Environmental Science and Ecotechnology 11 (2022) 100172

Contents lists available at ScienceDirect

Environmental Science and Ecotechnology

journal homepage: www.journals.elsevier.com/environmental-science-and-ecotechnology/

Original Research

Data augmentation and machine learning techniques for control strategy development in bio-polymerization process

Sizhou Wei [a], Zhiyuan Chen [b, *], Senthil Kumar Arumugasamy [c], Irene Mei Leng Chew [d]

[a] School of Computer Science, University of Nottingham, Nottingham, NG8 1BB, United Kingdom
[b] School of Computer Science, University of Nottingham Malaysia, Semenyih, 43500, Malaysia
[c] Department of Chemical and Environmental Engineering, University of Nottingham Malaysia, Semenyih, 43500, Malaysia
[d] School of Engineering, Monash University Malaysia, Subang Jaya, 47500, Malaysia

ARTICLE INFO

Article history:
Received 22 December 2021
Received in revised form
29 March 2022
Accepted 29 March 2022

Keywords:
Bio-polymerization
Variational autoencoder generative adversarial network
Random forest
Artificial neural network

ABSTRACT

Machine learning has been increasingly used in biochemistry. However, in organic chemistry and other experiment-based fields, data collected from real experiments are inadequate and the current coronavirus disease (COVID-19) pandemic has made the situation even worse. Such limited data resources may result in the low performance of modeling and affect the proper development of a control strategy. This paper proposes a feasible machine learning solution to the problem of small sample size in the bio-polymerization process. To avoid overfitting, the variational auto-encoder and generative adversarial network algorithms are used for data augmentation. The random forest and artificial neural network algorithms are implemented in the modeling process. The results prove that data augmentation techniques effectively improve the performance of the regression model. Several machine learning models were compared and the experimental results show that the random forest model with data augmentation by the generative adversarial network technique achieved the best performance in predicting the molecular weight on the training set (with an R^2 of 0.94) and on the test set (with an R^2 of 0.74), and the coefficient of determination of this model was 0.74.

1. Introduction

Enzymatic polymerization is the polymerization of enzymes as catalysts, in which low-mass monomers are converted to polymers of high molecular weight [1]. The traditional chemical polymerization process usually needs to be conducted under high temperature and pressure and the purity of the resulting polymer may be low. The enzymatic polymerization process is an alternative to the conventional polymerization technique to synthesize polycaprolactone. Conventionally, polymers are synthesized using metal or chemical catalysts, which are harmful to the environment. Hence, the enzymatic polymerization approach is regarded as an environmentally friendly technique that replaces the metal or chemical catalysts with enzymes. Enzymatic polymerization, by contrast, is an ideal process because of its milder reaction conditions and higher polymer purity [2]. Therefore, this emerging technique currently has good development prospects in the field of biochemistry [3]. Biopolymers, the reaction products of enzymatic polymerization, have been widely used in various fields. Efficient preparation of biopolymers has become one of the most active topics of current research [4–6].

Enzymatic polymerization is affected by many factors, of which temperature and reaction time are the most significant. In addition, because of the complexity of organic reactions, the relationships between the products and reaction conditions are usually nonlinear; this is the reason why molecular weight is difficult to measure. Some traditional techniques can be used to roughly measure the molecular weight of the polymerization process—for example, gel permeation chromatography and size-exclusion chromatography [7]—but they cannot achieve the purpose of accurate and rapid measurement.

Machine learning (ML) is a branch of artificial intelligence that has become widely applied in recent years [8]. The main purpose of ML is to improve and optimize the performance of computer programs or algorithms automatically by learning from past data or experience. ML can construct stable models by learning and mining

* Corresponding author.
E-mail address: zhiyuan.chen@nottingham.edu.my (Z. Chen).

https://doi.org/10.1016/j.ese.2022.100172

existing data and use these models to predict or classify unknown data. In particular, since the advent of the era of big data, ML has enabled the constructed models to make more timely and accurate predictions than ever before [9]. In the past several years, ML has also been used in the field of organic chemistry to control the efficiency of chemical production [10]. Data analysis technology based on ML has become one of the most active research topics and development trends.

ML usually optimizes an algorithm by continuous training on large datasets because the use of small datasets may result in overreliance on the original data and loss of generalization ability (i.e., overfitting). However, in organic chemistry and other experiment-based fields, data from real experiments inevitably suffers from the problem of inadequate dataset size. This is because the raw materials available for a complete experiment are usually limited for reasons such as experimental cost. In addition, because of the current coronavirus disease (COVID-19) pandemic, the situation is becoming worse. Such limited data resources may result in the low performance of modeling and affect the proper development of a control strategy. In addition, the inconvenience caused by the pandemic in the past two years has made it more difficult to obtain sufficient experimental data for ML models. Therefore, a scientific algorithm is urgently required to realize ML modeling on small datasets.

A variety of methods have been proposed to solve the above problems, of which the most well-known is data augmentation. Data augmentation generates data by transforming or characterizing the original data. It has been mainly used to avoid significant errors when constructing prediction models based on small datasets. During the past decade, various data augmentation algorithms—such as the variational auto-encoder (VAE) and generative adversarial network (GAN)—have been proved to improve the performance of models [11]. A VAE creates data instances that are based on the original data distribution, whereas a GAN generates data by confrontation between two networks. Previous researchers have reported that the VAE and GAN have achieved great success in improving the accuracy of classification problems. Franco et al. and Gallego et al. demonstrated that models constructed after data augmentation achieve better performance on small datasets when performing classification prediction [12,13]. In addition, Liu et al. used a GAN as a sample generator for cancer recognition and showed that the GAN improved the prediction accuracy of several classifier models [14]. The VAE and GAN have been proved to improve the performance of models when solving classification problems. However, the application of these algorithms to regression problems has rarely been considered. Ohno et al. discussed the possibility of the VAE in enlarging regression datasets and proposed the optimal solutions for diverse regression problems [15]. Haidar and Rezagholiradeh also attempted to use a GAN to solve the regression problem and proposed an improvement plan [16].

Previous researchers have proposed various efficient ML algorithms for data modeling. Researchers have demonstrated that the random forest (RF) and artificial neural network (ANN) have high learning efficiency and are widely used in regression prediction. An RF is a stable prediction algorithm with high prediction accuracy and low time consumption [17]. It is a supervised algorithm that is commonly used to solve regression and classification problems. Zhou et al. successfully predicted biomass in wheat using an RF algorithm [18]. They also believed that this result was superior to that of other ML models. The ANN has been an active research topic in ML in recent years. An ANN is a black-box model that learns complex function relations from given data. Lipik et al. established a mathematical model for synthesizing a variety of polymers, which mainly took polymerization parameters as variables and calculated the molecular weight of the polymer well [19]. On this basis, Arumugasamy, Uzir, and Ahmad established a nonlinear model using an ANN to determine the influence of temperature and impeller speed on large-scale production of enzymatic polymers [20]. Studies have proved that the RF and ANN are suitable for, and have made significant contributions to, organic fields.

This paper proposes a feasible ML solution to the problem of small sample size in the bio-polymerization process. To avoid overfitting, the VAE and GAN algorithms are used for data augmentation. The RF and ANN algorithms are implemented in the modeling process.

2. Data and methods

2.1. Data collection

The experimental data used in this study were provided by researchers from the University of Nottingham Malaysia Campus. This dataset, which records several conditions of the bio-polymerization process, includes 42 instances and three parameters: temperature (°C), time (hours), and molecular weight (g mol^{-1}). The current study focuses on using only one solvent, namely toluene, and one particular monomer concentration. More details about the experimental work can be found in Ref. [21]. During the experiment, a specific range of values (temperature 1–100 °C, time 1–7 h) was selected for each parameter, and a control experiment was designed to measure the molecular weight of the polymer under various conditions. Because the number of data instances was insufficient to construct a robust ML model, the data augmentation methods mentioned above were used to enlarge the dataset before the ML models were established. To determine whether the VAE and GAN could also improve the performance on regression problems, the original dataset was also used to construct models directly, as the base case. The programming language used in this study was Python 3.8. Panda, TensorFlow 2.0, and the scikit-learn library were used for data preprocessing and modeling.

2.2. Data preprocessing

2.2.1. Data normalization

Data were normalized to remove dimensional effects before analysis. Normalization enabled all parameters to be mapped to the homogeneity level. In addition, data normalization contributed to the stable convergence of weight and bias. This allowed the efficiency of the network to be improved. For each independent variable data X, the standardized Z-score was adopted, as defined in Eq. (1).

$$x_i^* = \frac{x - \mu}{\sigma} \tag{1}$$

where x_i^* represents the original value of each variable, and μ and σ represent the mean and standard deviation of each parameter, respectively.

2.2.2. Data division

Dividing the dataset into several subsets is critical when constructing a model. The original dataset is divided into three subsets, namely the training set, validation set, and test set. The training set participates directly in the training process, which is used to establish models and modify parameters. The validation set is used to adjust the hyperparameters of the model and prevent the model from overfitting. The test set is used to evaluate the performance and robustness of the model [17].

In this study, the normalized dataset was divided into the original training set (80%) and test set (20%) in the preprocessing

stage. For the model trained directly with the original data, 80% of the training set was then selected as the final training set. The remaining 20% was used as the validation set [17,18]. Conversely, for the model trained after data augmentation, the original training set was expanded by the VAE and GAN. Subsequently, 80% of the new training set was used as the final training set. The remaining 20% of the new training set was used as the validation set. The purpose of this was to ensure that the test data would never participate in the training process. The model design framework is shown in Fig. 1.

2.3. Data augmentation

2.3.1. Development of VAE

The VAE is a widely used data augmentation technique. It is divided into two parts: an encoding network and a decoding network. The former outputs the parameters of the Gaussian distribution, and the latter reconstructs and decodes the ANN according to the distribution of input data. The VAE first assumes that the data obey the Gaussian distribution. The encoding network then trains a probability distribution, namely a latent space, that maps the original probability distribution with the training set. Finally, the decoding network generates more data by selecting the data points on the created distribution and adding noise. After sampling by decoding the ANN, the obtained data obey the prior distribution. When generating data, the logarithmic maximum likelihood method is used to generate parameter estimates of the model. The training structure of the VAE is shown in Fig. 2.

In this study, the encoding and decoding network contain four hidden layers and one output layer, respectively. The hidden layers change the parameters of the weights for each neuron through a constant iterative training algorithm, which makes the network generate the values of each parameter following the distribution of the actual values. The loss function of this model consists of two parts: decoder loss and encoder loss. The former is the loss value from refactoring data, and the latter is the Kullback–Leibler divergence. This model finally constructs the optimal dataset by minimizing the loss function. Table 1 presents the parameters of the VAE iterative training.

2.3.2. Development of GAN

The GAN is currently one of the most promising approaches for measuring complex distributions. This model combines two specific models: the generative model and the discriminative model. The GAN generates data through the interaction of these two networks. The generative model maps data to a latent space and creates new data instances and the discriminative model evaluates the authenticity of these data; that is, it determines whether these generated data are from the original data distribution. In contrast to the VAE, it is not necessary to assume that data follow the Gaussian distribution. The GAN optimizes the generated data by using a discriminator network so that the distribution of data directly fits the distribution of the training data. The training structure of the GAN is shown in Fig. 3.

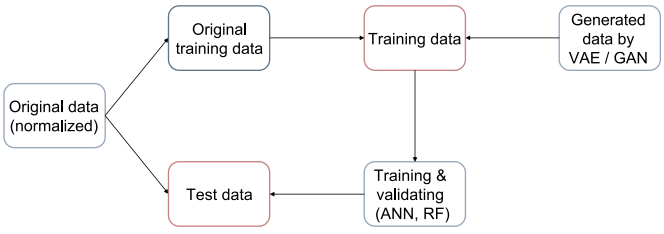

Fig. 1. Framework of experimental design.

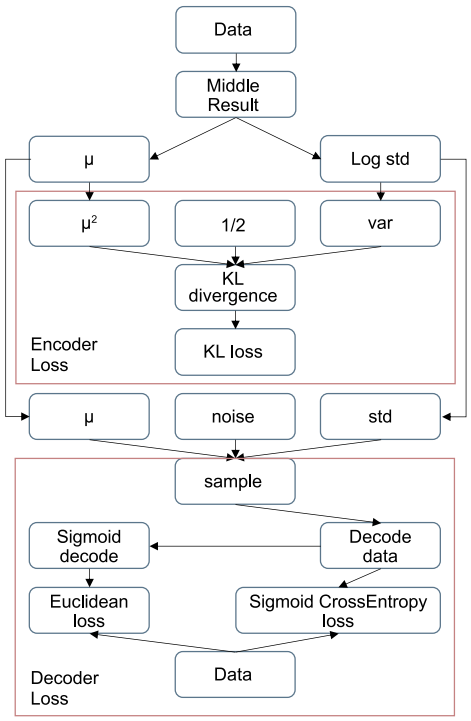

Fig. 2. Structure diagram of VAE.

Table 2 presents the parameters of the iterative training of the GAN used in this study. After the dataset has been augmented, the generated data with the minimum loss function are saved for training. The statistical and ML algorithms mentioned above are then used to construct prediction models on the new training dataset.

2.4. Data modeling

2.4.1. Development of ANN

An ANN usually consist of one input layer, one or more hidden layers, and one output layer. The layers are connected by various numbers of interconnected nodes (neurons), and each node represents a specific output function (or activation function). A connection between two nodes represents a specific weighted value for each signal, namely the weight. The ANN finally obtains the output by calculating the weight of each neuron [22]. The ANN can obtain the relationship between inputs and outputs by

Table 1
Parameters of the VAE.

Parameter	Value
Augmented data	80%
Encoder layers	4 hidden layers, 1 output layer
Dense	11,22,22,22
Activation (hidden layers and input layer)	Leaky ReLU
Batch normalization	0.99
Decoder layers	5 hidden layers, 1 output layer
Dense	22,22,22,22
Activation (hidden layers and input layer)	Leaky ReLU
Batch normalization	0.99
Loss function	decoder loss and kl loss
Epochs	3000
Batch size	36
Optimizer	Adam

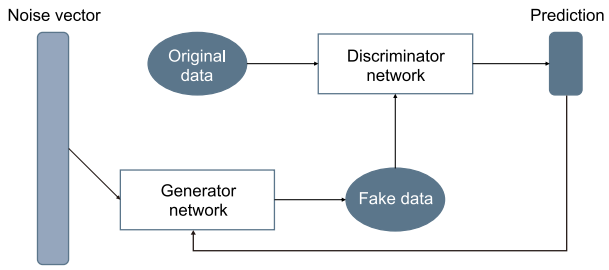

Fig. 3. Structure diagram of GAN.

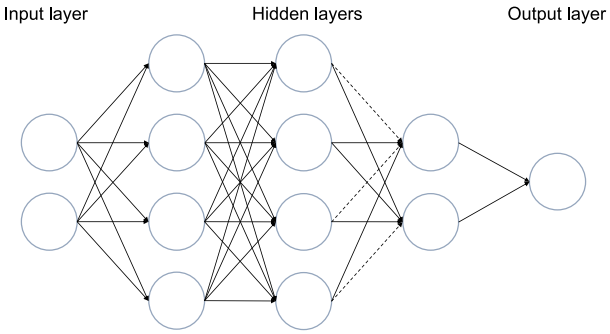

Fig. 4. Structure diagram of ANN.

Table 2
Parameters of the GAN model.

Parameter	Value
Augmented data	80%
Generator	2 hidden layers, 1 output layer
Dense	512, 1024
Activation (hidden layers and input layer)	Leaky ReLU
Batch normalization	0.8
Decoder layers	2 hidden layers, 1 output layer
Dense	512, 256
Activation (hidden layers and input layer)	Leaky ReLU
Batch normalization	0.8
Loss	Binary_crossentropy
Epochs	8000
Batch size	3
Optimizer	RMSprop

assigning weights to each network neuron. Moreover, it optimizes the model by changing the weights and applying various training algorithms to reduce errors. The number of neurons in input layers is determined by the experimental design parameters, and the number of neurons in the output layer is determined by the experimental analysis result. In the training stage, the input and output data participate in the training process so that the ANN can find the functional relationship between them by calculating the weight. Various training algorithms can be applied in the training process to modify the weights and train the network to perform a given task. The structure of an ANN is shown in Fig. 4.

In this study, the input layer consists of two neurons, corresponding to the two parameters that are to be trained as input (temperature and time in this case). The output layer contains a single neuron: the molecular weight. The activation function of each hidden layer is nonlinear (ReLU), and that function of the output layer is linear. Various numbers of hidden neurons (between 1 and 20) are used in the hidden layer to train and compare the results. The purpose of this is to find the optimal number of neurons in hidden layers, which strongly affects the weights of parameters. That is, the use of too few neurons is not conducive for the model to learn internal laws (underfitting), whereas using too many neurons makes the model more reliant on the training data but unable to generalize (overfitting). In addition, GAN and VAE data may require different optimal numbers of neurons. Therefore, the number of hidden neurons is adjusted separately in the two models. Moreover, the 'dropout' function is used to keep some neurons disconnected, to make the model less dependent on some local features. L2 regularization is also used: this function reduces the weight of each neuron (by weight decay) so that the model can converge better than without this function. Table 3 presents the parameters of the ANN iterative training.

2.4.2. Development of RF

RF originated from the bagging algorithm. In the training stage, RF uses bootstrap sampling to sample multiple sub-training sets from the input training dataset according to the attributes. It then uses these subsets to train multiple decision trees. Because different decision trees may output different predicted values, RF averages the prediction results of multiple internal decision trees to obtain the final prediction results in the prediction stage. As shown in Fig. 5, RF determines the results by a majority vote of the multiple decision trees.

When constructing the RF, the polynomial features were enhanced before training to ensure that the model could correctly respond to nonlinear functions [17]. In addition, various different numbers of trees were applied in this model to find the optimal parameters. Table 4 presents the parameters of the RF.

3. Results and discussion

3.1. Model optimization

The parameters of the models have a significant influence on the prediction results. Therefore, the parameters of each model need to be adjusted after the initial establishment of the model to obtain the optimal model. In addition, to evaluate the performance of the model in the training process, it is necessary to select an appropriate evaluation metric and loss function. This study evaluated the performance of models using mean square error (MSE), which provides the criterion for the model to find the optimal weight and bias by minimizing the MSE. The metric of the training model is the coefficient of determination (R^2), which does not affect the parameters of the model but is a method of evaluating each model [23]. The MSE and R^2 are defined in Eq. (2) and Eq. (3), respectively:

$$MSE = \frac{1}{n}\sum_{x=0}^{n}(P_x - A_x)^2 \quad (2)$$

Table 3
Parameters of the ANN.

Parameter	Value
Network	FANN
Training data	Augmented data
Test data	Original testing data
Hidden layers	3
Hidden neurons	16
Epochs	1000
Activation (hidden layers and input layer)	ReLU
Activation (output layer)	Linear
Loss	MSE

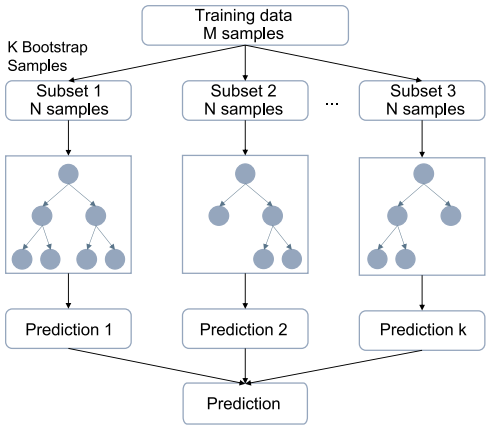

Fig. 5. Structure diagram of RF.

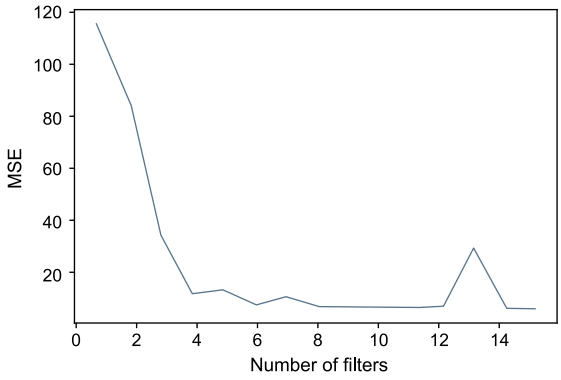

Fig. 6. Variation of MSE with the number of filters.

$$R^2 = 1 - \frac{\sum_{x=1}^{n}(P_x - A_x)^2}{\sum_{x=1}^{n}(P_x - \sigma_x)^2} \quad (3)$$

where P_x is the experimental value of the xth experiment, A_x is the actual value, and σ_x is the average of each experimental value.

3.1.1. VAE optimization

It is necessary to select and adjust the number of filters properly because VAE is sensitive to filters. If there are too few filters, it is possible for the models to extract insufficient information, which affects the loss value. However, an excessive number of filters results in redundancy and wastes time. In this study, the number of filters was adjusted through the loss function to obtain the most appropriate number. The number of filters was varied from 1 to 15, and the lowest loss value was recorded for each run.

Fig. 6 shows that, overall, the MSE decreased as the number of filters increased. It dropped quickly until the number of filters reached 4, indicating that the performance of the model rapidly improved with the increase in the number of filters. However, the MSE reached its minimum (3.04) with 11 filters and did not decrease or even increase with more filters. Therefore, the number of filters was set to 11 to optimize the VAE. Table 5 shows 14 of the 1000 data instances generated by the VAE.

Fig. 7 shows the histograms for each parameter generated by the VAE. As can be observed, the expanded data closely conform to the normal or skewed distribution, and this is consistent with the prior distribution of the VAE algorithm.

3.1.2. GAN optimization

According to previous work [24], the training of the GAN is usually challenging. It contains two network models. Thus, there are two loss functions. Usually, in image classification problems, generated datasets are evaluated by direct observation of results, but this is impossible in the case of regression problems. In this study, the optimal data generated by the GAN were determined by considering the distribution of data instances and loss values of the generative network and discriminative network. Table 6 shows 14 of the 1000 data instances generated by the GAN.

Fig. 8 shows the distribution of each generated parameter, indicating that the extended dataset approximately satisfies the requirement of symmetric distribution. It is worth mentioning that, although the VAE and GAN efficiently created new sample instances based on the distribution, it was difficult to compare these two models directly because the criteria used to assess them are not consistent. Therefore, the performance of the VAE and GAN was compared by establishing regression models on two generated datasets.

3.1.3. ANN optimization

It is essential to choose the appropriate number of neurons in the hidden layer of the ANN. If there are too few neurons, the model may not be able to learn enough information. In contrast, an excessively large number of neurons may cause overfitting. In this study, the number of hidden neurons used for the training model was continuously corrected and evaluated by comparing the loss values of MSE and R^2, as shown in Fig. 9.

Fig. 9 shows that the MSE of the ANN constructed with the

Table 4
Parameters of the RF.

Parameter	Value
Polynomial features	2
Training data	Augmented data
Test data	Original testing data
N_Estimators	50
Criterion	MSE
Random_State	None
Max_Depth	None
Max_Features	Auto

Table 5
Data instances generated by the VAE.

Time	Temperature	Molecular wt
2.72	71.0	11330.52
2.70	71.3	11272.89
2.68	71.5	11210.73
2.65	71.7	11160.72
3.13	63.3	12961.22
3.11	63.6	12919.06
3.10	63.8	12876.9
3.09	64.1	12834.74
3.07	64.4	12783.07
3.05	64.7	12723.4
3.04	64.9	12663
3.02	65.2	12602.6
3.01	65.5	12542.2
2.99	65.7	12478.3
2.97	66.0	12409.19
2.96	66.3	12338.31

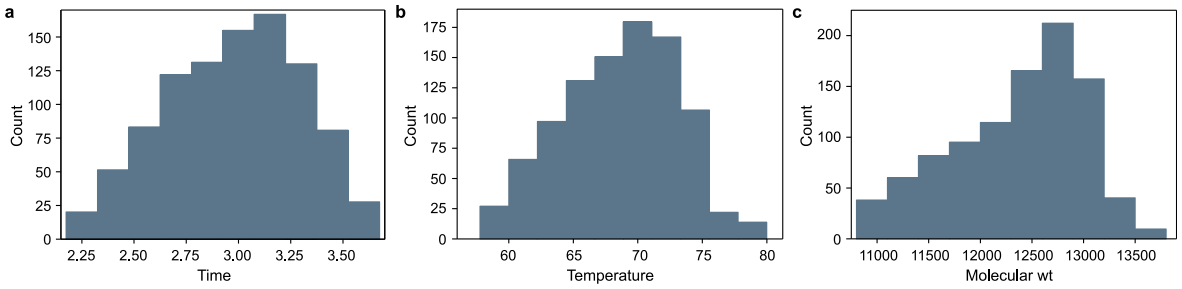

Fig. 7. Histograms of time (**a**), temperature (**b**), and molecular weight (**c**) generated by the VAE.

Table 6
Data instances generated by the GAN.

Time	Temperature	Molecular wt
1.99	59.3	9426.629
2.00	92.2	23025.64
2.12	65.5	12912.92
2.05	90.4	9323.119
2.12	59.8	16359.54
4.43	57.9	17731.7
5.94	62.7	8732.383
2.24	63.0	13265.61
1.99	92.1	14100.91
5.91	60.4	9269.199
1.98	61.1	15802.4
3.27	91.2	22025.82
6.02	57.8	9243.063
5.88	92.3	25864.12

original data decreased as the number of neurons increased, and started to converge at 16 neurons, when the MSE value was 0.4. The MSE of the VAE + ANN first decreased as the number of neurons increased, but it fluctuated noticeably when the number of neurons reached 4. Subsequently, the MSE continued to decrease and started to converge until the number of neurons reached 16. The MSE of the GAN + ANN decreased continuously until the number of neurons reached 11 but it did not change significantly as the number increased further. Moreover, the R^2 of the ANN constructed with the original data, VAE + ANN, and GAN + ANN all increased precipitously up to three neurons and then levelled off, indicating that the model could not learn enough knowledge when the number of neurons was less than 3. After adding more neurons, the R^2 of the ANN with the original data and the VAE + ANN continued to increase and peaked at 16 neurons, but did not change dramatically after that. The R^2 of the GAN + ANN also increased until the number of neurons reached 11. Therefore, the ANN with the GAN data augmentation algorithm (GAN + ANN) achieved the best performance with 11 neurons, when the MSE and R^2 were 0.41 and 0.60, respectively. The best performance for the VAE + ANN was achieved with 16 neurons, when the MSE and R^2 were 0.26 and 0.73, respectively.

Fig. 10 shows the evaluation curves of the training set and validation set for the two proposed models, after training for 1000 epochs. As shown in Fig. 10a, the MSE of the ANN trained with the original data decreased steadily during training until about 800 epochs. Therefore, the optimal model was determined when trained for 800 epochs, when the minimum MSE values of the training set and validation set were 0.47 and 0.17, respectively. As shown in Fig. 10b and c, for the VAE + ANN, the R^2 of the training set increased throughout the training process because the model attempted to improve the accuracy of prediction continuously. Therefore, the validation set is particularly important because it adjusts the hyperparameters of the model and ensures that the model will not overfit by estimating the performance of the model in the validation set. Fig. 10b shows that the R^2 of both the training set and validation set increased during the first 400 epochs. However, this value for the validation set started to decrease after more epochs were executed, indicating that the fitting to the training data gradually lost generalization. Therefore, the optimal model was determined when trained for about 400 epochs, when the optimal MSE values of the training set and validation set were 0.17 and 0.29, respectively, and the best R^2 values of the training set and validation set were 0.71 and 0.84, respectively.

Fig. 10d and e also show that the R^2 values of the training set and validation set increased precipitously in the first few epochs for the GAN + ANN. However, R^2 did not change significantly when 500 epochs were executed. In addition, the MSE of both sets converged to a minimum at 500 epochs. Therefore, the optimal model was determined when trained for 500 epochs, when the optimal MSE values of the training set and validation set were 0.43 and 0.48, respectively, and the best R^2 values of the training set and validation set were 0.59 and 0.63, respectively.

3.1.4. RF optimization

The performance of RF is also affected by the "number of trees" and "number of attributes" parameters. However, because there are only two input parameters in the dataset used for this study, the influence of the attributes was not particularly significant. Therefore, only the influence of the number of trees was considered for

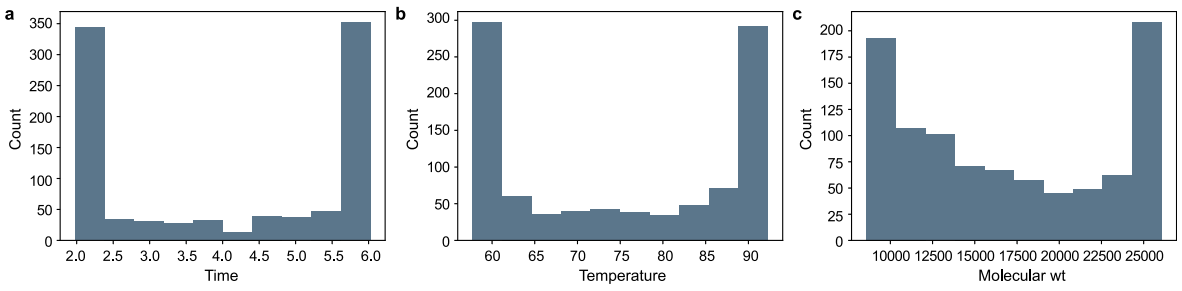

Fig. 8. Histograms of time (**a**), temperature (**b**), and molecular weight (**c**) generated by the GAN.

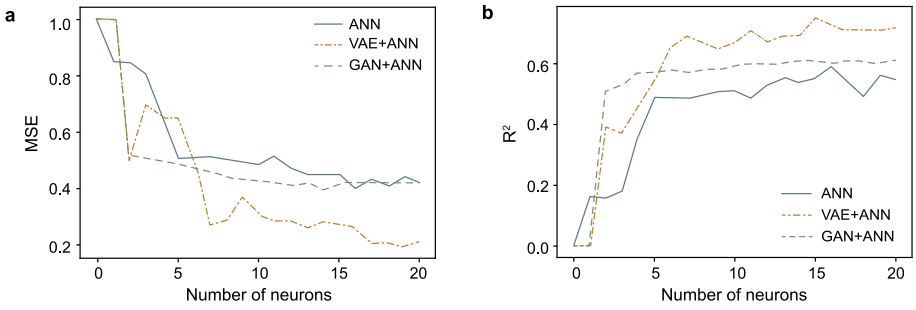

Fig. 9. Variation of MSE (**a**) and R^2 (**b**) with the number of neurons.

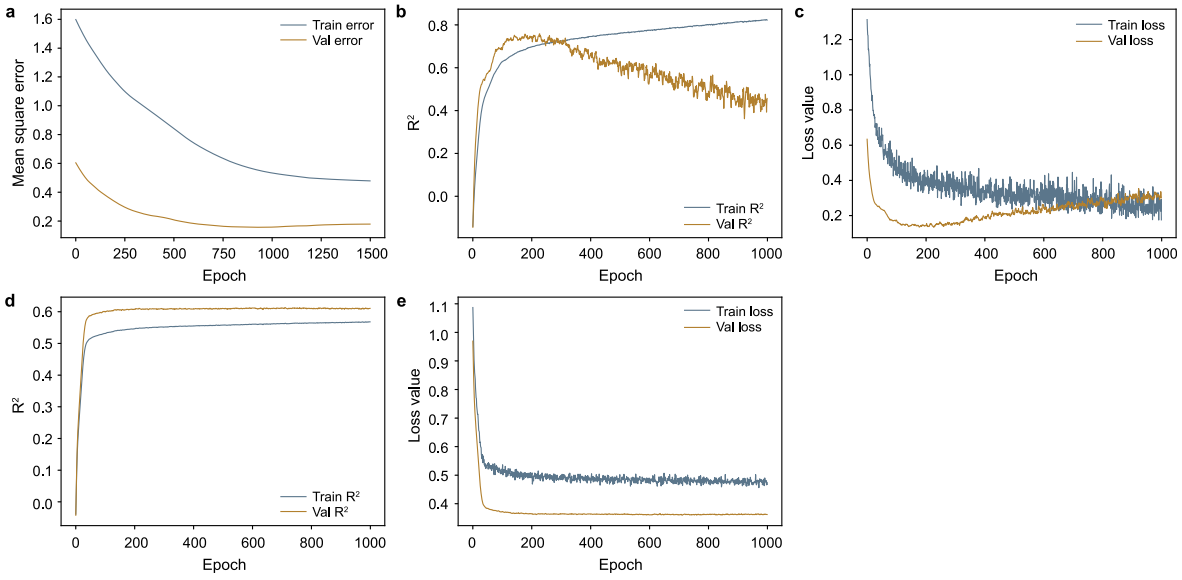

Fig. 10. Evaluation curves: **a,** R^2 of ANN; **b**, R^2 of VAE + ANN; **c**, MSE of VAE + ANN; **d**, R^2 of GAN + ANN; **e,** MSE of GAN + ANN.

optimizing the RF. Fig. 11 shows the effect of modifying the number of trees on the training set.

As shown by Fig. 11, the R^2 of the RF trained with the original data was maximized when the number of trees was 100 (0.89). Because few data were used for training, 100 trees seemed to be sufficient to train the RF well. In addition, the R^2 of the RF with the VAE augmentation technique (VAE + RF) reached its maximum value (0.94) when the number of trees was 200. The RF with the GAN augmentation technique (GAN + RF) reached its maximum value (0.94) when the number of trees was 150. Therefore, the optimal numbers of trees of the VAE + RF and GAN + RF were determined to be 200 and 150, respectively.

3.2. Comparing the performance of models on the test set

ML models using various algorithms have been established, and the parameters of each model have been optimized. In this section, the ability of each model to predict unknown data is discussed. For this purpose, the test data were used to test the robustness of the models. Each model was applied to predict the molecular weight of the test set. The predicted values were then compared with the actual values to test the performance and robustness of the models, as shown in Fig. 12. To select the optimal model, the errors in both the training stage and test stage were considered. The purpose of this was to avoid the predicted contingency caused by a small test set. A good model is expected to perform well on the training set first and generalize well to the test set. The R^2 values of each model on the training set and test set are shown in Table 7.

As can be seen in Table 7, the ANN and RF built with original dataset have poor performance on both training set (with R^2 of 0.57 and 0.89, respectively) and testing set (with R^2 of 0.77 and 0.60, respectively). Fig. 12a and b also show the predicted results of the ANN and RF comparing with the actual values, respectively,

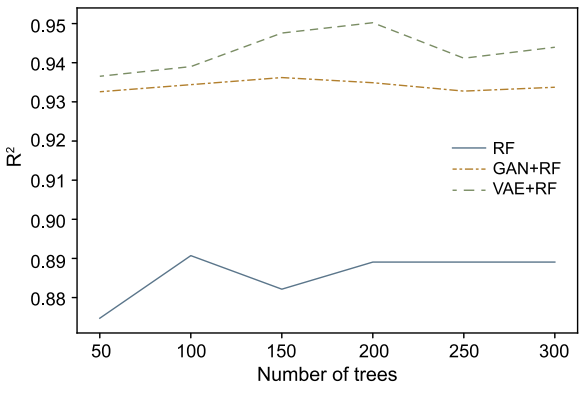

Fig. 11. Variation of R^2 with the number of trees.

constructed with the original dataset. Obviously, there is a significant difference between the predicted and actual values of the ANN. The predicted molecular weights are mostly greater than the actual values. The difference between actual and predicted values is also evident in the RF. In general, the models based on the original dataset performed poorly, mainly because of the small amount of data. In addition, these models did not summarize the general rules in the limited amount of data because of the significant difference between the data instances.

Table 7 also show that the ANN and RF augmented by VAE significantly increases the values of R^2 on training (0.71 and 0.94, respectively) and testing set (0.85 and 0.51, respectively), which indicated that the performances of models are improved. Fig. 12c and d show the molecular weights predicted by the ANN and RF, respectively, where the VAE was used for data augmentation. These figures show that the ANN and RF predicted relatively accurate molecular weights, but some of the predicted values differ significantly from the actual values. This is because the VAE has a condition that conforms the prior distribution and the features of generated data will be similar. Therefore, the predicted values of outliers are not ideal. Basically, the values predicted by the ANN are close to the actual values. Therefore, it is considered that the ANN performed well and demonstrated good robustness in this study. In contrast, the RF performed poorly on the testing set, indicating that RF is not sufficiently robust. Comparing these models, the ANN performed better than the RF when the VAE was applied to enlarge the dataset.

Fig. 12e and f show the predicted and actual molecular weights where GAN was used for data augmentation. The figure shows that the ANN roughly predicted the actual values but was not sensitive to the extreme values. That is, the ANN successfully predicted similar values when the actual molecular weights were close to the average but excessively high values caused obvious prediction errors. Therefore, the ANN with the GAN was not sufficiently robust in this study. The predicted values of the RF were much closer to the real values, and this model achieved satisfactory performance in both the training and testing stages. Therefore, it is reasonable to regard RF as the better model when the GAN was used for data augmentation.

Table 7
R^2 on the training set and test set.

	Original		VAE		GAN	
	ANN	RF	ANN	RF	ANN	RF
Train R^2	0.57	0.89	0.71	0.94	0.59	0.94
Test R^2	0.77	0.60	0.85	0.51	0.63	0.74

Taking every model into consideration, both the ANN and RF achieved ideal performance in this study because they could fit more complex functions by continuous learning. This means that these models could eliminate the influence of error caused by extreme values more significantly by constantly adjusting weights by normalization and training algorithms. Moreover, the ANN with VAE augmentation performed stably, compared with other ANN, because VAE generated a model based on the original distribution, causing the created data to have strong regularity. However, the ANN with GAN performed poorly because it was difficult to fit the distribution of the original data. In addition, the RF, with both augmentation models, seemed to achieve more satisfactory performance. The RF with the VAE model achieved the best R^2 in the training stage (0.95), but its performance on the test set (0.51) indicated that this model was not sufficiently robust. The RF with the GAN fitted well on the training set (with an R^2 of 0.94) and accurately predicted the actual values on the test set (with an R^2 of 0.74). Therefore, the RF with the GAN is considered to be the best model.

4. Conclusions

In this study, the data augmentation method and ML modeling techniques for solving regression problems on a small dataset were investigated. The results prove that data augmentation techniques effectively improved the performance of the regression model. In general, the performance of each variant of the RF was better than that of the ANN for this study. Among all the models, GAN data

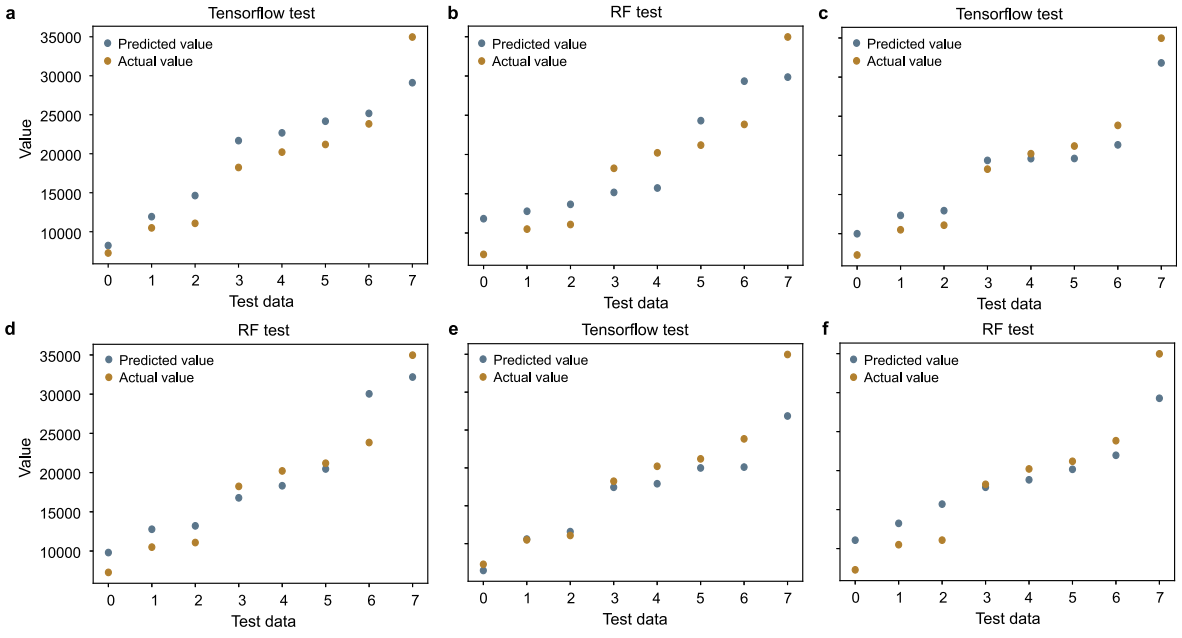

Fig. 12. Predicted values vs. actual values: a, ANN; b, RF; c, VAE + ANN; d, VAE + RF; e, GAN + ANN; f, GAN + RF.

augmentation combined with the RF is considered to be the optimal model.

This paper proposes a scientific method that could be applied to ML modeling when handling insufficient data; the method avoids overfitting so that a more robust model can be constructed. This enables the bio-polymerization process to be optimized using the minimum amount of resources, and hence contributes to improved environmental performance. However, it is worth noting that the models constructed in this study are not yet perfect, mainly because the data used have a small numbers of attributes. Although the VAE and GAN improved the performance of models by generating more data instances, the generated data were mostly similar because few attributes were taken into consideration. Therefore, only limited improvement of the model was achieved. If a dataset including multiple attributes is used in future work, the data augmentation technique will greatly improve the performance of models.

Although some models with ideal performance have been constructed, there is still scope for further improvement. In VAE, the loss function did not converge to an ideal value (the minimum value was 3.04). There is no doubt that minimization of the loss function can improve the efficiency of the prediction model, and this could be further improved in future research.

In some previous research, several other parameters—such as the use of different solvents and monomer concentrations—have been taken into consideration. In contrast, the current study focused on only one solvent, namely toluene, and one particular monomer concentration. The reaction conditions collected in the dataset have only one typical lactone or polyester monomer, namely e-caprolactone, and only one typical enzyme, namely candida antartica lipase B. However, the findings will be very useful because the prediction model developed can be used for larger-scale case studies in industry. The model can be used for process control where there is a need for a model-based control strategy. This means that, when the polymer PCL of a specific molecular weight is required, the corresponding temperature and time period can be optimized without actually conducting the experiment. Hence, the developed model can be used as a benchmark for developing similar models for other reaction systems and for systems with different lactones and enzymatic systems.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Abbreviations

ML:	Machine learning
VAE	Variational auto-encoder
GAN	Generative adversarial network
ANN	Artificial neural network
RF	Random forest
MSE	Mean square error
R^2	Coefficient of determination
PCL:	Polycaprolactone

References

[1] J.M. Asua, Miniemulsion polymerization, Prog. Polym. Sci. 27 (7) (2002) 1283–1346.
[2] M.R. Islam, M.D.H. Beg, S.S. Jamari, Development of vegetable-oil-based polymers, J. Appl. Polym. Sci. 131 (18) (2014).
[3] S. Mohan, O.S. Oluwafemi, N. Kalarikkal, S. Thomas, S.P. Songca, Biopolymers–application in nanoscience and nanotechnology, Recent Adv. Biopolym. 1 (1) (2016) 47–66.
[4] M. Rusu, M. Ursu, D. Rusu, Poly (vinyl chloride) and poly (e-caprolactone) blends for medical use, J. Thermoplast. Compos. Mater. 19 (2) (2006) 173–190.
[5] R.G. Schoenmakers, P. Van De Wetering, D.L. Elbert, J.A. Hubbell, The effect of the linker on the hydrolysis rate of drug-linked ester bonds, J. Contr. Release 95 (2) (2004) 291–300.
[6] A.C. Albertsson, I.K. Varma, Recent developments in ring opening polymerization of lactones for biomedical applications, Biomacromolecules 4 (6) (2003) 1466–1486.
[7] R.A. Greinke, L.H. O'connor, Determination of molecular weight distributions of polymerized petroleum pitch by gel permeation chromatography with quinoline eluent, Anal. Chem. 52 (12) (1980) 1877–1881.
[8] M.I. Jordan, T.M. Mitchell, Machine learning: trends, perspectives, and prospects, Science 349 (6245) (2015) 255–260.
[9] W.M. Chan, D.V.K. Le, Z. Chen, et al., Resource allocation in multiple energy-integrated biorefinery using neuroevolution and mathematical optimization, Process Integr. Optim. Sustain. 5 (2021) 383–416.
[10] S.K. Arumugasamy, Z. Chen, L.D. Van Khoa, et al., Comparison between artificial neural networks and support vector machine modeling for polycaprolactone synthesis via enzyme catalyzed polymerization, Process Integr. Optim. Sustain. 5 (2021) 599–607.
[11] L. Perez, J. Wang, The Effectiveness of Data Augmentation in Image Classification Using Deep Learning, 2017 arXiv preprint arXiv:1712.04621.
[12] F.J. Moreno-Barea, J.M. Jerez, L. Franco, Improving classification accuracy using data augmentation on small data sets, Expert Syst. Appl. 161 (2020), 113696.
[13] U. Garay-Maestre, A.J. Gallego, J. Calvo-Zaragoza, Data augmentation via variational auto-encoders, in: Iberoamerican Congress on Pattern Recognition, Springer, Cham, 2018, November, pp. 29–37.
[14] X. Liu, S. Guo, H. Zhang, K. He, S. Mu, Y. Guo, X. Li, Accurate colorectal tumor segmentation for CT scans based on the label assignment generative adversarial network, Med. Phys. 46 (8) (2019) 3532–3542.
[15] H. Ohno, Auto-encoder-based generative models for data augmentation on regression problems, Soft Comput. 24 (11) (2020) 7999–8009.
[16] M. Rezagholiradeh, M.A. Haidar, Reg-gan: semi-supervised learning based on generative adversarial networks for regression, in: 2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), IEEE, 2018, April, pp. 2806–2810.
[17] A. Liaw, M. Wiener, Classification and regression by randomForest, R. News 2 (3) (2002) 18–22.
[18] X. Zhou, X. Zhu, Z. Dong, W. Guo, Estimation of biomass in wheat using random forest regression algorithm and remote sensing data, Crop J. 4 (3) (2016) 212–219.
[19] V.T. Lipik, M.J. Abadie, Process Optimization of Poly (ε-Caprolactone) Synthesis by Ring Opening Polymerization, 2010.
[20] S.K. Arumugasamy, M.H. Uzir, Z. Ahmad, Neural network based modeling for polycaprolactone synthesis by bio-polymerization of [epsilon]-caprolactone, Int. J. Biosci. Biochem. Bioinf. 3 (1) (2013) 56.
[21] S.K. Arumugasamy, Z. Ahmad, Candida Antarctica as catalyst for polycaprolactone synthesis: effect of temperature and solvents, Asia Pac. J. Chem. Eng. 6 (3) (2011) 398–405.
[22] P.S. Pauletto, S.F. Lütke, G.L. Dotto, N.P.G. Salau, Forecasting the multicomponent adsorption of nimesulide and paracetamol through artificial neural network, Chem. Eng. J. (2020), https://doi.org/10.1016/j.cej.2020.127527.
[23] E. Betiku, V.O. Odude, N.B. Ishola, A. Bamimore, A.S. Osunleke, A.A. Okeleye, Predictive capability evaluation of RSM, ANFIS and ANN: a case of reduction of high free fatty acid of palm kernel oil via esterification process, Energy Convers. Manag. 124 (2016) 219–230.
[24] K.J. Liang, C. Li, G. Wang, L. Carin, Generative Adversarial Network Training Is a Continual Learning Problem. arXiv Preprint arXiv:1811.11083, 2018.