

An AutoEncoder-based Numerical Training Data Augmentation Technique

Jueun Jeong

Dept. Electrical and Computer Engineering
University of Seoul
Seoul, South Korea
jeongjueun33@gmail.com

Hanseok Jeong

Dept. Electrical and Computer Engineering
University of Seoul
Seoul, South Korea
hsjeong625@gmail.com

Han-joon Kim

Dept. Electrical and Computer Engineering
University of Seoul
Seoul, South Korea
khj@uos.ac.kr

Abstract— This paper aims to automatically augment numerical tabular data by using the variational autoencoder model. For this, we try to solve the problem of class imbalance in numerical data and to improve the performance of the classification model by augmenting the training data. In this paper, we propose a new augmentation technique called ‘D-VAE’ which performs data augmentation through variational autoencoder with discretization for numerical columns; D-VAE artificially increases the number of records and the number of columns for a given tabular data. The main features of the proposed technique are to perform discretization and feature selection in the preprocessing process. For the discretization process, we use k -means algorithm, through which records within a given table are grouped, and then converted into one-hot vectors according to the clustering results. In addition, for memory efficiency, we reduced the number of parameters of the VAE model by using a relatively small number of features through feature selection called REFCV. To evaluate the performance of the proposed technique, we conducted various experiments by numerical data augmentation ratio using four open datasets.

Keywords—Autoencoder, Data Augmentation, Deep learning, Tabular data, VAE

I. INTRODUCTION

In the era of big data, data scientist try to utilize vast amounts of data as training data to develop machine learning-based classification (or prediction) models. However, we often experience low-quality data that does not allow to create an appropriate performance level of training model. Particularly, low-quality data has a problem with class (label) imbalance. When training with imbalanced data, minority classes are ignored and generated models are biased toward majority classes, which is a major factor in slowing down the performance of machine learning-based learning models. To solve this problem, data augmentation is a very helpful technique; data augmentation technique has the advantage of producing high-quality data by increasing the amount of training data and resolving the issue of class imbalance.

In addition, data augmentation automatically expands the data without human intervention, helping to capture patterns not obtained from the insufficient data initially given. Also, data augmentation allows us to take less time and effort to obtain sufficient training data. As a result, it makes a significant contribution to high value-added production. Recently, as the importance of data has increased with the rapid development of deep learning technology, various studies on data augmentation has been actively conducted. In our work, we propose a new data augmentation technique for numerical tabular data. This technique solves class imbalances to be used as training data for machine learning.

In general, there is a high risk of underfitting or overfitting when learning process is performed with insufficient data [1].

This paper is organized as follows. Section II describes a number of related work such as data discretization, clustering, feature selection, sampling techniques and generative model. Section III introduces ‘D-VAE’, which optimizes numerical features and generates augmented data with variational autoencoder. Through this, we describe how to generate high-quality training data while solving the class imbalance problem. Section IV shows the effectiveness of the proposed technique through experiments with numerical datasets. Section V summarizes this paper and presents future research work.

II. RELATED WORK

In order to preprocess numerical tabular data for data augmentation, we have used a number of data wrangling techniques such as data discretization, clustering, and feature selection. After preprocessing by the aforementioned method, sampling methods and generative models are used to augment the training data. The sampling technique adjusts the proportion of class imbalance data; typical methods include oversampling and undersampling. Generative model learns the distribution of training data to generate similar aggravated data; representative models include GAN, CTGAN, and autoencoder.

A. Data Discretization

Data Discretization [2] is a data preprocessing technique with which continuous features are converted into discrete (categorical) features; that is, it groups data intervals of continuous features or converts them into independent category names. Through this, simplifying the given data speeds up the learning process and improves the performance of the resulting model as the association between model features and classes (labels) increases [3]. Most of regression and classification models, including decision trees, perform better when numerical values are converted to discrete values.

The discretization method can be roughly divided with three criteria. The first criterion is to divide the discretization methods into supervised learning and unsupervised learning [4]. The supervised learning-based discretization method performs discretization by utilizing the correlation between independent features and classes (labels) with statistics such as entropy and chi-square. This cannot be used when there is no class within the training data because it is necessary to classify the training data with class information. In contrast, in the unsupervised learning-based discretization method, the user directly specifies the interval without class information [5][6]. The second criterion is to divide the discretization

methods into univariate or multivariate discretization methods. The univariate discretization method discretizes all features independently; for example, there are equal-width and equal-frequency discretization methods. However, there is a possibility that the interaction pattern between features may be lost in the discretization process. The multivariate discretization method compensates for the shortcomings of the univariate discretization method, and its discretization job is performed while considering the correlation between features. Third, it is divided into splitting and merging according to the discretization direction [7]. The split discretization method finds a split point at multiple points and subdivides it. In contrast, the merge discretization method finds the split point of multiple points and then merges the sections according to specific conditions [8].

B. Clustering

Clustering groups similar data into the same group and separates dissimilar data into different groups. Clustering types are classified into hierarchical clustering and non-hierarchical clustering. Hierarchical clustering goes through a process of merging intermediate clustering results with high similarity step by step, which does not require the number of clusters in advance. Non-hierarchical clustering performs an optimization process on data belonging to a cluster through iterative similarity calculations. Representative algorithms of non-hierarchical clustering include K-modes, K-means [9], and K-prototypes [10]. Clustering corresponds to a representative unsupervised learning, and due to its characteristics, it is usefully applied to the discretization process.

C. Feature Selection

Feature selection is to find the optimal combination of independent features that can significantly contribute to data augmentation [11], and through this, redundant or irrelevant features among independent features are removed. As a result, feature selection greatly contributes to increasing the model performance by augmenting the training data and improving the quality. Feature selection is largely divided into wrapper, filter, and embedded methods, and among these, we adopt the filter method.

D. Sampling Method

Oversampling is a technique for generating and reinforcing samples for a low ratio class according to the number of samples occupying a high ratio [12]. This is highly likely to cause overfitting because it iteratively learns the same data. In general, oversampling is more advantageous in predictive performance than undersampling, and the representative technique is SMOTE [13]. SMOTE algorithm utilizes a nearest-neighbor algorithm that calculates the difference between nearest neighbors to produce low-ratio class data. Undersampling [14] is a method of removing samples occupying a high ratio in consideration of the number of samples of a small ratio class. Since this removes part of the given data, there is a risk of lowering the performance of the resulting model due to loss of information.

E. Generative Model

Generative Adversarial Networks (GAN) [15], a generative model, is known as an effective technique for generating similar samples according to the distribution of given data. This is mainly used for unstructured data such as images, and related researches has been actively conducted because it can significantly contribute to the generalization of sampling techniques for data augmentation. In GAN, a generator that generates data and a discriminator that distinguishes original

data from augmented data compete with each other to complement the performance. Fig. 1 shows the GAN architecture, whose purpose is to generate fake data like the real one. In the operation process of the GAN, data obtained by randomly adding noise to the original data is received as an input value to the generator. The data generated by the generator corresponds to augmented data, and the original data and the augmented data are combined and input to the discriminator to be learned. The probability distribution is estimated by determining whether the data input to the discriminator is real data or fake data.

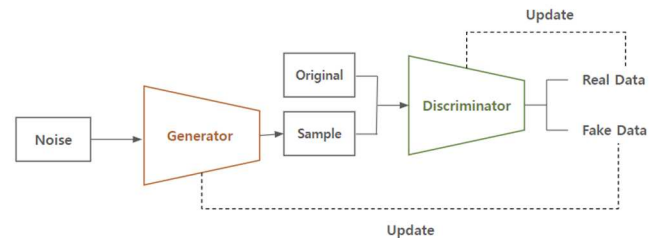


Fig. 1. GAN Architecture

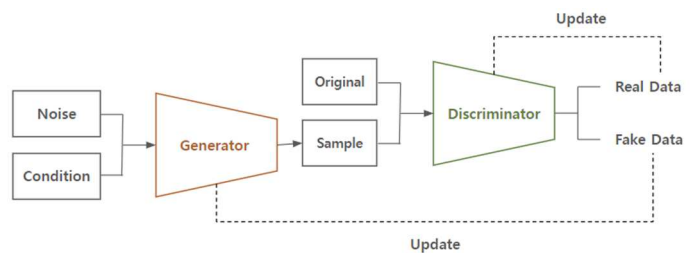


Fig. 2. CTGAN Architecture

Fig. 2 shows the CTGAN (Conditional Tabular Generative Adversarial Network) [16] architecture, which is a GAN-derived model and is applied to a structured table containing discrete and numeric features. Unlike GAN, CTGAN adds a special condition to the generator as an input value. The condition serves to learn the frequency of the original data so that minority classes are not forgotten. CTGAN has superior recall and balance performance compared to Vanilla GAN, WGAN, and TGAN proposed in the past for data with class imbalance problem [17]. However, it has the disadvantage that its performance responds sensitively according to the number of layers and the batch size. In addition, CTGAN still has the problem of being biased toward majority classes rather than minority classes when augmenting data.

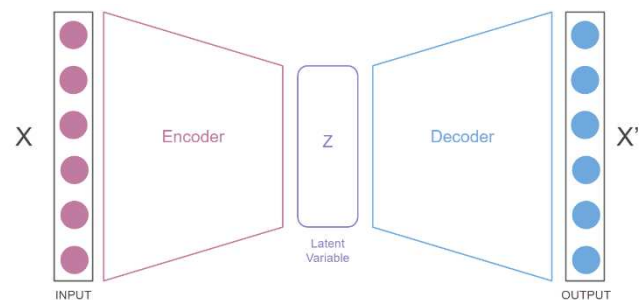


Fig. 3. autoencoder Model

Autoencoder [18] is an unsupervised learning technique that supports dimensionality reduction. The purpose is to learn to minimize the error between input X and output X' so that they have the same value as much as possible. In the autoencoder in Fig. 3, the encoder receives the input value X and converts it into a low-dimensional feature value, and the decoder

receives this feature value and converts it into an output value similar to the input value. A layer placed between the encoder and decoder layers includes nodes corresponding to latent features. These latent features are used to derive embedding vectors from given data.

III. PROPOSED METHOD

A. Data Augmentation Method Study

Tabular data is structured data and is widely used in our daily lives. As mentioned in Section II, GAN and autoencoder as examples of representative generative models are widely used in augmented techniques. The two previous research issues related to our proposed technique are as follows. First, the accuracy of the autoencoder was the highest through comparative experiments with bootstrap and GAN and autoencoder methods [19]. Second, an experiment was conducted with a voice medical dataset using the VAE model, and as a result of the experiment, VAE solved the data imbalance, resulting in significant improvement in F1-score performance.[20].

We used the VAE model to learn the value distribution of numerical tabular data and then generate the augmented data to improve performance.

B. VAE(variational AutoEncoder)

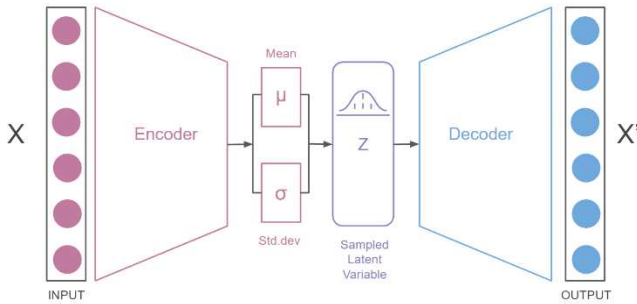


Fig. 4. variational AutoEncoder Model

Variational AutoEncoder (VAE) [21] is an AutoEncoder-based generative model. The purpose is to generate new augmented data similar to the distribution of training data. The latent space is a feature of the data that exhibits a uniform distribution. The encoder estimates the mean and variance of Z to represent the distribution of Z following a normal distribution. By inputting Z to the decoder, the input value is restored to generate augmented data.

Representative generative models for automatic augmentation are variational AutoEncoder and GANs. The differences between the two models is that variational autoencoder uses variational inference to learn data distribution. Feature inference is an approximation of the posterior probability distribution to an easy-to-handle probability distribution. And if the distribution is well learned, it is automatically sampled.

C. D-VAE

We propose a new augmentation technique called 'D-VAE', which is to obtain high-quality training data by resolving class imbalance and augmenting numerical data. Fig. 5 shows the detailed process of the proposed technique. The proposed technique is discretized and feature selection based on clustering results during data preprocessing.

Through the proposed technique, a given tabular data is grouped first and then discretized. We group using the k-means algorithm. For grouping, the proposed model obtains the minimum and maximum values and specifies a uniform width interval. The group is converted into a one-hot vector by one-hot encoding. Unlike label encoding, one-hot encoding has the advantage that it is not affected by the case relationship and order of numbers when training the model, so the one-hot encoding method is applied to the proposed technique.

However, as the number of features increases, the space required to store the vector continues to increase. To compensate for these shortcomings, we used the Recursive Feature Elimination with Cross Validation (REFCV) algorithm of feature selection [22]. REFCV removes features with low feature importance. After the two-step optimization process, augmented data with a distribution similar to the distribution of the training data is generated by the variational autoencoder. After pre-learning is completed, the deep neural network is fine-tuned using one-hot vectors. We used ReLU as an activation function and MSE (Mean Squared Error) as loss function.

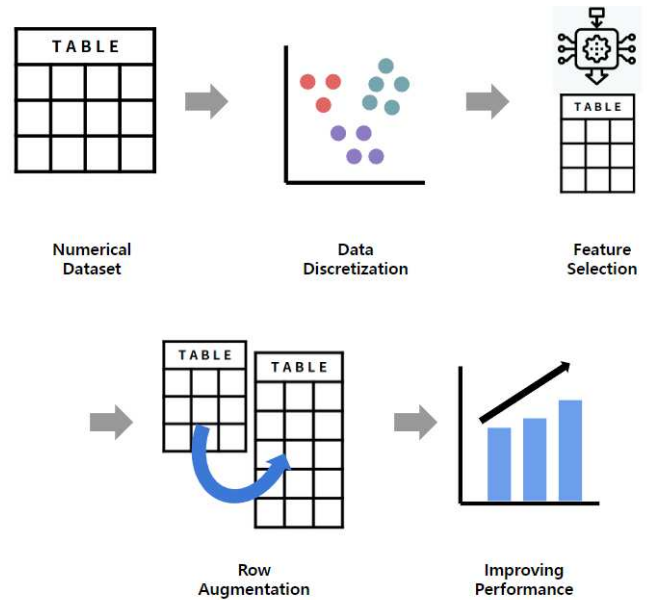


Fig. 5. D-VAE Process

D. D-VAE Process

The proposed technique sequentially performs the following four steps.

	A	B	C	...	G
T1					
T2					
...					
T200					

Fig. 6. D-VAE Process - Step 1

[Step 1] Fig. 6 shows an example of training data to illustrate the proposed technique. This is a numerical structured (tabular) dataset consisting of a total of 7 features from A to G and 200 records.

	A_01	A_02	A_03	B_01	B_02	B_03	C_01	C_02	C_03	...	G_01	G_02	G_03
T1													
T2													
...													
T200													

Fig. 7. D-VAE Process - Step 2

[Step 2] To augment the column, we firstly perform preprocessing with the following steps: First, it conducts a process of discretization as a result of clustering and grouping, which which one-hot vector is generated by one-hot encoding. Second, assuming that the user has three bins as shown in Fig. 7, the number of one-hot vector features increases by three per feature.

	B_01	B_02	B_03	C_01	C_02	C_03	...	G_01	G_02	G_03
T1										
T2										
...										
T200										

Fig. 8. D-VAE Process - Step 3

[Step 3] A feature selection process is performed using the discretized data. The REFCV technique removes features with low importance for label prediction. Assuming that feature 'A' has low importance in label prediction, feature 'A' is removed as shown in Fig. 8.

	B_01	B_02	B_03	C_01	C_02	C_03	...	G_01	G_02	G_03
T1										
T2										
...										
T799										
T800										

Fig. 9. D-VAE Process - Step 4

[Step 4] Fig. 9 shows the data after the preprocessing process, and it is augmented from 200 records to 800 records using variational autoencoder. As such, D-VAE obtains high-quality data by augmenting the records of the training data.

E. D-VAE Process Example

	ph	Hardness	Solids	Chloramines	Sulfate	Conductivity	Organic_carbon	Trihalomethanes	Turbidity	Potability
0	NaN	204.890455	20791.318981	7.300212	368.516441	564.308654	10.379783	86.990970	2.963135	0
1	3.716080	129.422921	18630.057858	6.635246	NaN	592.885359	15.180013	56.329076	4.500656	0
2	8.099124	224.236259	19909.541732	9.275884	NaN	418.606213	16.868637	66.420093	3.055934	0
3	8.316766	214.373394	22018.417441	8.059332	356.886136	363.266516	18.436524	100.341674	4.628771	0
4	9.092223	181.101509	17978.986339	6.546600	310.135738	398.410813	11.558279	31.997993	4.075075	0
...
3271	4.668102	193.681735	47580.991603	7.166639	359.948574	526.424171	13.894419	66.687695	4.435821	1
3272	7.808856	193.553212	17329.802160	8.061362	NaN	392.449580	19.903225	NaN	2.798243	1
3273	9.419510	175.762646	33155.578218	7.350233	NaN	432.044783	11.039070	69.845400	3.298875	1
3274	5.126763	230.603758	11983.869376	6.303357	NaN	402.883113	11.168946	77.488213	4.708658	1
3275	7.874671	195.102299	17404.177061	7.509306	NaN	327.459760	16.140368	78.698446	2.309149	1

3276 rows x 10 columns

Fig.10. Table before discretization(Features : 10)

The proposed technique utilizes the discretization method. Fig.10 shows the water quality data that was experimented to

explain the learning process. It consists of 3,276 records and 10 features. Its detailed information is given in Section IV.

	ph_0	ph_1	ph_2	ph_3	ph_4	Hardness_0	Hardness_1	Hardness_2	Hardness_3	Hardness_4	...	Trihalomethanes_0	Trihalomethanes_1	Trihalomethanes_2	Trihalomethanes_3	Trihalomethanes_4
0	1	0	0	0	0	0	0	1	0	0	...	0	0	0	0	1
1	0	1	0	0	0	1	0	0	0	0	...	0	0	1	0	0
2	0	0	0	1	0	0	0	0	1	0	...	0	0	1	0	0
3	0	0	0	1	0	0	0	0	1	0	...	0	0	0	0	1
4	0	0	0	0	1	0	0	1	0	0	...	0	1	0	0	0
...
3271	0	1	0	0	0	0	0	1	0	0	...	0	0	1	0	0
3272	0	0	0	1	0	0	0	1	0	0	...	1	0	0	0	0
3273	0	0	0	0	1	0	1	0	0	0	...	0	0	0	1	0
3274	0	1	0	0	0	0	0	0	1	0	...	0	0	0	1	0
3275	0	0	0	1	0	0	0	1	0	0	...	0	0	0	1	0

Fig. 11. Table after discretization(Features : 46)

As the first step of the proposed technique, Fig11 was divided into specified intervals in the range between the minimum and maximum values of the column. And in the experiment, since the interval was set to 5, the number of independent features increased from 9 to 45, resulting in a total of 46 features. Below is an example of specifying the range of the user-specified section. Table I shows the 'Hardness' value range of the water quality data. The minimum value of the feature is 40 and the maximum value is 320. To convert to one-hot encoding, it must go through discretization. In the discretization process, bins are divided into uniform widths based on the minimum and maximum values of the input values. And then it decides the data points based on the boundary values formed by segmentation.

Table I. Value interval for the feature 'hardness'

Division	value
Minimum	40
Maximum	320

If the user sets the interval to 5, the group is divided into five groups, as shown in Table II. Group 1 is 40-96, group 2 is 96-152, group 3 is 152-208, group 4 is 208-264, group 5 is 264-320.

TABLE II. RANGE BY GROUP OF 'HARDNESS' FEATURE(5 GROUPS)

Group	Range
1	40-96
2	96-152
3	152-208
4	208-264
5	264-320

Table III shows the IDs and values of the original data before one-hot encoding. A total of 3,276 records have values distributed between the minimum and maximum values. If the data value is a float, it is transformed into an integer and then into a one-hot vector under the experiment.

TABLE III. VALUE OF 'HARDNESS' FEATURE(BEFORE ONE-HOT ENCODING)

ID	Value
1	253

2	117
3	50
4	320
5	40
...	...
3276	66

Table IV shows the results of one-hot encoding when 5 bins are set in Table III. It can be seen that '1' is entered in the group and '0' is entered in the remaining groups in the range based on the value of the feature.

TABLE IV. VALUE OF 'HARDNESS' FEATURE(AFTER ONE-HOT ENCODING)

ID	Hardness_ 1	Hardness_ 2	Hardness_ 3	Hardness_ 4	Hardness_ 5
1	0	0	0	1	0
2	0	1	0	0	0
3	1	0	0	0	0
4	0	0	0	0	1
5	1	0	0	0	0
...					
3276	1	0	0	0	0

IV. EVALUATION

A. Experimental Setup

To verify the superiority of the proposed technique 'D-VAE' compared to the existing technique. The following four numerical datasets were selected as experimental data.

- **WDBC** : The data was released in 1995 by Wisconsin University Hospital in Madison, south of the U.S., and consists of a total of 569 records. The 30 numerical and class features, 'diagnosis', measured by digitally converting the images examined by removing the cells or tissues of the cyst using a thin needle, are marked as 1 if the result is negative and 0 if positive.
- **Diabetes** : Pima Indian diabetes data, consisting of a total of 768 records. Body characteristics such as age, number of pregnancies, glucose load test numbers, and blood pressure are expressed as numerical features. Of the total data, 65.1% are non-diabetes. The class feature 'Outcome' is marked as 1 for diabetes and 0 for non-diabetes.
- **Water Quality** : It is water quality data for determining whether it is safe drinking water, and consists of a total of 3,276 records. The degree of contamination of water such as pH value, chloramine, conductivity, and turbidity is expressed as numerical features. The class feature 'Potability' indicates whether a person is safe to drink. It is marked 1 if drinkable and 0 if not.

- **Credit Card** : In September 2013, European cardholders' credit card transactions consisted of a total of 284,807 records. The ratio of fraud among the total data is 0.172%, and the class distribution is very imbalanced. The data used in this experiment are composed of numerical features, which are the results of PCA conversion, and the class feature 'Class' is expressed as 1 for fraud and 0 for non-PCA.

TABLE V. DATASET USED IN THE EXPERIMENT

Dataset	Number of Features	Number of Records	Number of records per class	
			Class 1	Class 2
WDBC	32	569	357	212
Diabetes	9	768	500	268
Water Quality	10	3,276	1,998	1,278
Credit Card	31	284,807	235,821	48,986

As a condition given during the experiment, missing values in the given dataset were supplemented by applying the mean imputation method, and the unique number of the record was deleted because it does not help the learning model. And to implement the actual discretization process, we used the KBinsDiscretizer class that supports segmentation among Python packages.

A. Experiment Result

Before data augmentation, the proposed technique performs preprocessing. Thereafter, a feature that is helpful in label prediction is selected using the REFCV technique. After the optimization process, the data is augmented with a variational autoencoder. As shown in Table VI, the data augmentation ratio was divided into 5 ratios (20%, 40%, 60%, 80%, 100%) and performance analysis was performed.

TABLE VI. DATA AUGMENTATION RATIO

Dataset	Augmentation Ratio	Number of Records
WDBC	Original	569
	+20%	683
	+40%	797
	+60%	910
	+80%	1,024
	+100%	1,138
Dataset	Augmentation Ratio	Number of Records
Diabetes	Original	768
	+20%	922
	+40%	1,075
	+60%	1,229
	+80%	1,382
	+100%	1,536

Dataset	Augmentation Ratio	Number of Records
Water Quality	Original	3,276
	+20%	3,931
	+40%	4,586
	+60%	5,242
	+80%	5,897
	+100%	6,552
Dataset	Augmentation Ratio	Number of Records
Credit Card	Original	284,807
	+20%	341,768
	+40%	398,730
	+60%	455,691
	+80%	512,653
	+100%	569,614

The performance evaluation scales we used are accuracy and F1-score, and Logistic Regression, XGBoost, and Random Forest algorithms are used to construct the predictive model. We compare the performance of CTGAN and underlying VAE models together to demonstrate the performance of the proposed techniques. Table VII averaged the experimental performance of four numeric datasets. As a result of the experiment, the accuracy was improved by about 10% compared to the original data, and the F1-score was improved by about 20%. It was found that the performance of the autoencoder-based model was on average 30-40% better than that of the CTGAN.

TABLE VII. AVERAGE PERFORMANCE METRICS FOR FOUR DATASETS

DATASET		ORIGINAL	CTGAN	VAE	D-VAE
LOGISTIC Regression	ACCURACY	83%	64%	93%	96%
	F1-SCORE	75%	58%	95%	96%
Random Forest	ACCURACY	86%	63%	94%	95%
	F1-SCORE	74%	62%	96%	97%
XGBOOST	ACCURACY	89%	91%	99%	97%
	F1-SCORE	75%	90%	99%	98%

Table VIII compares the performance of the original data and 100% augmented data in four numerical datasets. Although the performance of the variational autoencoder and the proposed model is similar, the proposed model outperforms the conventional variational autoencoder on average by 1-2%

on small data sets with a small number of records. The tested credit card, diabetes and WDBC datasets perform best in the proposed model. However, the water quality dataset has the same performance as the variational autoencoder when increased by +80% from the original data. To my surprise, an increase of +100% results in Logistic Regression 2%, XGBoost 6% and Random Forest 9% lower.

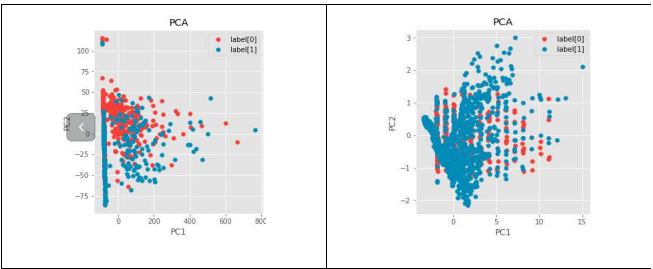
TABLE VIII. COMPARISON OF MODEL-SPECIFIC PERFORMANCE OF THE ORIGINAL DATA AND AUGMENTED DATA(+100%) OF THE DATASET

WDBC DATASET(NUMBER OF RECORDS : 569 -> 1,138)					
		ORIGINAL	CTGAN	VAE	D-VAE
LOGISTIC Regression	ACCURACY	96%	51%	90%	99%
	F1-SCORE	95%	44%	90%	99%
Random Forest	ACCURACY	96%	53%	90%	98%
	F1-SCORE	95%	57%	90%	99%
XGBOOST	ACCURACY	100%	100%	100%	100%
	F1-SCORE	100%	100%	100%	100%

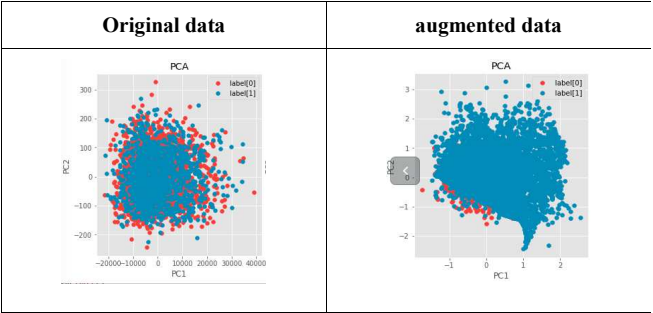
DIABETES DATASET(NUMBER OF RECORDS : 768 -> 1,536)					
		ORIGINAL	CTGAN	VAE	D-VAE
LOGISTIC Regression	ACCURACY	72%	53%	89%	92%
	F1-SCORE	72%	35%	93%	95%
Random Forest	ACCURACY	77%	47%	90%	92%
	F1-SCORE	67%	38%	94%	95%
XGBOOST	ACCURACY	92%	90%	99%	96%
	F1-SCORE	98%	89%	99%	98%

WATER QUALITY DATASET(NUBER OF RECORDS : 3,276 -> 6,552)					
		ORIGINAL	CTGAN	VAE	D-VAE
LOGISTIC Regression	ACCURACY	63%	58%	97%	95%
	F1-SCORE	63%	57%	99%	92%

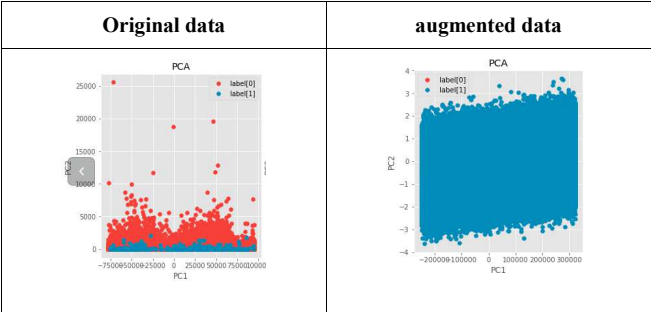
Random Forest	ACCURACY	70%	56%	98%	89%
	F1-SCORE	48%	54%	99%	93%
XGBOOST	ACCURACY	66%	75%	99%	93%
	F1-SCORE	41%	74%	99%	96%



Water Quality dataset



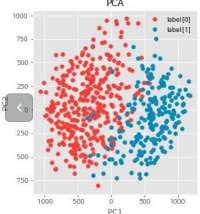
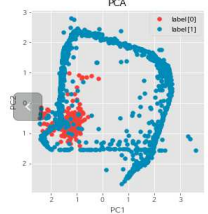
Credit Card dataset



CREDIT CARD DATASET(NUMBER OF RECORDS : 284,807->569,614)					
		ORIGINAL	CTGAN	VAE	D-VAE
LOGISTIC Regression	ACCURACY	99%	94%	96%	99%
	F1-SCORE	69%	94%	98%	99%
Random Forest	ACCURACY	99%	96%	99%	99%
	F1-SCORE	86%	97%	99%	99%
XGBOOST	ACCURACY	99%	97%	99%	99%
	F1-SCORE	69%	97%	99%	99%

One of the goals of the proposed technique is to resolve imbalanced data by amplifying the minority class. Table IX confirms the class distribution of original data and augmented data with PCA, a dimension reduction technique. We visualized the class distribution of the original data and the 100% augmented augmented data by reducing the dimensions of the data. In the original data, the classes were imbalanced because they were concentrated in one class. However, the class imbalance was complemented in augmented data compared to the original data.

TABLE IX. DATASET CLASS FEATURE DISTRIBUTION(PCA)

WDBC dataset	
Original data	augmented data
	
Diabetes dataset	
Original data	augmented data

The proposed technique selects features for memory efficiency and uses only features that are helpful in label prediction for learning. To compare the performance after applying the feature selection, the time taken before and after the feature selection is compared. If the number of bin is set to 5 during discretization, the number of feature increases to 5 per one. Under experimental conditions, if there were more than 30 duplicate values of a feature, it was converted to a one-hot vector. Table X shows the experimental results comparing the time taken before and after feature selection and the number of features. Features that are not helpful in label prediction were removed by using the REFCV technique during feature selection. As a result of the experiment, WDBC data decreased by 5.8326 seconds, diabetes data decreased by 0.5592 seconds, water quality data decreased by 6.9587 seconds, and credit card data decreased by 7.4557 seconds. By going through the feature optimization process through feature selection, the model training time was shortened.

TABLE X. COMPARISON OF TIMED REQUIRED BEFORE AND AFTER FEATURE SELECTION(UNIT : SECONDS)

Dataset	Before(Number of Features)	After(Number of Features)
WDBC	13.2903(151)	7.4577(64)
Diabetes	12.6607(37)	12.1015(32)

Water Quality	23.3419(46)	16.3832(44)
Credit Card	18.4951(151)	11.0394(102)

V. CONCLUSIONS

In this paper, we propose a new augmented technique called 'D-VAE' to augment the features and records of numerical data with a variational autoencoder. The problem of class imbalance in the given training data was solved through the proposed technique, and the efficacy of the proposed technique was confirmed through experiments using numerical data. The performance of the proposed technique was proven through experiments to be superior to CTGAN and the existing variational autoencoder. Considering the data augmentation ratio, the performance of the proposed technique was improved even when the number of records was augmented with a small amount compared to the variational autoencoder. In addition, we reduce model training time using feature selection to compensate for the shortcomings of memory waste in one-hot encoding. As a future study, we will augment the 'mixed' tabular data. In addition, research will be conducted on ways to solve information loss of discretized data and recover data.

ACKNOWLEDGMENT

This study was conducted with the support of the Information and Communication Planning Evaluation Institute (No. 2020-0-00121, Data Quality Assessment-based Data Advancement and Dataset Correction Technology Development Technology) and the results of the University ICT Research Center Support Project (IITP-2022-2018-01417) by the Ministry of Science and ICT Promotion.

REFERENCES

- [1] Douglas M. Hawkins, "The problem of overfitting", Journal of chemical information and computer sciences, Vol. 44, No. 1, pp. 1-12, Dec 2004.
- [2] Ian H. Witten, Eibe Frank, Mark A. Hall, "Data mining: Practical machine learning tools and techniques[3rd edition]", Morgan Kaufmann, Jan 2011.
- [3] James Dougherty, Bon Kohavi, Mehran Sahami, "Supervised and unsupervised discretization of continuous features", Proceedings of the 21th International Conference on Machine Learning 1995, pp. 194-202, Jun 1995.
- [4] Huan Liu, Farhad Hussain, Chew L. Tan, Manoranjan Dash, "Discretization: An enabling technique", Data Mining and knowledge Discovery Vol. 6, No. 4, pp.393-423, Jul 2002.
- [5] Sotiris Kotsiantis, Dimitris Kanellopoulos, "Discretization techniques: A recent survey" GESTS International Transactions on Computer Science and Engineering, Vol. 32, No. 1, pp. 47-58, 2006.
- [6] Kavita Das, Om P. Vyas, "A suitability study of discretization methods for associative classifiers", International Journal of Computer Applications, Vol.5, No. 10, pp. 46-51, Aug 2010.
- [7] Benjamin Johnston, Aaron Jones, "Applied Unsupervised Learning with Python", Packt Publishing, 2019.
- [8] Istvan Jonyer, Lawrence B. Holder, Diane J. Cook, "Graph-Based Hierarchical Conceptual Clustering in Structural Databases", Proceedings of AAAI, 2000.
- [9] Jackie A. Hartigan, Ma L. Wong, "Algorithm As 136:A K-Means Clustering Algorithm", Journal of the Royal Statistical Society. Series C, Vol. 28, No. 1, pp.100-108, Jan 1979.
- [10] Zhixue Huang, "Extensions to the k-means Algorithm for Clustering Large Data Sets with Categorical Values", Data Mining and Knowledge Discovery, pp.283-304, Sep 1998.
- [11] Manoranjan Dash, Hang Liu, "Feature Selection for classification", Intelligent data analysis, Vol. 1, No. 3, pp. 131-156, Jan 1997.
- [12] Alexander Liu, Joydeep Ghosh Member, "Generative Oversampling for Mining Imbalanced Datasets", Proceedings of International Conference on Data Mining 2007, pp. 25-28, 2007.
- [13] Hui Han, Wen-Yuan W, Bing-Huan M, "Borderline-SMOTE: a new over-sampling method in imbalanced data sets learning", ICIC, pp. 878-887, 2005.
- [14] Xu-Ying L, Jianxin Wu, Zhi-Hua Z, Senior Member, "Exploratory Undersampling for Class-Imbalance Learning", IEEE Transactions on Systems, Vol. 39, No.2, pp. 539-550, Dec 2009.
- [15] Ian J. Goodfellow, J Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, Yoshua Bengio, "Generative Adversarial Networks", arXiv, Jun 2014.
- [16] Lei Xu, Maria Skoularidou, Alfredo Cuesta-Infante, Kalyan Veeramachaneni, "Modeling Tabular using Conditional GAN" arXiv, Jul 2019.
- [17] Jiwon Choi, Jaewook Lee, Duksan Ryu, Suntae Kim, "Identification of Generative Adversarial Network Models Suitable for Software Defect Prediction", Journal of KIISE, Vol. 49, No. 1, pp. 52-59, Jan 2022.
- [18] Pascal Vincent, Hugo Larochelle, Yoshua Bengio, Pierre-Antoine M, "Extracting and composing robust features with denoising Autoencoders", Proceedings of the 25th international conference on Machine learning, Vol. 25, pp. 1096-1103, Jul 2008.
- [19] Mukrin Nakhwan, Rakkrit Duangsoithong, "Comparison Analysis of Data Augmentation using Bootstrap, GANs and Autoencoder", IEEE 14th International Conference, 2022.
- [20] Bahman Mirheidari, Yilin Pan, Daniel Blackburn, Ronan O'Malley, Traci Walker, Annalena Venneri, Markus Reuber, Heidi Christensen, "Data augmentation using generative networks to identify dementia, arXiv, Apr 2020.
- [21] Diederik P. Kingma, Max Welling, "Auto-Encoding Variational Bayes", arXiv, May 2014.
- [22] Puneet Misra, Arun S. Yadav, "Improving the classification accuracy using recursive feature elimination with cross-validation", International Journal on Emerging Technologies, Vol. 11, No. 3, pp.659-665, May 2020.