Received: March 13, 2023. Revised: May 30, 2023.

International Journal of
Intelligent Engineering & Systems

http://www.inass.org/

Improving Conditional Variational Autoencoder with Resampling Strategies for Regression Synthetic Project Generation

Robert Marco[1]* **Sharifah Sakinah Syed Ahmad[2]** **Sabrina Ahmad[2]**

[1]*Department of Informatics, Universitas Amikom Yogyakarta, Yogyakarta, Indonesia*
[2]*Faculty of Information & Communication Technology, Universiti Teknikal Malaysia Melaka, Malaysia*
*Corresponding author's Email: *robertmarco@amikom.ac.id*

Abstract: The uncertainty inherent in the predictive environment of software effort estimation, is due to the limited availability of data information and the expensive costs associated with data collection. Consequently, there is difficulty in making accurate predictions with insufficient software effort estimation data, due to the limited information in the data. The probabilistic deep generative model conditional variational autoencoder (CVAE) is capable of producing synthetic data and modeling complex data distribution characteristics that is similarity to the real Unfortunately, because the method was previously developed for the classification task, the application of the technique to the regression task has received little attention. This study aims to construct a CVAE model combined with the relevance function contained in the re-sampling method. Relevance function method aims to create a label on the target. Statistical tests, such as the Levene test, t-test, and Kolmogrove Smirnov test, are utilized to compare the results to the real data. This study considers seven popular algorithms for comparison, such as synthetic minority oversampling technique regression (SMOTER), generative adversarial networks (GAN), conditional tabular generative adversarial networks (CTGAN), gaussian-copula (GC), copula GAN (CGAN), and variational autoencoder (VAE). CVAE approach has the best augmentation performance quality with accuracy values (MAE; RMSE; RAE; R^2) compared to existing models in each china dataset (923.330; 1470.536; 0.436; 0.848) and desharnais dataset (9.540; 11.694; 0.041; 0.978), respectively. The proposed model generates synthetic data which is similarity to real data in the software effort estimation context. In addition, synthetic data improves the quality of performance on baseline machine learning in software effort estimation contexts rather than using real data.

Keywords: Conditional variational autoencoder, Re-sampling, Relevance function, Synthetic project generation, Software effort estimation.

1. Introduction

Uncertainty and imprecision are inherent in the predictive environment of software effort estimation (SEE) [1]. This is due to several factors hindering its practical use, which most researchers still ignore in SEE [2]. Such as the limited availability of data information and the expensive costs associated with data collection in the process of building the SEE model [3, 4]. Consequently, organizations often have a small number of completed projects to estimate the new project effort [3]. As a result, this can lead to difficulties in making accurate predictions with insufficient SEE data, due to the limited information available in the data it may not be sufficient to support the training of a model based on the SEE method [5]. Because, the use of a small training sample can have an impact on the learning method's performance [6].

The data augmentation (DA) technique is the process of completing a data collection with similar data (synthetic dataset) generated from the information in that data set [7]. In various prior research, one of the classic methods that are often used as data augmentation in overcoming small data in several previous studies, like synthetic minority oversampling technique (SMOTE) [8, 9], gaussian copula (GC) [10, 11], multivariate imputation by chained equations (MICE) [12, 13], random forest

(RF) [14], support vector machine (SVM) [15], and classification and regression tree (CART) [16].

When there are a number of samples available, the SMOTE oversampling method is particularly likely to be the one that is utilized the most [17]. However, SMOTE has trouble identifying what constitutes a minority value and what constitutes a majority value for regression [3, 18], and SMOTE is susceptible to problems when generating noise and boundary samples [19]. In the meantime, the gaussian copula generates synthetic data and evaluates its utility, concluding that it is expensive, esoteric, and convoluted [11]. MICE, on the other hand, is unsuitable for drawing from shared distributions when the conditional models are incompatible with data sets containing multiple variable types [20]. The CART model has problems with interpretation, a discontinuity at the boundaries of the partitions, and decreasing effectiveness when the relationships are correctly defined by the parametric model [16]. In contrast, as the number of variables to be synthesized increases in the RF model, the computation time increases approximately linearly [14]. Although, the possible increase in data utility can be achieved using SVM, however it is computationally expensive and can still lead to overfitting [15].

There are two popular synthetic data generation techniques in generative models based on the neural networks method, like generative adversarial networks (GAN) [18] and variational autoencoder (VAE) [21]. It has gained tremendous popularity due to its superiority in capturing complex data distributions [22, 23]. The use of GANs to generate data is becoming more popular in the core machine learning community, it requires multiple models to train, which causes difficulty and computational burden in finding optimal model parameters [24]. Meanwhile, the VAE method makes strong distributional assumptions, which can undermine the generative model [24].

Kingma et al. (2014) created the conditional variational autoencoder (CVAE), which stands for VAE, to overcome these shortcomings [25]. CVAE may reconstruct input characteristics utilizing output vectors [26]. CVAE can generate new samples automatically from predefined categories and extract high-order features, on the other hand it can reduces the dimensionality of network features automatically [27]. Probabilistic deep generative models CVAE are capable of modeling complex data distributions [27]. Using a gradient-based method makes training much easier [28], is effective at capturing the characteristics of the real sample distribution, and can make similar synthetic samples [29], and has a superior convergence property [30].

CVAE successfully extracted prospective features with a high learning ability using an encoder. The decoder reconstructs the input features and provides sufficient data to the deep neural network [26]. Unfortunately, since previous methods were made for classification tasks, applying the technique to regression tasks has received little attention. In the regression data, determining a standard sample is tricky. Ultimately, this classification issue is intuitively solvable. In contrast, in the regression problem, there is difficulty in setting an appropriate target value for the resulting synthetic data (continuous data). The relevance function method in re-sampling regression aims to create a label on the target $\phi(Y): y \rightarrow [0,1]$, where 0 denote minimum and 1 denote maximum relevance. This is because CVAE was built for tackled the classification problem. Thus, we changed the target y by labeling [0, 1, 8-9].

We propose the CVAE with relevance function method found in resampling regression, which aims to accurately label targets to model correlations between attributes by introducing additional monitoring tasks to facilitate correlation extraction. The neural network is then trained to conduct an inverse transformation of the generated data into the distribution target for each continuous column. This provides identical synthetic data to the real data in the SEE context. Also, decision boundaries can be chosen appropriately, which can help reduce the overfitting inherent in oversampling and synthetic generative methods.

The remainder of this paper's sections are organized as follows. The section 2 describes related works. The theory of CVAE and relevance function is discussed in section 3. The section 4 describes the design of experiments. The experimental results and discussion can be found in section 5. The section 6 of this paper's conclusion and the pontensial in the future.

2. Related works

Synthetic data could be a promising alternative to deal with the problem of small data sets. It has been the subject of research for almost three decades and has found use in a variety of fields [7, 15, 31]. Much research has been conducted in the past on data addition techniques for enlarging the training set and enhancing training performance [32]. Currently, the machine learning approach has been widely used to data generation [26], and has shown good performance. However, several data augmentation methods have been developed to handle classification [18, 33, 34] and time series [35-37] tasks in

generating synthetic data. Thus there are still limitations in the field of regression [3].

To our knowledge, only two studies in the SEE context have overcome small data to produce synthetic data using data augmentation, e.g. [3, 38].

Kamei et al. (2008), a collaborative version of the augmentation method using the SMOTE and k-nearest neighbors (kNN) approaches aims to change the classification method to regression by attributing class unbalances to the most predictive features of SEE. The data generator is designed to extend to other SEE models as data preprocessors easily [38]. Unfortunately, SMOTE may mistakenly produce synthetic examples attacking majority class decision areas, especially in the case of overlapping classes, which causes some statistical bias [39]. SMOTE has several significant limitations in the sample-generating process [40]. New samples are only created between two samples, and the range of the sampling area is limited, which can easily result in overfitting [41].

Song et al. (2018), the augmentation method using a significant probability approach based on the Gaussian/binomial distribution aims to increase the size of the SEE data set by slightly shifting some randomly selected training examples for use as data preprocessors [3]. While this method is simple and sometimes effective, on the other hand, the synthetic project may introduce noise but only form small variations in the project.

Meanwhile, researchers have used various neural network techniques to generate synthetic data in generative models. Goodfellow et al. (2014), proposed a GAN approach for simultaneously training two models. ReLU and sigmoid activation are among the generator network configurations we employ. On the other hand, the discriminator network uses max_out and dropout activation. The findings demonstrate the mean log-likelihood and standard error accuracy for MNIST (225; 2) and Toronto Face datasets, respectively (2057; 26) [18]. Although the use of GANs for data generation is becoming increasingly widely used within the core machine learning community, it requires neural networks with more and more layers to train, which causes difficulty and computational burden (time-consuming) in finding optimal model parameters [24].

Xu et al. (2019) proposed a conditional tabular generative adversarial networks (CTGAN) by training each model using 500 batch. Each model undergoes 300 epochs of training. Our CTGAN performs marginally better than MedGAN and TableGAN. CTGAN creates realistic synthetic data from tabular data with continuous and discrete columns. Unfortunately, continuous columns can have multiple modes, and discrete columns can be unbalanced, making modeling challenging [42]. On the other hand, the fluctuating loss function of CTGAN and difficulty in converging causes poor performance [43].

Kamthe et al. (2021), propose to use a probabilistic model, namely the copula GAN (CGAN) method, as a synthetic data generator. The results demonstrate that the CGAN method can generate synthetic data with extraordinary precision. However, GAN are frequently difficult to interpret in generated synthetic data. Consequently, the development of the GAN method based on copula theory permits the construction of interpretable and adaptable models for data generation [44]. Unfortunately, the CGAN encountered difficulties during training, which resulted in the loss of no explicit representation that had to be correctly synchronized, making the determination of optimal model parameters difficult and time-consuming.

Wan et al. (2018), producing synthetic data using a variational autoencoder (VAE) is presented as a solution for unbalanced learning. VAE can create new samples similar to those found in the initial dataset but not identical to those samples. According to the findings of our experiments, the proposed method is superior to more conventional approaches to synthetic sampling, such as SMOTE and ADASYN (recall, F1, and specificity) [34]. Meanwhile, the VAE technique makes (strong) distributional assumptions, possibly detrimental the generative model [24].

The CVAE-based time series approach that was developed by Fan et al. (2022) generates high-quality simulated data samples and has an RMSE performance ratio that falls somewhere between 12 and 18 percent. It is necessary to have a larger latent dimension in order to generate synthetic data in order to increase the generalization performance of the model [37]. However, the quality of synthetic data can be enhanced by increasing latent dimensions. On the other hand, the CVAE method is fully connected layers which causes a computational burden in finding the optimal model parameters [37].

While several studies have addressed the use of augmentation methods previously made for the classification task, the application of the technique to the regression task has received little attention. The augmentation method established for classification cannot be applied directly to SEE because there is no minority and majority class in regression. Possibly because of the difficulty in determining minority and majority values for regression. Although there are numerous studies on synthetic oversampling for

classification, relatively little is done in the field of regression [3].

While the CVAE approach was developed to deal with classification and time series tasks, it should be noted. We propose a CVAE method combined with a relevance function. To the best of our knowledge, the first investigation of the CVAE-relevance function and regression issues in the SEE field in a situation dealing with small data availability.

3. Our approach

3.1 Conditional variational autoencoder

The conditional variational autoencoder (CVAE) [25] was expanded from the fundamental model of the VAE [21], which consists of an encoder and decoder, was expanded to characterize the distribution of observed data through latent variables in an unsupervised manner. As shown in Fig. 1, the CVAE approximates the conditional distribution $p_d(x|y)$ [45]. It can outperform the deterministic model when the distribution of $p_d(x|y)$ is multi-modal (the probability of xs varies for a given y). Assume that x is real, a deterministic regression model with MSE loss will predict the average blurry value for x. CVAE examines the x distribution, from which diverse and realistic objects can be sampled [46].

Similar to VAE, the lower bound of variational for CVAE can be derived by conditioning all distributions considered at y [46], as:

$$L_{CVAE}(x,y;\theta,\psi,\phi) = E_{q_{\phi(z|x,y)}} \log p_\theta(x|z,y) - D_{KL}(q_\phi(z|x,y)||p_\psi(z|y)) \leq \log p_{\theta,\psi}(x|y) \quad (1)$$

Optimizing CVAE objectives with the reparameterization technique. Note that the prior distribution $p_\psi(z|y)$ is dependent on y and is represented by a neural network with the parameter ψ. CVAE employs three neural networks that can be trained, whereas VAE only employs two [46]. In our study, we will suggest CVAE modifications such as gaussian stochastic neural networks and hybrid models with relevance functions derived from re-sampling regression.

3.2 Utility based regression

While the CVAE approach was developed to deal with classification tasks, it should be noted. In the meantime, to address regression issues where the target Y is continuous, it needs to be changed to label

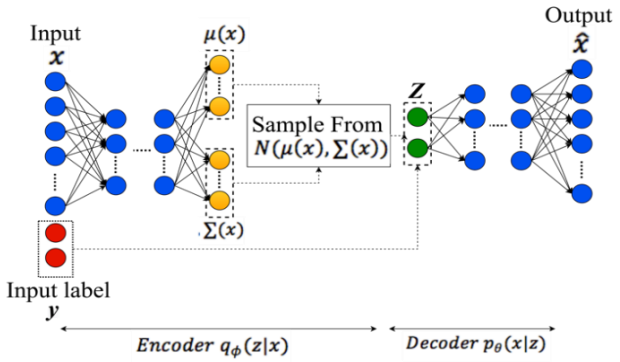


Figure. 1 CVAE Architecture

[0,1]. This study will use the approach found in the re-sampling regression model developed by Torgo et al. (2015). The concept of the value relevance of the target variable and the underlying assumption that this value relevance is not constant across domains are key to utility-based regression [8-9]. Relevance is an important property that reveals a domain-specific bias regarding the importance of different values. It is defined as a continuous function $\phi(Y): y \rightarrow [0,1]$ which maps the domain of the target variable y onto the relevance scale [0, 1], where 0 denote minimum and 1 denote maximum relevance.

The benefit of a prediction is defined by Torgo and Ribeiro (2007) as the proportion of the maximum benefit provided by the relevance of the true value of a given continuous objective variable [47-48], namely $\phi(y)$,

$$B_\phi(\hat{y},y) = \phi(y) * (1 - \Gamma_B(\hat{y},y)) \quad (2)$$

Where Γ_B is a bounded loss function for benefits.

While the predictive benefit depends on the usefulness of the actual value associated with $\phi(y)$, the cost depends on the relevance of the true value and the predicted value, as they are of different types. This indicates that the cost is proportional to the importance of the real and predicted values. This concept is captured by the shared relevance function, which calculates the weighted average of these two factors [9].

$$\phi^p(\hat{y},y) = (1-p) * \phi(\hat{y}) + p * \phi(y) \quad (3)$$

Where $p \in [0,1]$ is the factor that discriminates the type of error. Torgo and Ribeiro (2007) define the predictive cost as the proportion of the maximum cost determined by the joint relevance of the actual and predicted value [47]. Consequently, the cost function C_ϕ^p is defined as follows,

$$C_\phi^p(\hat{y},y) = \phi^p(\hat{y},y) * \Gamma_C(\hat{y},y) \quad (4)$$

Where ϕ^p represents the joint relevance function, and Γ_C represents the bounded cost loss function. The bounded loss is quite similar to Γ_B. The bounded loss Γ_C is a function with a range [0,1] that calculates the percentage of the maximum cost that should be assigned to a prediction based on the prediction error that is determined by the standard loss function. It needs to be equal to 0 for a perfect prediction, and it must increase to 1 as the prediction gets further and further away from being perfect. The utility of any prediction can be computed as the net balance of these two components, using the definitions of benefits and costs that were presented earlier in this section.

$$U_\phi^p(\hat{y}, y) = B_\phi(\hat{y}, y) - C_\phi^p(\hat{y}, y) \tag{5}$$

Where $B_\phi(\hat{y}, y)$ and $C_\phi^p(\hat{y}, y)$ are functions related to understanding the predicted costs and benefits.

4. Experiment design

In this section, we wanted to develop a specific augmentation method to address this problem of small dataset, which is very common in software engineering datasets. Our technique is the most appropriate and effective way to solve this problem by increasing the accuracy of the SEE method to overcome the estimation uncertainty caused by the small data available in the SEE context.

4.1 Problem statement

The challenge in constructing a SEE model typically involves data collection, which is time-consuming, workload, and expensive. Consequently, the available training data sample is a limited data set, which results in unsatisfactory SEE prediction model performance. To improve the performance of the prediction machine on the SEE model, it would be more efficient to generate a data synthesis project than to acquire as much data as possible on the completed software project (which would take a significant amount of time).

By gathering small data in the context of SEE. Based on data $\mathcal{D}$, we will construct a data synthesis project from the completed data.

$$\mathcal{D}^* = \mathcal{D} \cup \mathcal{D}' \tag{6}$$

Where $\mathcal{D}'$ represents the outcome of the data synthesis project, $\mathcal{D}^*$ is the new data used in the training process. Given an example of a randomly selected training $\mathcal{D} = \{(X_n, y_n)\}_{n=1}^N$, $X \in R^d$ where $X_n = (x_1, x_2, \dots, x_i) \in R^d$ synthetic project creation with all conditions distribution considered at $\phi(Y): y \to [0,1]$, for $y \in R^1$ as:

$$Q_{\phi(Encoder)}: (R^d, R) \to P_{\theta(decoder)}: (R^d, R) \tag{7}$$

$$\left(\begin{bmatrix} Q_\phi(Z|x_1) \\ \dots \\ Q_\phi(Z|x_d) \end{bmatrix}, [Q_\phi(Z|y)]\right) = P_\theta(\hat{X}, \hat{y}) \tag{8}$$

Where Q_ϕ is the encoder used for input data feed and P_θ is the decoder that produce synthesis data from Z.

4.2 Proposed synthetic data generator

The conditional variational autoencoder (CVAE) extends VAE by conditioning the encoder and decoder to class Y, as shown in Fig. 1. Encoder $Q(Z|X,Y)$ is now conditional on X and Y, while decoder $P(X|Z,Y)$ is now conditional on Z and Y. The objective of the variational lower bound on CVAE [25] is therefore described as follow:

$$L_{CVAE}(\theta, \phi; X, Y) = \mathbb{E}[logP(X|Z,Y)] - D_{KL}[Q(Z|X,Y)||P(Z|Y)] \tag{9}$$

The Y class designation is related to the encoder and decoder's conditional probability distributions. To initialize the DNN network parameters using the CVAE encoder network, the CVAE structure was modified to incorporate the Y class label only in the decoder network.

While the above approach was developed to handle the classification task, we apply the process using the relevance function method to change the continuous Y target to $\phi(Y): y \to [0,1]$.

Our proposed CVAE model is illustrated in Fig. 2. In the encoder portion, it is not connected to $Y[0,1]$, whereas the class label is an additional input in the decoder. Therefore, the decoder's probability distribution is dependent on the latent variable Z and the class label Y. The latent variable Z and label Y are connected and fed into the decoder in order to produce a new sample of the specified class. Thus, developing a new formula as:

$$L_{CVAE}(\theta, \phi; X, Y[0,1]) = \mathbb{E}[logP(X|Z,Y)] - D_{KL}[Q(Z|X)||P(Z|Y)] \tag{10}$$

L_{CVAE} consists of probability reconstruction $\mathbb{E}[logP(X|Z,Y)]$ and kullback–leibler (KL)

Algorithm 1: The pseudocode of training CVAE-relevance function

Input: the real dataset $\mathcal{D} = \{(X_n, y_n)\}_{n=1}^{N}$, $X \in R^d$, n is the dimension of data set

1. **Initialization:** supervised, decoder, encoder, prior distribution, latent space dimensions
2. **Data preprocessing:** normalization the training, validation and test dataset for all data is scaled to [0, 1]
3. Change $\phi(Y): y \rightarrow [0,1]$ for label using relevance function in re-sampling regression
4. **Training** CVAE**:**
5. **for** number of iterations **do**
6. **for** number of batches **do**
7. Input batch into the encoder;
8. Calculate loss on $Z \approx Encoder(X) = Q_\phi(Z|X)$, $\hat{X}, \hat{Y} \approx Decoder(Z) = P_\theta(X|Z,Y)$ according to Equation (5.5) and optimize through training data;
9. **Generated Synthetic data:**
10. Determine sample Z of the multivariate standard normal distribution $\mathcal{N}(0, I)$;
11. Determine the class label on the decoder as a conditional variable *Y;*
12. **for** number of samples **do**
13. Sample from latent space;
14. Use a trained decoder from the training step to get the synthetic data ($\hat{X}, \hat{Y} \approx \mathcal{D}'$);
15. **end for**
16. **return** Complete dataset $\mathcal{D}^* = \mathcal{D} \cup \mathcal{D}'$

Output: $\hat{X}, \hat{Y}$ synthetic data projects $\{(X_m, y_m)\}_{m=1}^{M}$

divergence $D_{KL}[Q(Z|X)||P(Z|Y)]$. The first term is to reconstruct X using the conditional probability distribution $P(X|Z,Y)$, and the second term is to characterize the $Q(Z|X)$ encoder distribution, which is close to the prior distribution $P(Z|Y)$ using the KL divergence metric. In this model, we use the class label as the conditional variable Y by sampling Z from the multivariate standard normal distribution $\mathcal{N}(0, I)$. The output $\hat{X}$ and $\hat{Y}$ will be transformed into inverse normalization to get synthesis results similar to real data.

We will show more details in Algorithm 1, as the development of the CVAE-relevance function to generate synthetic data in this study.

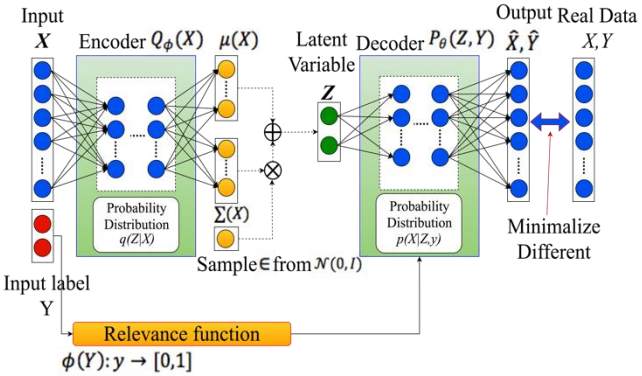


Figure. 2 Our Proposed CVAE-relevance function

4.3 Hyperparameter Setting

In this study, we conduct experiments to evaluate the efficacy of the proposed model. We use two different small datasets from the PROMISE Repository set (china and desharnais dataset) in the SEE field. This study will compare the proposed method to other conventional techniques for synthesizing data. The randomly accessible data sequence was partitioned into 70% train, 15% test, and 15% validation data set to define the parameters in our experiment. Training data is the only data set used for data augmentation. Each participant generates synthetic data based on the real data, and then combines the synthetic and real data to form the resultant synthetic data.

CVAE is a technique for generating synthetic data employing layers that are completely interconnected. The model was developed with a one-dimensional convolution layer and hyperparameter setting intended to capture the temporal relationships in the regression data [37]. In model optimization, the number of hidden layers and the neuron sizes of hidden layers are adjusted to reduce computational costs because they have a substantial effect on model performance. The trained model's hyperparameters use an embedding dimension of 256 for the hidden layer for all relevant data sets to improve the DNN's generalization performance [27, 37]. In this investigation, the default learning rate of Adam and the rectified linear unit (ReLU) optimizer in TensorFlow were utilized. Our model employs the Adam algorithm for a network with a learning rate of 1×10^{-3} (1e-3) [27, 49]. Other than the encoder and decoder output layers, which use the Linear activation function, all other layers use the ReLU activation function. We will select a size of four for the latent variable. ReLU optimal value may vary depending on the model [37, 50]. L_2 is utilized to avoid the problem of overfitting. If L_2 is 0, then the

Table 1. CVAE hyperparameter values

Hyperparameter	Values
the number hidden layer	256
batch size	50
latent size	4
epoch	100
regularization ($\mathcal{L}_2$)	1×10^{-4} (1e-4)
dropout probabilities	0.5
optimization	Adam
learning rate	1×10^{-3} (1e-3)
activation	ReLU
gaussian noise	0.2

original model is returned. However, if L_2 is too large, it will add too much weight and cause under-fitting, therefore, L_2 is 1×10−4 (1e-4).

We ran the model ten times to ensure that each data subset had an equal probability of being included in the test portion. The score was then determined by summing the precision of the model within the test subset. Lastly, the optimal parameter has the maximum cross-validation score. Table 1 depicts the hyperparameters utilized by this model.

4.4 Performance analysis

In this experiment, the efficacy of various regression models was evaluated and compared using four performance metrics. The metrics regression is imported from the *sklearn package*.

$$MAE = \sum_{i=1}^{n} \frac{|y_i - \hat{y}_i|}{n} \tag{11}$$

$$RMSE = \sqrt{\frac{\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}{n}} \tag{12}$$

$$RAE = \frac{\sum_{i=1}^{n}(|y_i - \hat{y}_i|)}{\sum_{i=1}^{n}(|y_i - \bar{y}_i|)} \tag{13}$$

$$R^2 = 1 - \frac{\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}{\sum_{i=1}^{n}(y_i - \bar{y}_i)^2} \tag{14}$$

5. Result and discussion

5.1 Convergence of model loss analysis

In this investigation, our CVAE served as the variational lower bound of the data's marginal likelihood. We employ a loss-measure method. The marginal likelihood also includes Kullback–Leibler (KL) divergence losses. Fig. 3 presents the losses of training the CVAE component for one repetition of running.

We can see that some of the datasets on training and testing loss drastically with increasing iteration

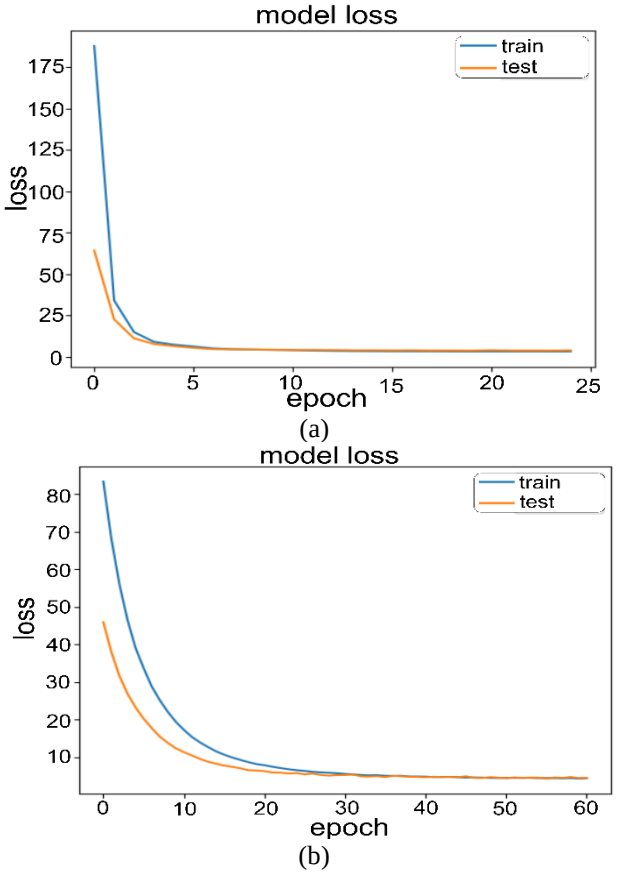


Figure. 3 Validation loss during training of CVAE models: (a) China dataset, and (b) Desharnais dataset

time and became stable after reaching values above 10 epochs. Loss function values tend to be stable, which indicates that CVAE has entered a convergence condition. Overall it shows that the CVAE training process is very stable. The results indicate that the loss function is close to constant; therefore, the model has converged. On the other hand, our model takes little time to train and find the generalizability of the data set.

5.2 Latent feature representation

We evaluated the distribution of synthetic data using two conventional approaches, namely t-distributed stochastic neighbor embedding (t-SNE) and principal component analysis (PCA) [51]. The t-SNE plots (left position) and PCA (right position) can be seen in Fig. 4, respectively, showing that the modeler can visually search for correlations using high-dimensional data set visualization techniques. PCA reduces data dimensions while maintaining variation [52]. t-SNE preserves the metric characteristics of the original height dimension data. It keeps data that indicates which points are neighbors [51].

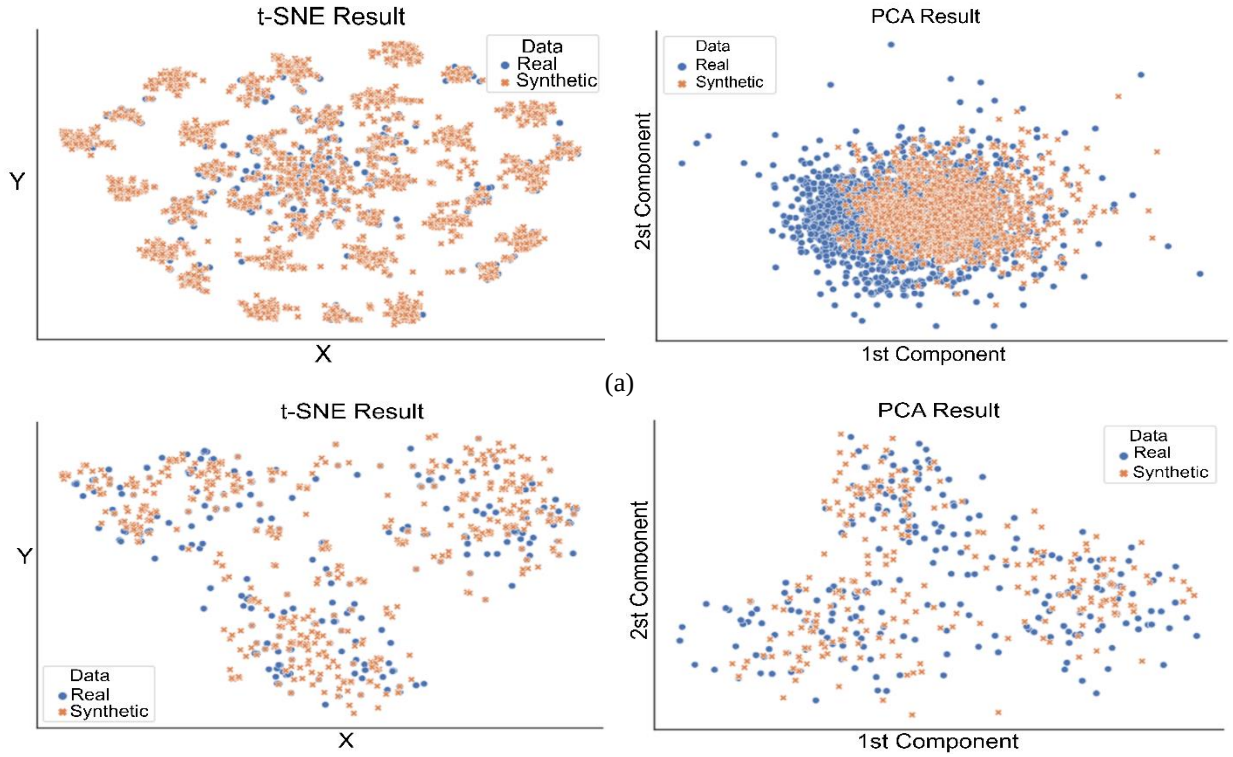


Figure. 4 t-SNE and PCA Visualizations of latent mean Vector *Z*: (a) The t-SNE and PCA for China data and (b) The t-SNE and PCA for Desharnais data

Fig. 4 shows a scatterplot of the learned features to visualize how well the distribution of synthetic data (orange color) resembles the real data set (blue color) in 2-Dimensional space. For each pattern, the plot indicates that the data points are closer together. Here, we will demonstrate how latent feature representation can be applied to a synthetic dataset. Therefore, we employ the concept of data point similarity. By visualizing the results, we obtain a better understanding of how well the model learns the distribution of the data. The results indicate that our method can help bridge the gap between real and synthetic data.

5.3 Cumulative distribution of sums per feature

In Fig. 5, we consider the cumulative number per feature both the real and synthetic data generated by our CVAE model. Most of the features in the synthetic data closely match the real data. In summary, our model with low privacy settings exhibits high-quality synthesis performance. In all cases, the synthetic data table is statistically similar to the real data table. We presume that the distribution per feature and difference plot generated by our model indicate a relatively high degree of similarity between the real data and the synthesized data.

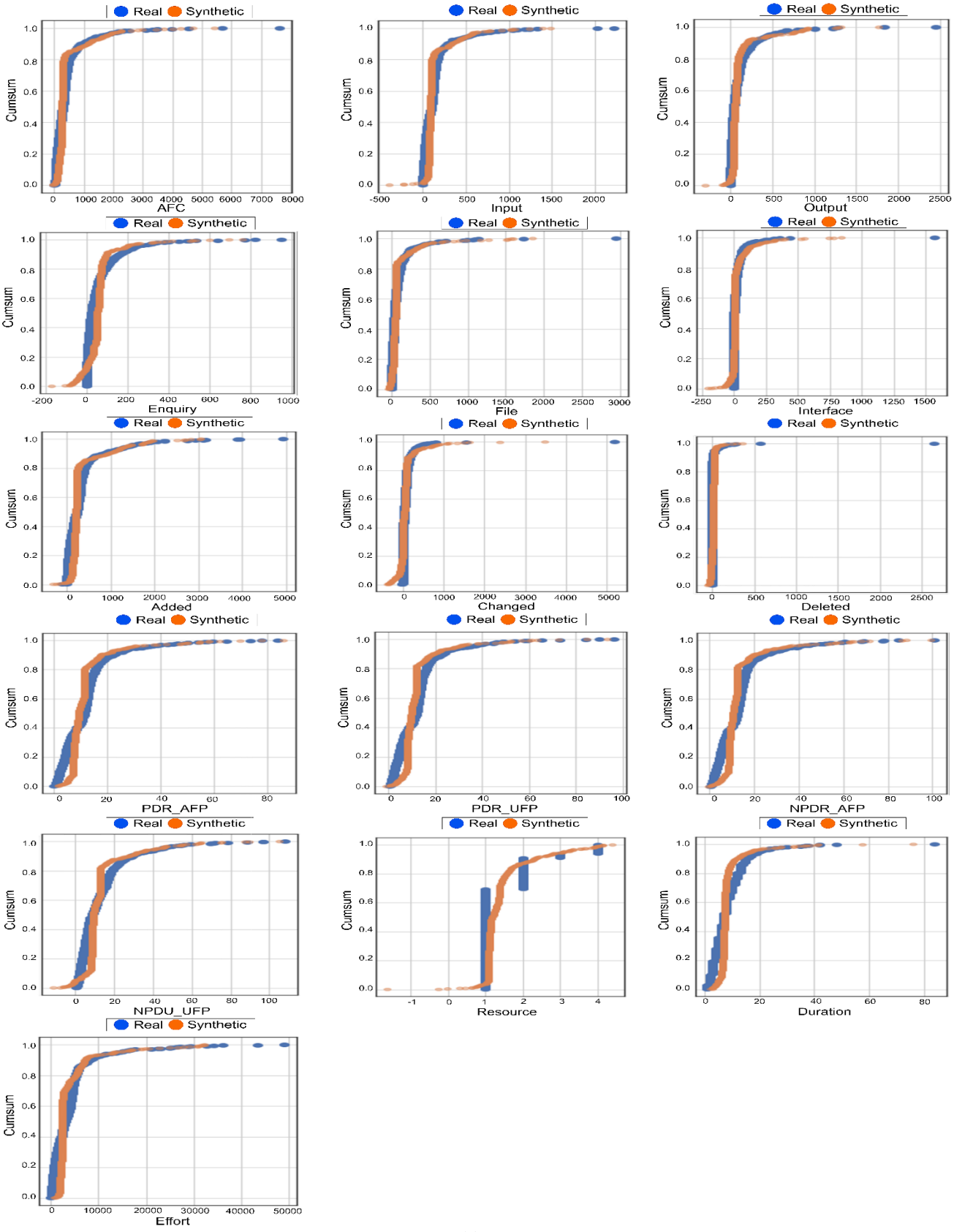
5.4 Statistical evaluation

On the other side, we also perform testing based on statistical and descriptive analysis. Descriptive analysis is summarized from all variables in real and synthetic data using the minimum (Min), maximum (Max), mean, and standard deviation (Std) values presented in tables 2 and 3.

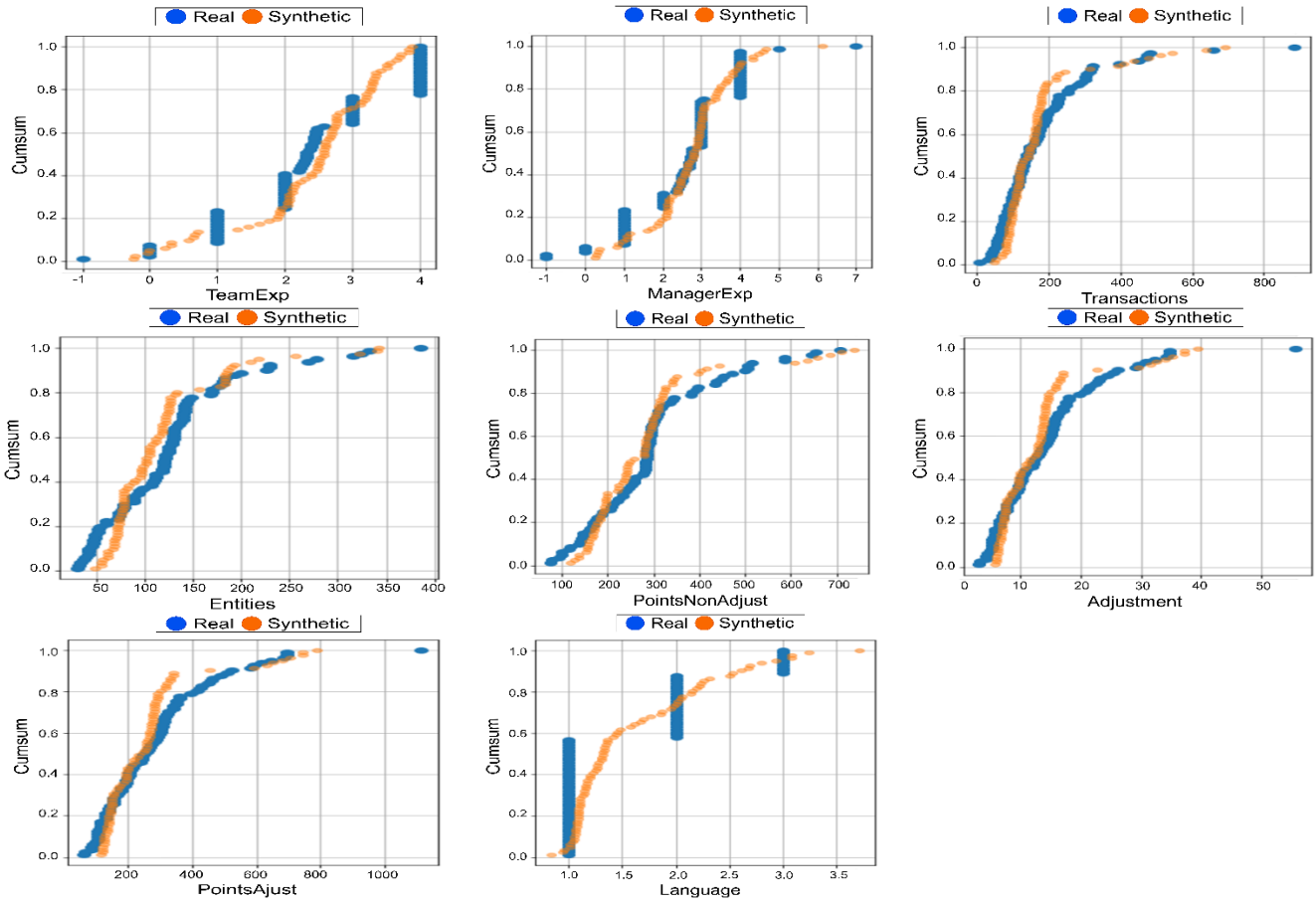
Three analyses were performed to assure that the two datasets were statistically identical. Thirty random samples were drawn from two datasets. Tables 2 and 3 display the p-values for every test. All p-values for the t-test variables are greater than 0.05. This indicates that the means of the two datasets are not significantly different. To evaluate the variance of the two datasets, a Levene test was performed. All variables have p-values more than 0.05. Similarly, the p-value of the Kolmogorov-Smirnov test is greater than 0.05, indicating that the distribution of variables between the two datasets is statistically significant. Therefore, the two datasets are statistically comparable. If the p-value is below 0.05, the variance of the dataset is not statistically significant.

5.5 Comparison with existing methods

We compare our CVAE proposed technique with six method state-of-the-art synthesizers [53, 55-56], including SMOTER [54] which was modified from


Real Synthetic
Cumsum
AFC
Input
Output
Enquiry
File
Interface
Added
Changed
Deleted
PDR_AFP
PDR_UFP
NPDR_AFP
NPDU_UFP
Resource
Duration
Effort

(a)



(b)

Figure. 5 Cumulative distribution in dataset per feature: (a) Similarities China dataset between real (blue) and synthetic (orange) and (b) Similarities Desharnais dataset between real (blue) and synthetic (orange)

Table 2. Descriptive statistical features of china dataset

Variable	Real Data				Synthetic Data				Statistical tests (p-value)		
	Min	Max	Mean	Std	Min	Max	Mean	Std	t-test	Levene	ks-test
AFP	9.00	17518.00	486.85	1059.17	3.39	5402.85	459.45	627.33	0.62	0.01	0.11
Input	0.00	9404.00	167.09	486.33	10.58	1484.38	151.69	180.91	0.51	0.00	0.24
Output	0.00	2455.00	113.60	221.27	0.07	1770.48	112.68	202.37	0.94	0.12	0.26
Enquiry	0.00	952.00	61.60	105.42	0.11	762.34	70.97	78.91	0.11	0.00	0.13
File	0.00	2955.00	91.23	210.27	0.12	1847.76	93.63	207.39	0.85	0.47	0.28
Interface	0.00	1572.00	24.23	85.04	0.02	842.83	36.69	88.43	0.02	0.17	0.29
Added	0.00	13580.00	360.35	829.84	7.04	3070.12	353.55	458.05	0.87	0.00	0.54
Changed	0.00	5193.00	85.06	290.85	0.05	3466.76	125.00	263.14	0.02	0.49	0.61
Deleted	0.00	2657.00	12.35	124.22	0.13	358.83	16.55	26.75	0.45	0.67	0.26
PDR_AFP	0.30	83.80	11.77	12.10	1.89	86.05	12.17	9.60	0.56	0.21	0.50
PDR_UFP	0.30	96.60	12.07	12.81	0.70	93.55	12.54	10.05	0.53	0.49	0.22
NPDR_AFP	0.40	101.00	13.26	14.00	0.27	99.20	13.55	11.24	0.72	0.86	0.78
NPDU_UFP	0.40	108.30	13.62	14.84	0.02	104.77	13.84	12.87	0.80	0.00	0.65
Resource	1.00	4.00	1.45	0.82	0.00	4.40	1.45	0.66	0.86	0.06	0.66
Duration	1.00	84.00	8.71	7.34	1.98	75.98	8.73	6.07	0.97	0.80	0.65
Effort	26.00	54620.00	3921.04	6480.85	742.52	32307.78	4239.07	4941.27	0.39	0.00	0.16

SEE, generative adversarial networks (GAN) [18, 55], Conditional Tabular Generative Adversarial Networks (CTGAN) [42, 56], Gaussian Copula (GC) [53, 56], Copula GAN (CGAN) [44], and Variantional Autoencoder (VAE) [34].

The lower MAE, RMSE, and RAE values conclusively show better results. On the other hand, the R^2 have higher values. Table 4 compares the performance metrics for the proposed CVAE model with different data augmentation methods for the two

Table 3. Descriptive statistical features of desharnais dataset

Variable	Real Data				Synthetic Data				Statistical tests (p-value)		
	Min	Max	Mean	Std	Min	Max	Mean	Std	t-test	Levene	ks-test
TeamExp	-1.00	4.00	2.18	1.41	0.00	3.89	2.39	0.99	0.50	0.00	0.00
ManagerExp	-1.00	7.00	2.53	1.64	0.27	6.10	1.07	1.07	0.55	0.00	0.01
Transactions	9.00	886.00	182.12	144.03	48.71	690.55	173.82	126.22	0.72	0.22	0.17
Entities	7.00	387.00	122.33	84.88	49.61	341.94	117.77	61.04	0.68	0.01	0.01
PointsNonAdjust	73.00	1127.00	304.45	180.21	117.80	737.37	282.12	132.68	0.68	0.03	0.24
Adjustment	5.00	52.00	27.62	10.59	2.23	47.68	25.82	8.67	0.64	0.06	0.02
PointsAdjust	62.00	1116.00	289.23	185.76	116.54	790.27	265.66	160.19	0.51	0.13	0.05
Language	1.00	3.00	1.55	0.70	0.83	3.71	1.60	0.64	0.62	0.45	0.85

resulting datasets. The results show that our CVAE approach has the best data augmentation performance quality with the accuracy value (MAE; RMSE; RAE; R^2) in each dataset of China (923.330; 1470.536; 0.436; 0.848) and Desharnais (9.540; 11.694; 0.041; 0.978).

Our CVAE method achieves the best working accuracy under the MAE, RMSE, RAE, and R^2 parameter assessment. In contrast, CTGAN has the worst performance (MAE and RMSE) in the China and Desharnais datasets. Meanwhile, CTGAN has the worst performance (RAE and R^2) in the China dataset. Copula GAN has the worst performance RAE on the Desharnais dataset. Lastly, Copula GAN has the worst performance R^2 on the Desharnais.

Fig. 4 demonstrates that the probability distribution function of the scatterplot generated by CVAE closely resembles that of the real dataset, indicating that CVAE makes a strong assumption that it can analyze the distribution characteristics of historical data to generate high-quality samples that closely resemble real data. Where progressive CVAE training is used to get a reasonably large training speed and make very good synthetic data from the input data. CVAE is utilized to examine encoder and decoder that can learn complex probability distributions from provided data and infer posterior distribution values based on latent variables.

Although according to Liang et al. (2018), how to include an automatic encoder of variational through augmentation structures while failing to consider additional information can be used more elegantly [57]. Thus, we concentrate on the selected probability function and study the regularization hyperparameter tuning to address this. Berthelot et al. (2018) have investigated the use of autoencoders within the framework of regularization in order to enhance linear interpolation [58]. This is because the Kullback-Leibler divergence (KL) is used as a conventional metric to identify the difference between the desired probability distributions [59]. The KL divergence enables the model to seamlessly normalize and interpolate latent space. However, if the KL divergence is not fine-tuned, it can result in a suboptimal network model [60].

We argue that modifying the CVAE method could encourage the model to learned interpretable data representations. The modified CVAE has an embedding component to conduct training in a supervised manner. A new objective function is applied to CVAE training to improve reconstruction capability.

Unfortunately, the GAN and Copula GAN encountered difficulties during training, which resulted in the loss of no explicit representation that had to be correctly synchronized, making optimal model parameters difficult and time-consuming. On the other hand, the fluctuating loss function of CTGAN and difficulty in converging causes poor performance [43], and this method has difficulty resolving the issue that continuous columns may have multiple modes while discrete columns are occasionally unbalanced, making modeling difficult. SMOTER, on the other hand, makes synthetic minority samples based on real samples and their neighbors. However, the over-sampling method tends to cause overfitting problems because duplicated samples are sometimes meaningless. In addition, this traditional method belongs to the shallow learning division and has trouble with data that is imbalanced and high-dimensional data [61]. However, this technique is susceptible to errors in generating noise samples, leading to overgeneralization or high variance.

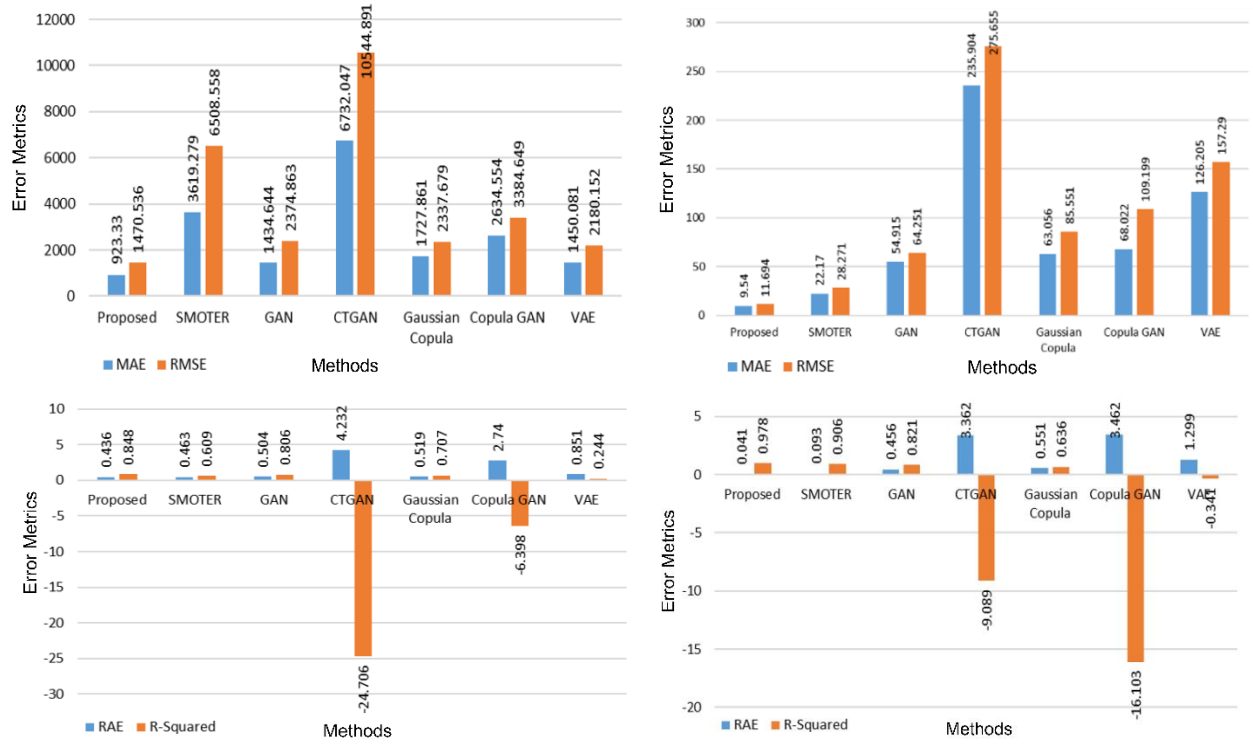
Meanwhile, Fig. 6 compares the data augmentation method with MAE, RMSE, RAE, and R^2. In this regard, we also observe that the measurement error rates for all models are presented in the visualization to make it easier to observe the performance of the data augmentation method.

5.6 Impact of synthetic data on regression

The subsection of this study was to explore the effect of generating synthetic data using the data

Table 4. The Comparison of data augmentation methods

Methods	MAE		RMSE		RAE		R^2	
	China	Desharnais	China	Desharnais	China	Desharnais	China	Desharnais
Proposed	**923.330**	**9.540**	**1470.536**	**11.694**	**0.436**	**0.041**	**0.848**	**0.978**
SMOTER	3619.279	22.170	6508.558	28.271	0.463	0.093	0.609	0.906
GAN	1434.644	54.915	2374.863	64.251	0.504	0.456	0.806	0.821
CTGAN	*6732.047*	*235.904*	*10544.891*	*275.655*	*4.232*	3.362	*-24.706*	-9.089
GC	1727.861	63.056	2337.679	85.551	0.519	0.551	0.707	0.636
CGAN	2634.554	68.022	3384.649	109.199	2.740	*3.462*	-6.398	*-16.103*
VAE	1450.081	126.205	2180.152	157.290	0.851	1.299	0.244	-0.341



(a) (b)

Figure. 6 Comparison of the evaluation metrics across various data augmentation methods: (a) China dataset and (b) Desharnais dataset

augmentation approach that was presented in this paper. We used synthetic data that was produced from two real-life data sets that are available to the public in order to assess how much of an impact they had on the performance of baseline machine learning algorithms in the field of regression. Our research produces synthetic data the same quantity of real data, which is then combined with real data and synthetics (synthetic, where $\mathcal{D} \approx \mathcal{D}'$ then $\mathcal{D}^* = \mathcal{D} \cup \mathcal{D}'$).

Several machine learning methods, such as support vector regression (SVR), and artificial neural networks (ANN), have been widely applied in the context of SEE, which is considered a necessary step [62]. On the other hand, other machine learning methods, like k-nearest neighbor (kNN), classification and regression tree (CART), and random forest (RF), are still ignored. In this study, five baseline machine learning algorithms, including CART, kNN, Multilayer Perceptron (MLP), SVR, and RF, will be used to evaluate the quality of the generated synthetic data.

Tables 5 and 6 list five baseline machine learning performance comparisons. The reported values are ten times the average of each SEE data set used to measure the difference in performance between the two datasets (real and synthetic dataset) using the MAE and RMSE performance values for each ML regressor model (without parameter setting, where parameters are chosen randomly within a range).

Table 5 shows that the number of estimated attempts is correctly regressed for the CART (781.972; 2400.498), kNN (657.558; 1466.325), and RF (464.299; 1159.219) models for our proposed CVAE. Meanwhile, the MLP (2000.044; 3162.108) and SVR (1565.578; 2536.080) models are better for VAE. The performance of CVAE data is better than the five methods of SMOTER, GAN, CTGAN, Gaussian copula, and Copula GAN. Overall, this

Table 5. Comparison of the machine learning in evaluating the quality of China synthetic data

ML	Evaluation Metric	Generated China Data Synthetic							
		Real	Proposed	SMOTER	GAN	CTGAN	GC	CGAN	VAE
CART	MAE	1032.260	**781.972**	1265.821	1351.340	*10898.370*	1866.030	3396.620	1296.500
	RMSE	2771.457	2400.498	3573.893	3440.989	*16763.359*	2634.495	5890.566	**1977.003**
kNN	MAE	1301.996	**657.558**	1650.629	1430.827	*8806.346*	2030.300	2788.059	1155.856
	RMSE	2584.747	**1466.325**	4219.714	3537.873	*13285.989*	2777.262	3936.318	1706.865
MLP	MAE	3212.537	2888.008	*9704.693*	2929.992	7227.012	5736.290	2916.550	**2000.044**
	RMSE	6578.298	5174.416	*14566.449*	6411.518	12466.135	7341.010	4544.227	**3162.108**
SVR	MAE	2592.019	2105.460	*8097.759*	2774.480	5310.070	3538.668	2465.435	**1565.578**
	RMSE	5997.487	4651.251	*11396.782*	6165.452	10448.966	4876.865	3799.945	**2536.080**
RF	MAE	626.885	**464.299**	777.796	869.848	*8240.996*	1683.556	2773.386	1119.743
	RMSE	2116.267	**1159.219**	1696.306	2110.294	*11820.325*	2279.200	3546.330	1543.388

Table 6. Comparison of the machine learning in evaluating the quality of desharnais synthetic data

ML	Evaluation Metric	Generated Desharnais Data Synthetic							
		Real	Proposed	SMOTER	GAN	CTGAN	GC	CGAN	VAE
CART	MAE	22.058	26.423	**20.642**	41.217	*254.411*	109.764	150.529	87.382
	RMSE	33.760	35.345	**32.971**	83.942	*321.637*	149.342	196.939	114.668
kNN	MAE	62.686	**37.866**	43.809	67.459	*235.568*	68.098	75.333	89.588
	RMSE	69.242	**60.025**	64.163	107.319	*271.431*	87.190	104.622	112.965
MLP	MAE	154.280	212.267	*395.142*	218.003	317.533	225.420	**67.912**	156.001
	RMSE	193.998	262.215	*472.998*	289.818	413.542	260.046	**123.525**	181.816
SVR	MAE	118.515	119.061	211.630	138.575	*214.598*	116.831	**61.391**	88.328
	RMSE	134.559	161.803	*281.686*	204.607	265.351	142.598	118.491	**100.689**
RF	MAE	18.710	**17.027**	18.849	34.677	*221.729*	73.170	73.205	89.349
	RMSE	31.257	**26.081**	25.729	65.470	*257.585*	96.209	107.865	104.557

shows that China synthetic data is better than real data.

Table 6 shows that the number of estimated attempts is correctly regressed for the kNN (37.866; 60.025) and RF (17.027; 26.081) models for our proposed CVAE. While the CART model (20.642; 32.971) is better for SMOTER, and Copula GAN is better for MLP (67.912; 123.525) and SVR (61.391; 118.491) models. The performance of CVAE data is better than the four methods of GAN, CTGAN, Copula GAN, and VAE. Overall, this shows that Desharnais' synthetic data is better than the real data.

In this case, the comparison of using synthetic results in improving the quality of performance on baseline machine learning baselines in the SEE context rather than using real data. This can be seen in the MAE and RMSE values which have the lowest values indicating the best performance. Our synthetic project generator (CVAE) consistently improves machine learning performance, such as CART, kNN, SVR, MLP, and RF on china dataset. Meanwhile, the desharnais dataset shows that it consistently improves the performance of machine learning, such as KNN and RF. In addition, we can see that the magnitude of the difference in performance varies depending on the competing data sets and methods.

VAE performance is better for machine learning models such as CART, MLP, and SVR than other data augmentation methods on china data set. On the other hand, SMOTER is also known as minority sampling technique and performs better than random sampling on desharnais datasets, consistently improves machine learning performance, such as CART. On the other hand, CGAN works well on desharnais datasets by enhancing the performance of machine learning, such as MLP and SVR. Overall it shows that using the generated synthetic data can improve machine learning performance better than using real data.

6. Conclusion

Our proposed CVAE-relevance function method is better than existing data augmentation methods. We employ a robust strategy by redesigning the DNN-based CVAE to perform categorical data conversion using vector-embedded ordinal coding and attribute labels to accelerate the convergence of the training process. Furthermore, to strengthen the correlation between attributes in the regression, we use a reconstructed probability distribution which aims to account for the variability of the distribution of variables in assisting the extraction of correlations in synthetic data. We train examples randomly in each epoch and perform hyperparameter tuning on our model. This helps increase generalizability. Our use of synthetic data results in improved quality of performance on popular machine learning baselines in SEE contexts than using real data.

The creation of the dataset's ordinal features (categories) is modeled by ordinal encoding. This modeling makes sense because of the effectiveness of

our data imputation and synthetic projects. However, our feature modeling may not correspond to reality. Other strategies that emphasize customer preferences or encode expert knowledge into a selection of training examples could improve predictive performance, leading to future research questions.

Conflicts of interest

The authors declare no conflict of interest.

Author contributions

Conceptualization, R. Marco; methodology, R. Marco, S. S. S. Ahmad and S. Ahmad; validation, R. Marco; formal analysis, R. Marco; investigation, R. Marco, S. S. S. Ahmad and S. Ahmad; resources, R. Marco, S. S. S. Ahmad and S. Ahmad; data curation, R. Marco; writing—original draft preparation, R. Marco, S. S. S. Ahmad and S. Ahmad; writing—review and editing, S. S. S. Ahmad and S. Ahmad; visualization, R. Marco; supervision, S. S. S. Ahmad and S. Ahmad; funding acquisition, R. Marco, S. S. S. Ahmad and S. Ahmad.

References

[1] S. Ezghari and A. Zahi, "Uncertainty management in Software effort estimation using a consistent fuzzy analogy-based method", *International Journal of Applied Soft Computing*, Vol. 67, pp. 540–557, 2018.

[2] L. Song, L. L. Minku, and Y. A. O. Xin, "Software effort interval prediction via Bayesian inference and synthetic bootstrap resampling", *International Journal of ACM Transactions on Software Engineering and Methodology*, Vol. 28, No. 1, pp. 1–43, 2019.

[3] L. Song, L. L. Minku, and X. Yao, "A novel automated approach for software effort estimation based on data augmentation", In: *Proc. of International Conf. on ACM Joint Meeting on European Software Engineering Conference and Symposium on the Foundations of Software Engineering*, Lake Buena Vista, USA, pp. 468–479, 2018.

[4] S. Amasaki, T. Yokogawa, and H. Aman, "Applying cross project defect prediction approaches to cross-company effort estimation", In: *Proc. of International Conf. on ACM International Conference Proceeding Series*, Recife, Brazil, pp. 76–79, 2019.

[5] H. Lu, E. Kocaguneli, and B. Cukic, "Defect prediction between software versions with active learning and dimensionality reduction", In: *Proc. of International Conf. on Software Reliability Engineering*, Naples, Italy, pp. 312–322, 2014.

[6] T. I. Tsai and D. C. Li, "Utilize bootstrap in small data set learning for pilot run modeling of manufacturing systems", *International Journal of Expert Systems with Applications*, Vol. 35, No. 3, pp. 1293–1300, 2008.

[7] J. Lemley, S. Bazrafkan, and P. Corcoran, "Smart Augmentation Learning an Optimal Data Augmentation Strategy", *International Journal of IEEE Access*, Vol. 5, pp. 5858–5869, 2017.

[8] L. Torgo, R. P. Ribeiro, B. Pfahringer, and P. Branco, "SMOTE for Regression", In: *Proc. of International Conf. on Artificial Intelligence Springer*, Berlin, Heidelberg, Vol. 8154, No. October 2015, pp. 378–389, 2013.

[9] L. Torgo, P. Branco, R. P. Ribeiro, and B. Pfahringer, "Resampling strategies for regression", *International Journal of Expert Systems*, Vol. 32, No. 3, pp. 465–476, 2015.

[10] Z. Li, Y. Zhao, and J. Fu, "SynC: A Copula based Framework for Generating Synthetic Data from Aggregated Sources", In: *Proc. of International Conf. on IEEE International Conference on Data Mining Workshops*, Sorrento, Italy, Vol. 2020-Novem , pp. 571–578 , 2020.

[11] A. Chen, S. Chen, and D. Zhao, "Creation of Fully-Synthetic, Multivariate Continuous Data Using Multivariate Adaptive Regression Splines with Multiple-Imputation", *International Journal of Researchgate*, No. March , 2020.

[12] T. E. Raghunathan, "Synthetic data", *International Journal of Annual Review of Statistics and Its Application*, Vol. 8, pp. 129–140, 2021.

[13] T. Raghunathan, J. Lepkowski, J. V. Hoewyk, and P. Solenberger, "A multivariate technique for multiply imputing missing values using a sequence of regression models", *International Journal of Survey methodology*, Vol. 27, No. 1, pp. 85–96, 2001.

[14] G. Caiola and J. P. Reiter, "Random forests for generating partially synthetic, categorical data", *International Journal of Transactions on Data Privacy*, Vol. 3, No. 1, pp. 27–42, 2010.

[15] J. Drechsler, "Using support vector machines for generating synthetic datasets", In: *Proc. of International Conf. on Privacy in Statistical Databases*, Springer, Berlin, Heidelberg, Vol. 6344, pp. 148–161, 2010.

[16] J. Reiter, "Using CART to generate partially synthetic public use microdata", *International Journal of Official Statistics*, Vol. 21, No. 3, pp. 441–462, 2005.

[17] A. Fernández, S. D. Río, N. V. Chawla, and F. Herrera, "An insight into imbalanced Big Data classification: outcomes and challenges", *International Journal of Complex & Intelligent Systems*, Vol. 3, No. 2, pp. 105–120, 2017.

[18] I. Goodfellow, J. P. Abadie, M. Mirza, B. Xu, D. W. Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative Adversarial Nets", *International Journal of Communications of the ACM*, Vol. 63, No. 11, pp. 139–144, 2020.

[19] J. Sun, J. Lang, H. Fujita, and H. Li, "Imbalanced enterprise credit evaluation with DTE-SBD: Decision tree ensemble based on SMOTE and bagging with differentiated sampling rates", *International Journal of Information Sciences*, Vol. 425, pp. 76–91, 2018.

[20] R. A. Hughes, I. R. White, S. R. Seaman, J. R. Carpenter, K. Tilling, and J. A. C. Sterne, "Joint modelling rationale for chained equations", *International Journal of BMC Medical Research Methodology*, Vol. 14, No. 1, 2014.

[21] D. P. Kingma and M. Welling, "Auto-encoding variational bayes", *International Journal of arXiv preprint*, No. 10, pp. 1–14, 2014.

[22] M. Simão, P. Neto, and O. Gibaru, "Improving novelty detection with generative adversarial networks on hand gesture data", *International Journal of Neurocomputing*, Vol. 358, pp. 437–445, 2019.

[23] P. Xu, R. Du, and Z. Zhang, "Predicting pipeline leakage in petrochemical system through GAN and LSTM", *International Journal of Knowledge-Based Systems*, Vol. 175, pp. 50–61, 2019.

[24] N. Tagasovska, D. Ackerer, and T. Vatter, "Copulas as high-dimensional generative models: Vine copula autoencoders", *International Journal of Advances in Neural Information Processing Systems*, Vol. 32, No. NeurIPS, pp. 1–13, 2019.

[25] D. P. Kingma, S. Mohamed, D. J. Rezende, and M. Welling, "Semi-supervised Learning with Deep Generative Models", *International Journal of Advances in Neural Information Processing Systems*, pp. 1–9, 2014.

[26] X. Gong, B. Tang, R. Zhu, W. Liao, and L. Song, "Data augmentation for electricity theft detection using conditional variational auto-encoder", *International Journal of Energies*, Vol. 13, No. 17, pp. 1–14, 2020.

[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", *International Journal of Sensors (Switzerland)*, Vol. 19, No. 11, 2019.

[28] M. Bregere and R. J. Bessa, "Simulating Tariff Impact in Electrical Energy Consumption Profiles with Conditional Variational Autoencoders", *International Journal of IEEE Access*, Vol. 8, pp. 131949–131966, 2020.

[29] S. Shao, P. Wang, and R. Yan, "Generative adversarial networks for data augmentation in machine fault diagnosis", *International Journal of Computers in Industry*, Vol. 106, pp. 85–93, 2019.

[30] Y. Lyu, J. Chen, Z. Song, and Q. Zhang, "Synthesizing data by transferring information in data-intensive regions to enhance process monitoring performance in data-scarce region", *International Journal of Canadian Journal of Chemical Engineering*, Vol. 99, No. S1, pp. S521–S539, 2021.

[31] J. Drechsler and J. P. Reiter, "Accounting for intruder uncertainty due to sampling when estimating identification disclosure risks in partially synthetic data", In: *Proc. of International Conf. on Privacy in Statistical Databases, Istanbul*, Turkey, Vol. 5262, pp. 227–238, 2008.

[32] K. He, X. Zhang, S. Ren, and J. Sun, "Identity Mappings in Deep Residual Networks", *International Journal of Springer International Publishing*, Vol. 9908, pp. 632–645, 2016.

[33] Z. Islam, M. A. Aty, Q. Cai, and J. Yuan, "Crash data augmentation using variational autoencoder", *International Journal of Accident Analysis and Prevention*, Vol. 151, No. December 2020, pp. 105950, 2021.

[34] Z. Wan, Y. Zhang, and H. He, "Variational autoencoder based synthetic data generation for imbalanced learning", In: *Proc. of International Conf. on IEEE Symposium Series on Computational Intelligence, Honolulu*, USA, Vol. 2018-Janua , pp. 1–7 , 2018.

[35] H. Lu, M. Du, K. Qian, X. He, Y. Sun, and K. Wang, "GAN-based Data Augmentation Strategy for Sensor Anomaly Detection in Industrial Robots", *International Journal of IEEE Sensors Journal*, No. c, pp. 1–11, 2021.

[36] D. Salinas, M. B. Schneider, L. Callot, R. Medico, and J. Gasthaus, "High-dimensional multivariate forecasting with low-rank Gaussian copula processes", In: *Proc. of International Conf. on Neural Information Processing Systems*, Vancouver, Canada, pp. 1–11, 2019.

[37] C. Fan, M. Chen, R. Tang, and J. Wang, "A novel deep generative modeling-based data augmentation strategy for improving short-term building energy predictions", *International*

Journal of Building Simulation, Vol. 15, No. 2, pp. 197–211, 2022.

[38] Y. Kamei, J. Keung, A. Monden, and K. Matsumoto, "An Over-sampling Method for Analogy-based Software Effort Estimation", In: *Proc. of International Conf. on ACM-IEEE international symposium on Empirical software engineering and measurement*, New York, United States, pp. 312–314, 2008.

[39] C. Bunkhumpornpat, K. Sinapiromsaran, and C. Lursinsap, "Safe-level-SMOTE: Safe-level-synthetic minority over-sampling technique for handling the class imbalanced problem", In: *Proc. of International Conf. on Advances in Knowledge Discovery and Data Mining, Springer, Berlin, Heidelberg*, Vol. 5476 LNAI, pp. 475–482, 2009.

[40] D. Li, W. Lin, C. Chen, H. Chen, and L. Lin, "Rebuilding sample distributions for small dataset learning", *International Journal of Decision Support Systems*, Vol. 105, No. January 2018, pp. 66–76, 2018.

[41] X. W. Liang, A. P. Jiang, T. Li, Y. Y. Xue, and G. T. Wang, "LR-SMOTE-An improved unbalanced data set oversampling based on K-means and SVM", *International Journal of Knowledge-Based Systems*, Vol. 196, p. 105845, 2020.

[42] L. Xu, M. Skoularidou, A. C. Infante, and K. Veeramachaneni, "Modeling tabular data using conditional GAN", *International Journal of Advances in Neural Information Processing Systems*, Vol. 32, No. NeurIPS, 2019.

[43] X. Gong, B. Tang, R. Zhu, W. Liao, and L. Song, "Data augmentation for electricity theft detection using conditional variational auto-encoder", *International Journal of Energies MDPI*, Vol. 13, No. 17, pp. 1–14, 2020.

[44] S. Kamthe, S. Assefa, and M. Deisenroth, "Copula Flows for Synthetic Data Generation", *International Journal of arXiv*, Vol. 5, 2021.

[45] K. Sohn, X. Yan, and H. Lee, "Learning structured output representation using deep conditional generative models", *International Journal of Advances in Neural Information Processing Systems*, Vol. 2015-Janua, pp. 3483–3491, 2015.

[46] O. Ivanov, M. Figurnov, and D. Vetrov, "Variational autoencoder with arbitrary conditioning", In: *Proc. of International Conf. on Learning Representations*, pp. 1–25, 2019.

[47] L. Torgo and R. Ribeiro, "Utility-Based Regression" , In: *Proc. of International Conf. on Principles and Practice of Knowledge Discovery in Databases*, Warsaw, Poland, pp. 597–604, 2007.

[48] P. Branco, R. P. Ribeiro, and L. Torgo, "UBL: an R package for Utility-based Learning", *International Journal of arXiv preprint*, 2016.

[49] D. P. Kingma and J. L. Ba, "Adam: A method for stochastic optimization", In: *Proc. of International Conf. on Learning Representations*, pp. 1–15, 2015.

[50] A. Krizhevsky and G. Hinton, "Convolutional deep belief networks on cifar-10", *International Journal of Citeseer*, pp. 1–9, 2010.

[51] L. V. D. Maaten and G. Hinton, "Visualizing data using t-SNE", *International Journal of Machine Learning Research*, Vol. 219, No. 1, pp. 2579–2605, 2008.

[52] H. Abdi and L. J. Williams, "Principal Component Analysis", *International Journal of Wiley Interdisplinary Reviews: Computational Statistics*, Vol. 2, No. 4, pp. 433–459, 2010.

[53] N. Patki, R. Wedge, and K. Veeramachaneni, "The synthetic data vault", In: *Proc. of International Conf. on Data Science and Advanced Analytics (DSAA)*, Montreal, Canada, pp. 399–410, 2016.

[54] P. Branco, L. Torgo, and R. P. Ribeiro, "Pre-processing approaches for imbalanced distributions in regression", *International Journal of Neurocomputing*, Vol. 343, No. xxxx, pp. 76–99, 2019.

[55] B. Chaudhari, H. Choudhary, A. Agarwal, K. Meena, and T. Bhowmik, "FairGen: Fair Synthetic Data Generation", In: *Proc. of International Conf. on Machine Learning*, Baltimore, Maryland, USA , pp. 1–5, 2022.

[56] M. Beigi and J. G. Mezey, "Synthetic Clinical Trial Data while Preserving Subject-Level Privacy", *International Journal of NeurIPS on Synthetic Data for Empowering ML Research*, pp. 1–8, 2022.

[57] D. Liang, R. G. Krishnan, M. D. Hoffman, and T. Jebara, "Variational Autoencoders for Collaborative Filtering", In: *Proc. of International Conf. on Neural Networks*, Vol. April 23-24, pp. 689–698, 2018.

[58] D. Berthelot, I. Goodfellow, C. Raffel, and A. Roy, "Understanding and improving interpolation in autoencoders via an adversarial regularizer", *International Journal of arXiv*, 2018.

[59] F. Naaz, A. Herle, J. Channegowda, A. Raj, and M. Lakshminarayanan, "A generative adversarial network-based synthetic data augmentation technique for battery condition evaluation", *International Journal of Energy Research*, No. June, pp. 1–16, 2021.

[60] P. Cristovao, H. Nakada, Y. Tanimura, and H. Asoh, "Generating In-Between Images through Learned Latent Space Representation Using Variational Autoencoders", *International Journal of IEEE Access*, Vol. 8, pp. 149456–149467, 2020.

[61] Y. Zhao, K. Hao, X. S. Tang, L. Chen, and B. Wei, "A conditional variational autoencoder based self-transferred algorithm for imbalanced classification", *International Journal of Knowledge-Based Systems*, Vol. 218, p. 106756, 2021.

[62] Y. Mahmood, N. Kama, A. Azmi, A. S. Khan, and M. Ali, "Software effort estimation accuracy prediction of machine learning techniques: A systematic performance evaluation", *International Journal of Software: Practice and Experience*, Vol. 52, No. 1, pp. 39–65, 2022.