fu and Y. Ma, Efficient coalescence of microdroplet in the cross-focused microchannel with symmetrical chamber, *J. Taiwan Inst. Chem. Eng.*, 2020, **112**, 52–59.
39 L. Grinsztajn, E. Oyallon and G. Varoquaux, Why do tree-based models still outperform deep learning on tabular data? *arXiv*, 2022, preprint, arXiv:2207.08815, DOI: **10.48550/arXiv.2207.08815**.
40 H. H. Patel and P. Prajapati, Study and analysis of decision tree based classification algorithms, *Int. J. Comput. Sci. Eng.*, 2018, **6**(10), 74–78.
41 H. Gong, S. Cheng, Z. Chen, Q. Li, C. Quilodrán-Casas, D. Xiao and R. Arcucci, An efficient digital twin based on machine learning svd autoencoder and generalised latent assimilation for nuclear reactor physics, *Ann. Nucl. Energy*, 2022, **179**, 109431.
42 M. Pal and P. M. Mather, An assessment of the effectiveness of decision tree methods for land cover classification, *Remote Sens. Environ.*, 2003, **86**(4), 554–565.
43 P. Turney, Bias and the quantification of stability, *Mach. Learn.*, 1995, **20**(1), 23–33.
44 T. K. Ho, Random decision forests, in *Proceedings of 3rd international conference on document analysis and recognition*, IEEE, 1995, vol. 1, pp. 278–282.
45 L. Breiman, Bagging predictors, *Mach. Learn.*, 1996, **24**(2), 123–140.
46 S. Cheng, Y. Jin, S. P. Harrison, C. Quilodrán-Casas, I. C. Prentice, Y.-K. Guo and R. Arcucci, Parameter flexible wildfire prediction using machine learning techniques: Forward and inverse modelling, *Remote Sens.*, 2022, **14**(13), 3228.
47 L. Breiman, Random forests, *Mach. Learn.*, 2001, **45**(1), 5–32.
48 J. H. Friedman, Greedy function approximation: a gradient boosting machine, *Ann. Stat.*, 2001, 1189–1232.
49 T. Chen, T. He, M. Benesty, V. Khotilovich, Y. Tang, H. Cho, K. Chen *et al.*, *Xgboost: extreme gradient boosting*, *R package version 0.4-2*, 2015, vol. 1, no. 4, pp. 1–4.
50 H. W. Kuhn and A. W. Tucker, *Contributions to the Theory of Games*, Princeton University Press, 1953, vol. 28.
51 S. M. Lundberg, G. Erion, H. Chen, A. DeGrave, J. M. Prutkin, B. Nair, R. Katz, J. Himmelfarb, N. Bansal and S.-I. Lee, From local explanations to global understanding with explainable ai for trees, *Nat. Mach. Intell.*, 2020, **2**(1), 2522–5839.
52 M. A. Kramer, Nonlinear principal component analysis using autoassociative neural networks, *AIChE J.*, 1991, **37**(2), 233–243.
53 S. Cheng, J. Chen, C. Anastasiou, P. Angeli, O. K. Matar, Y.-K. Guo, C. C. Pain and R. Arcucci, Generalised latent assimilation in heterogeneous reduced spaces with machine learning surrogate models, *J. Sci. Comput.*, 2023, **94**(1), 1–37.
54 S. Cheng, I. C. Prentice, Y. Huang, Y. Jin, Y.-K. Guo and R. Arcucci, Data-driven surrogate model with latent data assimilation: Application to wildfire forecasting, *J. Comput. Phys.*, 2022, 111302.
55 M. Yousefi-Azar, V. Varadharajan, L. Hamey and U. Tupakula, Autoencoder-based feature learning for cyber security applications, in *2017 International joint conference on neural networks (IJCNN)*, IEEE, 2017, pp. 3854–3861.
56 B. Esmaeili, H. Wu, S. Jain, A. Bozkurt, N. Siddharth, B. Paige, D. H. Brooks, J. Dy and J.-W. Meent, Structured disentangled representations, in *The 22nd International Conference on Artificial Intelligence and Statistics*, PMLR, 2019, pp. 2525–2534.
57 T. Zhao, R. Zhao and M. Eskenazi, Learning discourse-level diversity for neural dialog models using conditional variational autoencoders, in *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 2017, pp. 654–664.
58 I. Loshchilov and F. Hutter, Stochastic gradient descent with warm restarts, *Proceedings of the 5th Int. Conf. Learning Representations*, 2016, pp. 1–16.