This article has been accepted for publication in IEEE Access. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2025.3582016

IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Date of publication xxxx 00, 0000, date of current version xxxx 00, 0000.

Digital Object Identifier 10.1109/ACCESS.2024.0429000

Dataset Construction Using Item Response Theory for Educational Machine Learning Competitions

TAKEAKI SAKABE[1], YUKO SAKURAI[1], EMIKO TSUTSUMI[2], SATOSHI OYAMA[3 4](Member, IEEE)

[1]Department of Computer Science, Nagoya Institute of Technology, Showa-ku, Nagoya, Aichi 466-8555, Japan
[2]Faculty of Science and Engineering, Hosei University, Koganei, Tokyo 184-8584, Japan
[3]Graduate School of Data Science, Nagoya City University, Mizuho-ku, Nagoya, Aichi 467-8501, Japan
[4]RIKEN Center for Advanced Intelligence Project, 1-4-1 Nihonbashi, Chuo-ku, Tokyo 103-0027, Japan

Corresponding author: Takeaki Sakabe (cmj14069@ict.nitech.ac.jp).

This work was partially supported by JSPS KAKENHI Grant Numbers JP21K19833, JP24K01112, and by JST CREST Grant Number JPMJCR21D1.

ABSTRACT Machine learning has been integrated into numerous applications and has emerged as one of the most transformative technologies in our daily lives. In recent years, the number of individuals studying machine learning has grown substantially, leading to the emergence of numerous educational competitions focused on building expertise in machine learning. In these competitions, the participants are tasked with constructing machine learning (ML) models. However, the dataset used to compare the performances of competing models is often selected arbitrarily, causing discrepancies between the dataset and participants' skill levels. This can result in competition outcomes that fail to accurately reflect the participants' abilities. We have developed a framework for generating image datasets that enable the abilities of competition participants to be accurately assessed. Specifically, we introduce the use of item response theory (IRT), commonly used in test creation and ability assessment, to estimate parameters such as item discrimination and difficulty for each image in existing datasets. Additionally, we utilize a conditional variational autoencoder (CVAE) that generates images with specific parameter values. These parameter values are generated based on the ability distribution of the competition participants and used to generate a dataset aligned with their ability distribution. To evaluate the effectiveness of the proposed framework, we conduct experiments using 810 ML models automatically created using 6 parameters with multiple values. Comparison of their performances between the original and the generated dataset showed that the latter was more effective in differentiating model performance. Unlike conventional IRT-based methods, which require human effort for dataset generation, our proposed framework fully automates the dataset generation process. By automating dataset generation, our approach streamlines the organization of ML competitions and ensures that datasets are well-suited to participants' skill levels. This automation reduces the challenges of hosting competitions, promoting their broader adoption in educational settings.

INDEX TERMS Item response theory, conditional VAE, data analysis competition, generating dataset.

I. INTRODUCTION

Machine learning (ML) is advancing across diverse fields, including natural language processing, image recognition, and speech recognition. Its growing integration into daily life is making the technology increasingly familiar to people. As a result, the number of individuals studying ML has increased, and educational ML competitions aimed at building expertise in ML are being actively held [1, 2, 3]. In most competitions involving both students and engineers studying ML, the dataset is selected arbitrarily. As a result, it may fail to effectively differentiate between model performances.

As mentioned above, the datasets widely used in competitions often fail to effectively differentiate between the performance of competing ML models. The MNIST [4] dataset is commonly used in competitions aimed at learners. It comprises images of handwritten numbers from 0 to 9, accompanied by labels representing the numbers in the images. In most cases, ML models achieve a correct response rate of 90% or higher on the MNIST dataset when using the default configurations provided by ML libraries like scikit-learn. MNIST is therefore not well-suited for assessing the performance of ML models used in competitive settings. Datasets that are more complex than MNIST, such as Fashion-MNIST [5] and Iris

This work is licensed under a Creative Commons Attribution 4.0 License. For more information, see https://creativecommons.org/licenses/by/4.0/

This article has been accepted for publication in IEEE Access. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2025.3582016

[6], are not necessarily suitable for competitions involving learners. For example, Fashion-MNIST achieves a correct response rate exceeding 90% in most cases regardless of the hyperparameter settings used with the scikit-learn MLPClassifier, a deep neural network algorithm. Similar to human examinations, the difficulty level of an ML competition should align with the skill level of the participants. In addition to difficulty, discrimination is also important. If the scores of the examinees are not sufficiently distributed, the assessment may fail to accurately reflect their abilities.

To address this problem, we have developed a framework for generating image datasets that facilitates an effective assessment of ML model performance in competitions. The framework combines item response theory (IRT) [7], a theory for creating examination questions and assessing examinees' abilities, with the conditional variational autoencoder (CVAE) [8], a deep generative model for generating images on the basis of specific parameters.

Although IRT is widely used to estimate item discrimination and difficulty parameters from human-prepared datasets, the process of constructing such datasets has traditionally relied on manual effort. This reliance on human intervention makes it challenging to construct datasets tailored to specific assessment needs. In contrast, our proposed framework combining IRT with CVAE automates the generation of datasets that match a given ability distribution and thereby markedly reduces the burden of dataset preparation while ensuring that each generated dataset is optimally structured for assessing participants' skills. Our framework also facilitates personalized learning environments by enabling the adjustment of datasets to match individual learning progress. Figure 1 provides an overview of the proposed dataset generation framework.

IRT is used to estimate the item discrimination and difficulty of each image in an existing dataset and to generate images with item discrimination and difficulty parameters that match the ability distribution of the participants in the competition. IRT is a statistical framework that uses mathematical models to design, analyze, and score assessments. In recent years, it has been used in various high-stakes tests. When evaluating the difficulty of examination questions or assessing the ability of examinees, "classical test theory" [9] is commonly applied. This theory bases its assessment on the percentage of correct answers or rough scores. In classical test theory, a key issue is sample dependence, meaning that the evaluation of the difficulty of a question depends on the level of the examinees in the test group. Moreover, it is difficult to consider the assessment ability of each question and to design a test that exhibits high overall assessment ability. IRT can estimate the item discrimination and difficulty of each question in a test through mathematical modeling. It can then design a test with an assessment ability matching the ability distribution of the examinees. We use IRT to estimate the item discrimination and difficulty of images in an existing image dataset and then generate a dataset using these estimates. The data used in this study are more biased than the test data typically used in IRT, so the estimated values may not converge. To address this, we used the Markov chain Monte Carlo (MCMC) [10] method to estimate the parameters.

When attempting to construct a new dataset by extracting images with suitable item discrimination and difficulty parameters from an existing dataset, the resulting dataset is inevitably smaller than the original. To overcome this limitation, we use a CVAE model, a deep generative model derived from the variational autoencoder (VAE) model. We trained the CVAE model on a large set of images paired with their corresponding parameter values. By providing specific parameter values to the trained model, we can generate new images with similar characteristics. This approach enables the generation of a new dataset that is comparable in size to the original, while retaining control over the item parameters.

To evaluate the effectiveness of the framework, we conducted experiments using real data from the MNIST, Fashion-MNIST, and FER-2013 datasets. The results demonstrated that the variance in the percentage of correct responses per model was larger and that the distribution was more peaked for the generated datasets. This indicates that the generated datasets are more suitable than the original datasets.

In summary, we aim to construct datasets that maximize assessment capability by accurately reflecting the skill differences among participants. This approach enables fair and effective assessment in educational ML competitions, particularly in cases where existing datasets, such as MNIST, lead to uniformly high model performance. To achieve this, we have developed a dataset generation framework that integrates IRT with CVAE to automatically generate image datasets with adjustable item difficulty and discrimination parameters.

Specifically, this study investigated two research questions: (1) Can we automatically construct datasets that more effectively differentiate ML model performance in educational competitions compared with existing datasets such as MNIST and Fashion-MNIST? (2) Can the integration of IRT and CVAE enable the automated generation of datasets that align with the skill distribution of participants, thereby improving the granularity of assessment? By answering these questions, we present new perspectives and methodologies for designing assessment systems for use in educational ML competitions.

The structure of this paper is as follows. Section II introduces research related to this study. Sections II-B and II-C explain the IRT and CVAE methodologies applied in this study. Section III describes the proposed framework, and Section IV describes the evaluation experiments and results. Finally, Section V concludes the paper with a summary of the key points and a mention of future work.

II. BACKGROUND AND RELATED WORKS

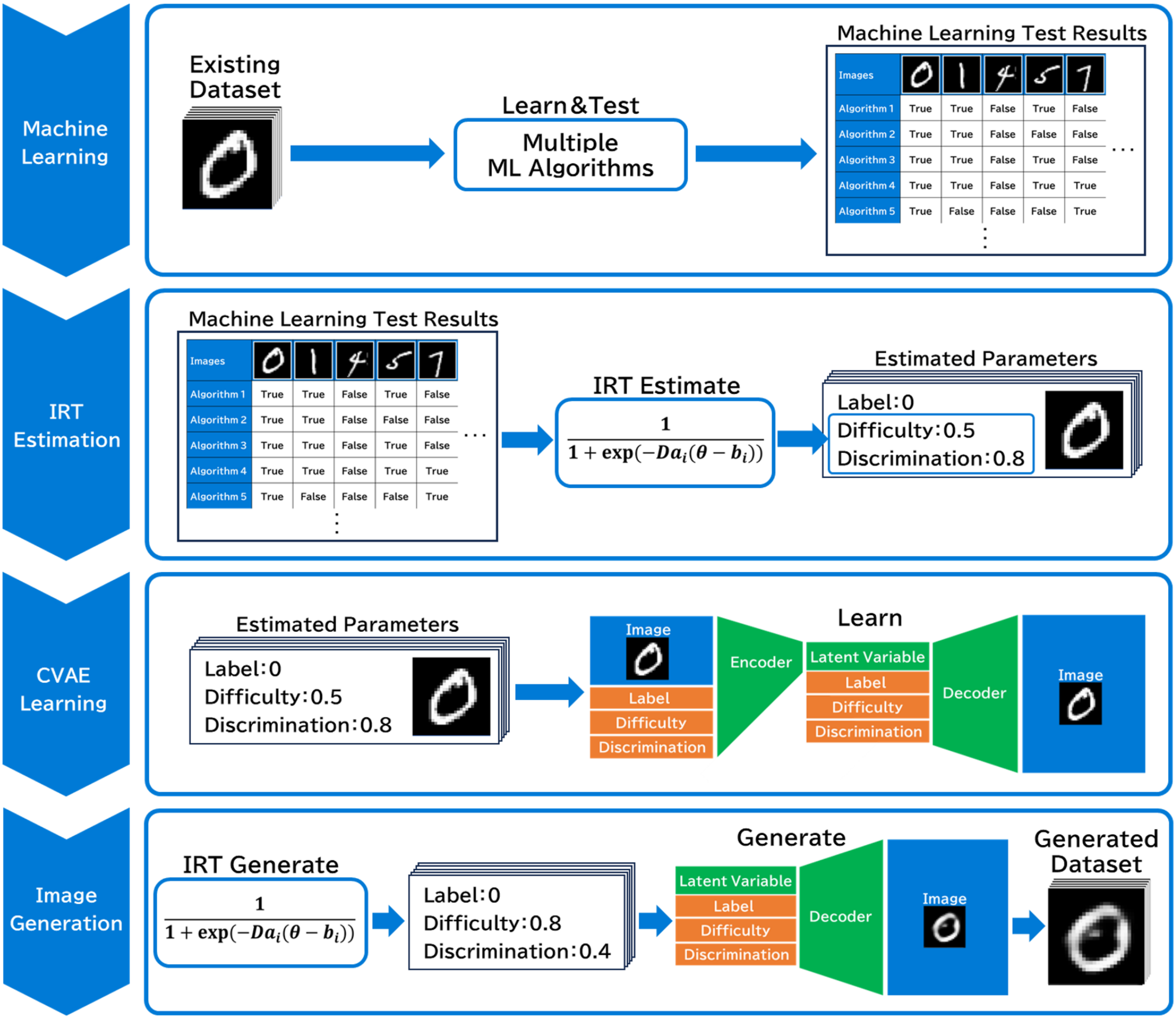
This section introduces research related to this study, with particular focus on IRT and the variational autoencoder.

A. HYPERPARAMETERS

In ML, the setting of the hyperparameters is as important as the choice of algorithm for creating an effective ML model

This work is licensed under a Creative Commons Attribution 4.0 License. For more information, see https://creativecommons.org/licenses/by/4.0/

This article has been accepted for publication in IEEE Access. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2025.3582016



Images					
Algorithm 1	True	True	False	True	False
Algorithm 2	True	True	False	False	False
Algorithm 3	True	True	False	True	False
Algorithm 4	True	True	False	True	True
Algorithm 5	True	False	False	False	True

FIGURE 1. Overview of the proposed dataset generation framework.

[11]. The hyperparameters govern the behavior of the model. They include the number of layers in the neural network and the dimensions of the latent variables in deep learning. They greatly affect the accuracy of the generated model. However, the parameter values to be used have typically been chosen based on intuition.

van Rijn and Hutter [12] devised a method for identifying the hyperparameters that significantly affect the accuracy of the model, analyzing them across various datasets and algorithm configurations. They also devised a method for analyzing and identifying hyperparameter values that are generally suitable across different datasets. Hutter et al. used functional ANOVA to identify the hyperparameters that significantly affect the accuracy of the model and used kernel density estimation to analyze the identified hyperparameters [13]. Yang and Shami [14] focused on the generality of auto-optimization methods across multiple datasets and compared several auto-optimization methods by using a large number of datasets. They found that a Bayesian optimization method called Bayesian optimization and Hyperband (BOHB) [15], which combines Bayesian optimization and the Hyperband algorithm, is the most versatile. Although there are several methods for automatically optimizing hyperparameters, their performance varies depending on the dataset of interest; moreover, there has not been a full discussion of which one has the best performance.

B. ITEM RESPONSE THEORY

IRT was established as a mean to overcome the problems with classical test theory. These problems include sample

This work is licensed under a Creative Commons Attribution 4.0 License. For more information, see https://creativecommons.org/licenses/by/4.0/

This article has been accepted for publication in IEEE Access. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2025.3582016

dependence, in which the evaluation of test difficulty and assessment of examinee ability depends on the level of the examinees in the group, and item dependence, in which the assessment of examinee ability depends on the level of the test questions. IRT can be used to evaluate the difficulty of the questions in a test and the ability of the examinees, independent of the group's proficiency level or the specifics of the test questions, by using the item characteristic curve (ICC). It can also be used to estimate examinees' abilities while minimizing the effect of questions with low accuracy in assessing examinee ability.

In an ICC, the horizontal axis represents the examinee's ability θ, and the vertical axis corresponds to the probability of the correct response to the given question. The item parameters representing the characteristics of the question and θ are used to express the probability of the correct response to the question. In this way, IRT can estimate an examinee's ability by considering the question's characteristics. Logistic models are commonly used to represent the ICC. Depending on the number of unknown parameters (item parameters) included in the model, variations such as one-parameter logistic models and two-parameter logistic models are commonly used. The following subsections provide an overview of these models.

1) One-parameter logistic models

The ICC of a one-parameter logistic model (1PLM), a model including one item parameter, is expressed by equation (1). In this paper, the ICC is expressed as $P_i(\theta)$, i.e., as a function of θ.

$$P_i(\theta) = \frac{1}{1+\exp(-Da(\theta - b_i))}, \qquad -\infty < \theta < \infty \quad (1)$$

where a is the item discrimination parameter common to all questions, b_i is the item difficulty of question i, and D is the scale factor (which is typically set to 1.7). This equation represents the probability that an examinee with ability θ answers question i correctly, and only b_i represents the characteristics of the individual problem.

For example, consider the ICCs of three questions: question 1 ($b = 1.0$), question 2 ($b = 0.0$), and question 3 ($b = -1.0$), with $D = 1.7$ and $a = 1$ (see Figure 2). These ICCs represent the probability that an examinee with ability θ will answer questions 1 through 3 correctly; the probability increases with ability θ. Item difficulty b in the 1PLM represents ability θ, for which the probability of a correct response is 0.5, as shown in the figure.

Thus, the 1PLM expresses the relationship between question difficulty, examinee ability, and correct answer probability.

2) Two-parameter logistic model

The ICC of a two-parameter logistic model (2PLM), which has two item parameters, is expressed as

$$P_i(\theta) = \frac{1}{1+\exp(-Da_i(\theta - b_i))}, \qquad -\infty < \theta < \infty \quad (2)$$

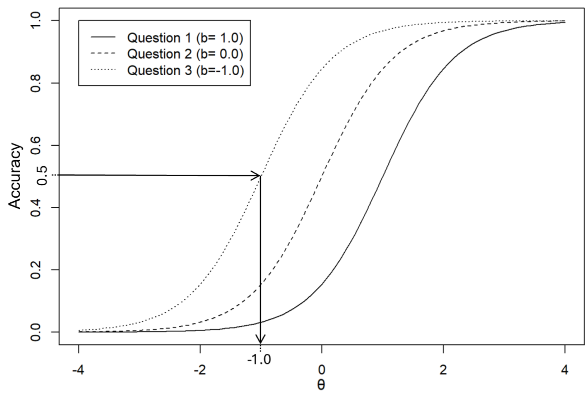


FIGURE 2. Example ICCs based on 1PLM; all items share the same discrimination parameter.

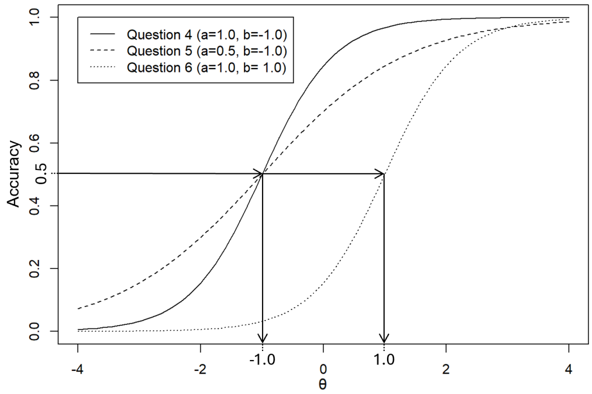


FIGURE 3. Example ICCs based on 2PLM; items vary in both discrimination and difficulty.

where a_i and b_i are the item discrimination and item difficulty parameters of question i, and D is the scale factor (which is typically set to 1.7).

Equation (2) represents the probability that an examinee with ability θ answers question i correctly, and a_i and b_i represent the characteristics of individual questions.

For example, consider the ICCs of three questions, question 4 ($a = 1.0, b = -1.0$), question 5 ($a = 0.5, b = -1.0$), and question 6 ($a = 1.0, b = 1.0$) with $D = 1.7$ (see Figure 3). As in the 1PLM, item difficulty b is equal to the examinee's ability, which gives a 0.5 probability of answering a question correctly. In this example, the item difficulties of questions 4 and 5 are both -1.0, and ability θ must be -1.0 for the 0.5 probability of correct answers to these questions. The item discrimination of the 2PLM represents the magnitude of the gradient of the curve; question 4 with $a = 1.0$ has a larger gradient around $\theta = -1.0$ than question 5 with $a = 0.5$. This indicates that question 4 with $a = 1.0$ has a higher ability to assess the examinee's ability around $\theta = -1.0$.

Thus, the 2PLM expresses the relationship between item discrimination, item difficulty, examinee ability, and correct answer probability.

This work is licensed under a Creative Commons Attribution 4.0 License. For more information, see https://creativecommons.org/licenses/by/4.0/

This article has been accepted for publication in IEEE Access. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2025.3582016

3) Parameter estimation

One of the primary challenges in applying IRT to ML competitions is the presence of biased datasets, which can lead to inaccurate estimates of item difficulty and discrimination. Traditional IRT methods are based on the assumption of a balanced and representative sample of participants, whereas real-world datasets often contain skewed distributions of response accuracies. To mitigate this, we use the MCMC method for parameter estimation, which improves convergence stability even when dealing with biased datasets. MCMC enables iterative refinement of the item discrimination and difficulty parameters, ensuring that the estimated values more accurately reflect the true characteristics of the dataset, which improves the reliability of competence assessment.

In this study, we used a 2PLM, which uses item discrimination and difficulty, to generate a dataset that is highly suitable for assessing the abilities of the participants in the competition. We will thus focus on 2PLM hereafter.

Item discrimination a, item difficulty b, and ability θ (described in Section II-B2) are not initially given; their true values need to be estimated on the basis of existing test results. However, the data used in this study was more biased than the test data typically used in IRT, so the estimated values may not converge. Therefore, as mentioned above, we used an algorithm based on the Metropolis method, a variant of the MCMC approach [16], which uses random numbers for numerical estimation.

In this section, we outline the basic estimation procedure for the Metropolis method. The target of estimation is denoted as $\boldsymbol{x} = \{x_i \mid i = \{1, \dots, I\}\}$; $\boldsymbol{x}$ is iteratively updated to maximize probability $P(\boldsymbol{x})$ represented by $\boldsymbol{x}$. The parameter updates use a value analogous to the log-likelihood of $P(\boldsymbol{x})$, denoted as $S(\boldsymbol{x})$.

$$S(\boldsymbol{x}) = -\log_e P(\boldsymbol{x}) - \log_e Z \tag{3}$$

where Z is the distribution function, which depends on the estimation target.

The estimation process consists of five steps.

Step1) Randomly sample all initial values of $\boldsymbol{x}$, such as ones drawn from a normal distribution.

Step2) Designate the estimation target as x_j, and generate a new set $\boldsymbol{x}^{'}$ by modifying only x_j in $\boldsymbol{x}$ to values sampled from a fixed interval centered around x_j.

Step3) Generate a uniform random number r between 0 and 1. If $r < e^{S(\boldsymbol{x})-S(\boldsymbol{x}^{'})}$, accept $\boldsymbol{x}^{'}$ and substitute it for $\boldsymbol{x}$. If not, reject and discard $\boldsymbol{x}^{'}$.

Step4) Repeat Steps 2 and 3 for $\{x_i \mid i \in \{i, \dots, I\}\}$.

Step5) Repeat Steps 2 to 4 as a single update until the values converge.

As described above, the Metropolis method iteratively updates $\boldsymbol{x}$ to maximize the log-likelihood. As the effect of the updates on the log-likelihood diminishes, the method becomes increasingly likely to maintain similar values, favoring stability in the parameter estimates.

4) General trends in the application and development of IRT

IRT typically handles answer data in a binary format, i.e., correct or incorrect, and the response data considered in this paper is also binary. Samejima [17] devised a graded response model that can handle more than three ordinal scales. This model considers "the examinee's response to question j is categorized into $k(k = 0, 1, \dots, K_j)$ or higher." This allows the model to be applied to tests in which partial scores need to be considered, such as a mathematics exam. Uto et al. [18] devised a model for automatically creating reading comprehension questions with adjustable difficulty that combines IRT with a deep learning model. It automatically creates questions related to target sentences. Conventional methods for doing this cannot create questions with a difficulty level that matches the learner's ability. Uto et al.'s model overcomes this problem by fine-tuning reading comprehension questions and difficulty estimates from IRT to a large-scale language model, automatically creating questions at the target difficulty level.

IRT has been applied and utilized in various ways, not only for ability assessment but also for modeling cognitive processes and test-taking behavior. For instance, De Boeck and Jeon [19] extended IRT to jointly model response accuracy and response times, enabling a deeper understanding of cognitive processes and improving the precision of ability assessment. Moreover, IRT has found increasing application in intelligent learning systems. For example, Wang et al. [20] integrated IRT with other psychometric models and deep learning techniques to construct an interpretable learning diagnosis framework that enables both accurate performance prediction and cognitive insight into intelligent tutoring systems. Furthermore, IRT has been integrated into the development of automatic scoring systems. For example, Uto et al. [21] devised an IRT-based method for integrating the prediction scores from multiple automated essay scoring models, treating each model as a virtual rater. By incorporating rater characteristics such as severity and consistency through a generalized many-facet Rasch model, their approach achieved higher scoring accuracy and improved interpretability of each model's scoring behavior.

C. CONDITIONAL VAE

Deep generative models integrate deep learning with a generative model and define probability distributions of generative models through the use of a deep neural network. Various deep generative models have been introduced. In this study, we use a CVAE model.

1) Variational autoencoder

An autoencoder is an ML model comprising two neural networks: an encoder that compresses input images into latent variables and a decoder that reconstructs the images from the latent variables. The latent variables generated by traditional autoencoders have complex distributions, making them difficult to handle. The variational autoencoder (VAE) model introduced by Kingma and Welling [22] consists of an encoder and a decoder, similar to an autoencoder, but the latent

This work is licensed under a Creative Commons Attribution 4.0 License. For more information, see https://creativecommons.org/licenses/by/4.0/

This article has been accepted for publication in IEEE Access. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2025.3582016

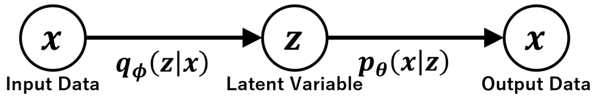


FIGURE 4. Overview of variational autoencoder model.

variables are normally distributed, and learning is conducted so that similar images have similar representations. Therefore, latent variables can be easily handled in a variational autoencoder.

A VAE model has the structure shown in Figure 4, in which $\boldsymbol{x}$ represents the input and output data and z represents the latent variables. Learning is conducted by maximizing equation (4), called the variational lower bound, for ϕ and $\boldsymbol{\theta}$ as the objective function.

$$\mathcal{L}(\boldsymbol{x};\phi,\boldsymbol{\theta}) = -D_{KL}\left(q_\phi(z\mid\boldsymbol{x})\|p(z)\right) + E_{q_\phi(z|\boldsymbol{x})}\left[\log p_{\boldsymbol{\theta}}(\boldsymbol{x}\mid z)\right] \quad (4)$$

where $q_\phi(z\mid\boldsymbol{x})$ is a probability distribution determined by parameter ϕ and can be regarded as an encoder, $p_{\boldsymbol{\theta}}(\boldsymbol{x}\mid z)$ is a probability distribution determined by parameter $\boldsymbol{\theta}$ and can be regarded as a decoder, ϕ is the weight parameter in the encoder (neural network), and $\boldsymbol{\theta}$ is the weight parameter in the decoder (neural network). The prior distribution $p(z)$ of latent variable z is the standard normal distribution.

In equation (4), the first term on the right is the inverted value of the Kullback-Leibler divergence (KL divergence) between $q_\phi(z\mid\boldsymbol{x})$ and $p(z)$; it represents the similarity between the two probability distributions. The smaller the KL divergence, the greater the similarity. The closer the latent variable z output from the encoder and the standard normal distribution $p(z)$, the larger the first term. The second term is the negative reconstruction error, which is larger the smaller the difference between input data $\boldsymbol{x}$ and output data $\boldsymbol{x}$.

Maximizing this variational lower bound results in latent variable z output from the encoder having a distribution close to the standard normal distribution and data $\boldsymbol{x}$ output from the decoder having a value close to the input data.

VAE has been used in various ways. For example, Esser et al. [23] proposed a method that combines a VAE-based tokenizer (VQGAN) with a transformer to enable high-resolution image synthesis. In their approach, the VAE model efficiently encodes images into discrete latent tokens, which are then modeled by the transformer to generate realistic and coherent images. Preechakul et al. [24] extended this idea by integrating a diffusion model into the autoencoder framework, separating the latent space into a semantic part and a stochastic part. This design enables both high-quality image reconstruction and semantically meaningful manipulation, overcoming the limitations of conventional VAE models in representation learning. Chen et al. [25] applied VAE to semi-supervised anomaly detection in multivariate time series. By integrating a long short-term memory network and a transformer into the VAE framework, their model captures both short- and long-term dependencies while effectively reconstructing normal patterns. This approach enables accurate detection and fine-grained localization of anomalies with limited labeled data.

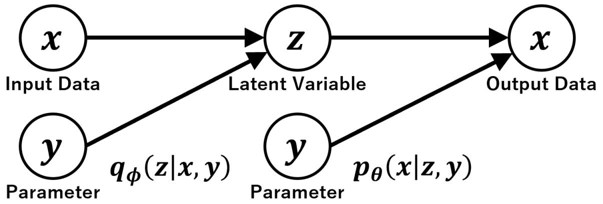


FIGURE 5. Overview of conditional variational autoencoder model.

The VAE has been improved in various ways. One shortcoming of VAE is that the generated images are often blurred. Therefore, Razavi et al. [26] devised VQ-VAE2, a method for generating high-resolution images by modifying VAE. VQ-VAE2 has two hierarchical latent variables, one with local information and one with global information. Suzuki et al. [27] devised a method for bidirectional conversion of multiple modalities as an improvement of VAE. A modality is a type of representation of a thing. As an example, the modality of a thing called the sea can be an image (image of the sea) or attribute information ([blue, sky, sand]). They demonstrated that training multiple encoders and decoders to have similar latent variables in both directions makes the transformation more effective than transformation in one direction.

2) Conditional variational autoencoder

The images generated by VAE vary depending on the latent variables, so VAE is rarely used directly for image generation. Instead, specific images are generated by applying VAE.

The CVAE model proposed by Sohn et al. [8] enables image generation based on specified parameters. The VAE model described in Section II-C1 has the structure shown in Figure 4, in which the input to the encoder is only image $\boldsymbol{x}$, and the input to the decoder is only latent variable z. In contrast, the CVAE model we used has the structure shown in Figure 5, in which an arbitrary number of parameters $\boldsymbol{y}$ are added to each input to the encoder and decoder in VAE.

The objective function of CVAE is represented by equation (5), and the learning process involves maximizing this function.

$$\mathcal{L}(\boldsymbol{x},\boldsymbol{y}) = -D_{KL}\left(q_\phi(z\mid\boldsymbol{x},\boldsymbol{y})\|p(z)\right) + E_{q_\phi(z|\boldsymbol{x},\boldsymbol{y})}\left[\log p_{\boldsymbol{\theta}}(\boldsymbol{x}\mid z,\boldsymbol{y})\right] \quad (5)$$

where $q_\phi(z\mid\boldsymbol{x},\boldsymbol{y})$ is a probability distribution determined by parameter ϕ and can be considered an encoder, and $p_{\boldsymbol{\theta}}(\boldsymbol{x}\mid z,\boldsymbol{y})$ is a probability distribution determined by parameter $\boldsymbol{\theta}$ and can be considered a decoder.

As with VAE, the first term on the right in equation (5) is the inverted value of the KL divergence between $q_\phi(z\mid\boldsymbol{x},\boldsymbol{y})$ and $p(z)$. The closer latent variable z output from the encoder

This work is licensed under a Creative Commons Attribution 4.0 License. For more information, see https://creativecommons.org/licenses/by/4.0/

This article has been accepted for publication in IEEE Access. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2025.3582016

and standard normal distribution $p(z)$, the larger the first term. The second term is the negative reconstruction error, which is larger the smaller the difference between input data $\boldsymbol{x}$ and output data $\boldsymbol{x}$.

Maximizing this variational lower bound results in latent variable z output from the encoder having a distribution close to the standard normal distribution and data $\boldsymbol{x}$ output from the decoder having a value close to the input data. The CVAE decoder is defined as $p_{\boldsymbol{\theta}}(\boldsymbol{x} \mid z, \boldsymbol{y})$, so, after training, it is possible to generate data $\boldsymbol{x}$ in accordance with parameter $\boldsymbol{y}$ by passing latent variable z and parameter $\boldsymbol{y}$ to the decoder.

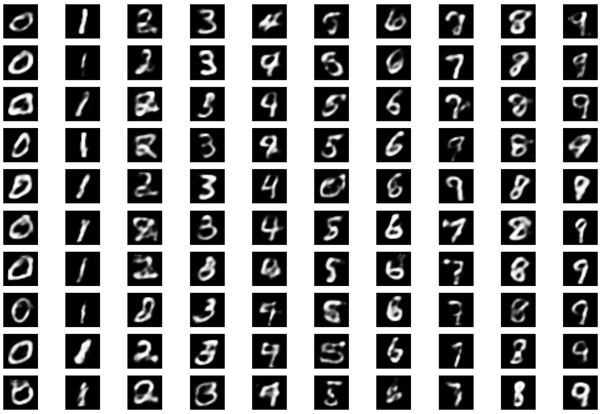


FIGURE 6. Results of image generation using CVAE decoder trained on MNIST. Each column corresponds to a digit label (value of parameter y), and each row corresponds to a latent variable z.

Figure 6 shows the result of generating images for each label by passing randomly generated latent variables z and numeric labels $\boldsymbol{y}$ to the decoder after training the CVAE model with MNIST images $\boldsymbol{x}$ and numeric labels $\boldsymbol{y}$ of the images.

We used the CVAE model to generate images in accordance with the specified item discrimination and item difficulty parameters, constructing a dataset.

D. DATASET SELECTION

As discussed above, traditional dataset selection methods in educational ML competitions often rely on arbitrary choices, leading to datasets that do not match participants' skill distributions. While manually curated datasets attempt to address this drawback, they require extensive human effort and are not scalable. IRT-based methods have been used to evaluate difficulty and discrimination parameters in pre-existing datasets, but they do not facilitate dataset construction as they rely heavily on human intervention. Although generative models like VAE ones have been used for dataset augmentation, they lack the capability to integrate evaluative criteria into dataset generation. Our proposed framework combines the strengths of IRT and CVAE to overcome these limitations. IRT enables precise estimation of dataset difficulty and discrimination, while CVAE automates dataset generation, ensuring that datasets are both well-structured and tailored to the target skill distribution. By integrating these methods, our approach eliminates the need for manual dataset preparation, improves ability assessment, and enhances adaptability across diverse educational applications.

III. PROPOSED FRAMEWORK

In this section, we present our framework for generating image datasets with high assessment ability on the basis of an existing image dataset for classification problems. The proposed framework is divided into two main phases: estimating the item discrimination and item difficulty parameters of each image in the existing image dataset (Section III-A) and generating images using the estimated parameters (Section III-B).

A. ESTIMATION OF ITEM DISCRIMINATION AND ITEM DIFFICULTY

In IRT, we consider examinees to be ML models and questions to be images in an image dataset and estimate the item discrimination and item difficulty parameters of each image.

1) Machine learning

To estimate the item discrimination and item difficulty parameters of each image in an existing image dataset, we need response data from multiple ML models. There are two types of response data: data we have prepared ourselves and response data from actual competitions. In both cases, the response data to the test data of the models generated by ML $\{d_{ij} \mid i \in \{1, \ldots, I\}, j \in \{1, \ldots, J\}\}$ are used, where i is the image number (question number), and j is the model number (examinee number). The test data are $\{\boldsymbol{v}_i, w_i \mid i \in \{1, \ldots, I\}\}$; the i-th image–label pair are $\boldsymbol{v}_i$ and w_i.

2) Estimation of item discrimination and item difficulty using IRT

Using the response data generated as described in Section III-A1, we estimate the item discrimination and item difficulty parameters of each image using equation (2) as the probability of the correct response. As explained in Section II-B3, the data used in this study is more biased than the test data on which IRT is usually used, so the estimated values may not converge. Therefore, we use MCMC for estimation. To accurately estimate the ability parameters of the ML model, this framework selects a subset of questions from the response data and uses MCMC to estimate the ability parameters; $\{\theta_j \mid j \in \{1, \ldots, J\}\}$ is estimated first. Fixing the ability parameters at the estimated values, we estimate the item discrimination and item difficulty parameters of each image $\{a_i, b_i \mid i \in \{1, \ldots, I\}\}$ in the test data.

B. GENERATION OF IMAGES

This section describes the training of the CVAE model, the generation of the item discrimination and item difficulty parameters to be passed to the decoder of the trained model, and the generation of images using the trained model.

This work is licensed under a Creative Commons Attribution 4.0 License. For more information, see https://creativecommons.org/licenses/by/4.0/

This article has been accepted for publication in IEEE Access. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2025.3582016

1) Training of CVAE model

We train the CVAE model using the image data of the test data $\{\boldsymbol{v}_i \mid i \in \{1, \ldots, I\}\}$ as input data $\boldsymbol{x}$ and $\{w_i, a_i, b_i \mid i \in \{1, \ldots, I\}\}$ as parameters $\boldsymbol{y}$ of the model. This training maximizes the following objective function.

$$\begin{aligned}\mathcal{L}(\boldsymbol{v}_i, [w_i, a_i, b_i]) = &-D_{KL}\left(q_{\boldsymbol{\phi}}(\boldsymbol{z}_i \mid \boldsymbol{v}_i, [w_i, a_i, b_i]) \| p(\boldsymbol{z_i})\right) \\ &+ E_{q_{\boldsymbol{\phi}}(\boldsymbol{z}_i|\boldsymbol{v}_i,[w_i,a_i,b_i])}\left[\log p_{\boldsymbol{\theta}}(\boldsymbol{v}_i \mid \boldsymbol{z}_i, [w_i, a_i, b_i])\right] \quad (6)\end{aligned}$$

where $q_{\boldsymbol{\phi}}(\boldsymbol{z}_i \mid \boldsymbol{v}_i, [w_i, a_i, b_i])$ is a probability distribution determined by parameter $\boldsymbol{\phi}$ and can be regarded as an encoder, and $p_{\boldsymbol{\theta}}(\boldsymbol{v}_i \mid \boldsymbol{z}_i, [w_i, a_i, b_i])$ is a probability distribution determined by parameter $\boldsymbol{\theta}$ and can be regarded as a decoder. Parameters $\boldsymbol{\phi}$ in the encoder (neural network) and $\boldsymbol{\theta}$ in the decoder (neural network) are the weight parameters. The prior distribution $p(\boldsymbol{z}_i)$ of latent variable $\boldsymbol{z}_i$ is the standard normal distribution.

2) Generation of item discrimination and item difficulty

We generate the item discrimination and item difficulty parameters $\{a', b' | k \in \{a'_k, b'_k \mid k \in \{1, \ldots, K\}\}$ to be passed to the CVAE decoder after training. To reduce bias in the average accuracy across images, these parameters are selected from a set of candidates sampled from a two-dimensional uniform distribution. Specifically, to ensure that the distribution of average probabilities of correct responses across images approximates a normal distribution, we sample a large number of (discrimination, difficulty) pairs from the uniform distribution, compute the corresponding average probabilities of correct responses, and select parameter pairs that produce the desired distribution.

The average probability of a correct response for each image k is computed using the generated discrimination a_k, difficulty b_k, and estimated ability parameters $\{\theta_j \mid j \in \{1, \ldots, J\}\}$ using the two-parameter logistic model (2PLM):

$$P_k = \frac{1}{J} \sum_{j=1}^{J} \frac{1}{1 + \exp(-Da_k(\theta_j - b_k))} \quad (7)$$

where P_k represents the average probability of a correct response for image k, across a set of virtual ML models characterized by the estimated ability parameters $\{\theta_j \mid j \in \{1, \ldots, J\}\}$.

3) Generating images using CVAE model

We pass to decoder $p_{\boldsymbol{\theta}}(\boldsymbol{v}_i \mid \boldsymbol{z}_i, [w_i, a_i, b_i])$, trained as described in Section III-B1, latent variables $\{\boldsymbol{z}_k \mid k \in \{1, \ldots, K\}\}$ generated from a standard normal distribution and parameter set $\{w'_k, a'_k, b'_k \mid k \in \{1, \ldots, K\}\}$, which consists of label data uniformly generated and item discrimination and item difficulty parameters generated as described in Section III-B2. This enables the generation of images with item discrimination and item difficulty based on IRT, generating a new dataset.

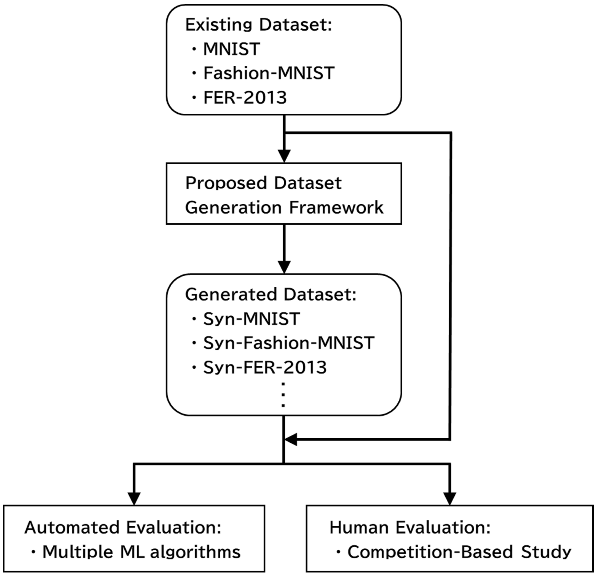


FIGURE 7. Process for evaluating proposed framework for generating synthetic datasets from existing datasets such as MNIST, Fashion-MNIST, and FER-2013. Generated datasets are evaluated using two approaches: automated evaluation using multiple ML models (Section IV-B, IV-C) and human evaluation based on a competition-based study (Section IV-D).

IV. EXPERIMENTS

To evaluate the effectiveness of the proposed framework, we conducted experiments using real data (the MNIST [4], Fashion-MNIST [5], and FER-2013 [28] datasets).

The datasets described in Section IV-A were generated separately from MNIST, Fashion-MNIST, and FER-2013. Those described in Section IV-B were generated using Syn-MNIST, Syn-Fashion-MNIST, and Syn-FER-2013 and were evaluated by comparing the performance of ML models on the generated datasets with that on the original datasets. A detailed analysis of the proposed framework is presented in Section IV-C. Specifically, we conducted experiments by modifying the input parameters of the CVAE model (Section IV-C1), altering the hyperparameters of the model (Section IV-C2), and replacing the model with a conditional deep convolutional generative adversarial network (CDCGAN) (Section IV-C3).

Section IV-D presents a user study designed to evaluate the datasets generated by the proposed framework within a ML competition format. Figure 7 illustrates the process for evaluating the proposed framework.

A. DATASET GENERATION USING PROPOSED FRAMEWORK

MNIST consists of 70,000 handwritten digit images ranging from 0 to 9, along with their corresponding numerical labels. Similarly, Fashion-MNIST consists of 70,000 fashion images belonging to 10 categories, also with their corresponding labels. Likewise, FER-2013 consists of 35,887 grayscale facial images categorized into 7 emotion classes, providing a benchmark for facial expression recognition tasks.

For the estimation of parameters in IRT within the proposed

This work is licensed under a Creative Commons Attribution 4.0 License. For more information, see https://creativecommons.org/licenses/by/4.0/

This article has been accepted for publication in IEEE Access. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2025.3582016

framework, we used self-prepared data. In this experiment, the MNIST and Fashion-MNIST datasets were each divided into training sets of 35,000 images and test sets of 35,000 images. Similarly, the FER-2013 dataset was divided into a training set of 17,943 images and a test set of 17,944 images. Training and testing were then conducted. We used a total of 810 ML models, which were automatically created by varying the hyperparameters in the MLPClassifier in scikit-learn. The estimated item discrimination and item difficulty by IRT resulted in the distributions shown in Figures 8 (a)–(c) for MNIST, Fashion-MNIST, and FER-2013, respectively. The CVAE model was trained on the estimated item discrimination and item difficulty of the images, along with their labels and images. Figure 9 shows the images generated for the number 9 by decoder $p_{\boldsymbol{\theta}}(\boldsymbol{v}_i \mid \boldsymbol{z}_i, [w_i, a_i, b_i])$ of the trained CVAE model.

The distributions of the item discrimination and item difficulty parameters, generated to ensure that the probability of correctness represented by equation (2) for each image follows a normal distribution, are shown in Figures 10 (a)–(c) for Syn-MNIST, Syn-Fashion-MNIST, and Syn-FER-2013, respectively. To match the number of images in the generated dataset to that in the original dataset, 70,000, 70,000, and 35,887 values, respectively, were generated for item discrimination and item difficulty. The datasets were generated by passing the item discrimination, item difficulty, and latent variables generated from a standard normal distribution to decoder $p_{\boldsymbol{\theta}}(\boldsymbol{v}_i \mid \boldsymbol{z}_i, [w_i, a_i, b_i])$ of the CVAE model after training.

The 3 generated datasets contained 70,000, 70,000, and 35,887 image–label pairs, matching the original MNIST, Fashion-MNIST, and FER-2013 datasets.

B. EVALUATION USING MACHINE LEARNING MODELS

To evaluate the generated datasets, the 810 automatically created ML models were trained and tested. For each of the MNIST, Fashion-MNIST, and FER-2013 datasets and the 3 derived datasets, the training was conducted with 35,000, 35,000, and 17,943 images, respectively, and the testing was conducted with 35,000, 35,000, and 17,944 images, respectively. The same 810 models used for preparing the response data were used for dataset evaluation.

Figure 11 plots the average accuracy distributions across models for the original MNIST, Fashion-MNIST, and FER-2013 datasets. Similarly, Figure 12 plots the distributions for Syn-MNIST, Syn-Fashion-MNIST, and Syn-FER-2013. Table 1 displays the average accuracy and variance for each dataset.

Comparison of Figures 11 and 12 shows that the generated datasets exhibited more gently peaked unimodal distributions of the average accuracy across models as well as greater variance. This indicates that the generated datasets better enhance participants' learning motivation in competitions and improve the effectiveness of ML model performance assessment than the original datasets.

TABLE 1. Average accuracy and variance for each dataset.

Dataset	Accuracy	Variance
MNIST	0.942	0.000604
Syn-MNIST	0.926	0.000682
Fashion-MNIST	0.860	0.000301
Syn-Fashion-MNIST	0.970	0.000483
FER-2013	0.346	0.002799
Syn-FER-2013	0.815	0.046872

For the datasets derived from Fashion-MNIST and FER-2013, which exhibited higher visual diversity compared with MNIST, CVAE may not fully capture and reproduce the variability inherent in the original data during the image generation process. As a result, the generated datasets may be simpler than the originals, potentially leading to the observed higher average accuracy than MNIST. In contrast, MNIST has relatively simple and homogeneous visual features, enabling CVAE to better preserve the characteristics of the original data. This may explain why the accuracy remained largely unchanged for the MNIST-derived datasets.

C. DETAILED ANALYSIS OF PROPOSED FRAMEWORK

1) Effect of Different CVAE Parameters

To evaluate the effectiveness of input parameters of the CVAE model within the proposed framework, we conducted three additional experiments using item difficulty and discrimination and accuracy as input parameters.

While the data used for training and generation in the proposed framework consists of {images, image labels} along with {discrimination, difficulty}, in the experiments, we trained the CVAE model, generated datasets, and evaluated the results using three different data patterns ({images, image labels} with either {discrimination}, {difficulty}, or {accuracy}), following the procedures described in Sections IV-A and IV-B.

For all patterns in these experiments, the images, image labels, discrimination, and difficulty data were the MNIST data used in the experiment described in Section IV-A. The latent variables were generated from a standard normal distribution, as described in Section IV-A. The accuracy values for the third pattern were derived from the experimental results presented in Section IV-A, using the actual accuracy for each image in the original MNIST dataset.

Figure 13 presents the distribution of average accuracy across ML models for each of the three data patterns. Table 2 presents the average accuracy and variance for each pattern. Compared with the MNIST and Syn-MNIST experiments described in Section IV-B, the accuracy values for all patterns were more concentrated at higher values, and the variance was smaller. These results indicate that using {discrimination, difficulty} as input parameters for the CVAE model is effective in generating datasets with enhanced assessment capability.

2) Effect of Latent Dimensionality on CVAE Model

To investigate how the hyperparameter settings of the CVAE model affect the quality of generated datasets and the perfor-

This work is licensed under a Creative Commons Attribution 4.0 License. For more information, see https://creativecommons.org/licenses/by/4.0/

This article has been accepted for publication in IEEE Access. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2025.3582016

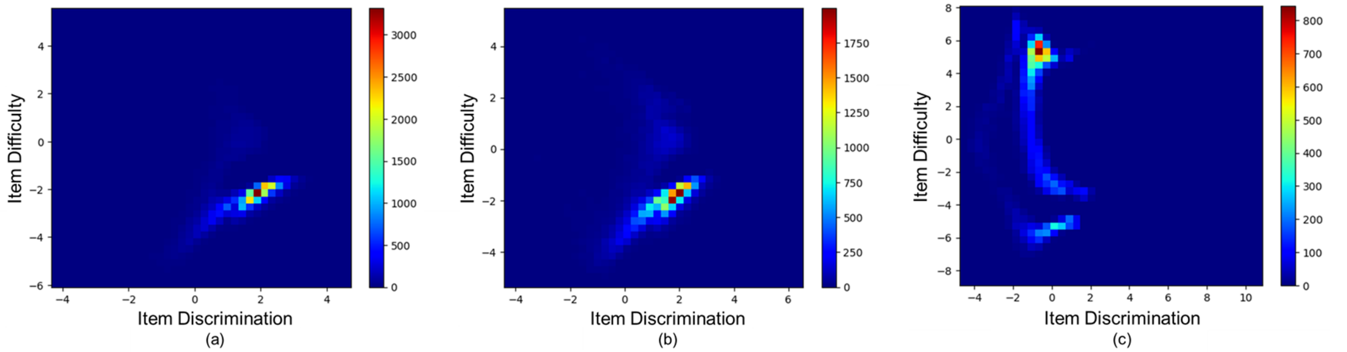


FIGURE 8. Distributions of item discrimination and item difficulty for each image within original datasets as estimated by IRT: (a) MNIST, (b) Fashion-MNIST, (c) FER-2013.

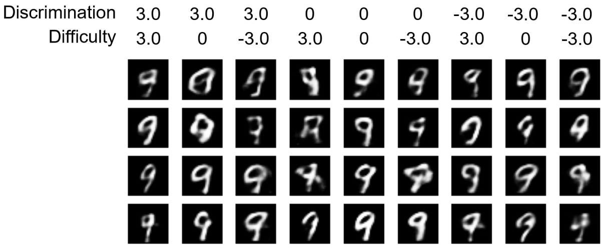


FIGURE 9. Images corresponding to digit label 9, generated by passing various combinations of item discrimination and item difficulty, ranging from $\{3, 0, -3\}$, to CVAE decoder $p_\theta(v_i \mid z_i, [w_i, a_i, b_i])$ after training.

TABLE 2. Average accuracy and variance across ML models on datasets generated with discrimination, difficulty, or accuracy as input parameters to CVAE model.

CVAE Input	Accuracy	Variance
{discrimination}	0.966	0.000090
{difficulty}	0.970	0.000101
{accuracy}	0.968	0.000099

mance of ML models, we conducted experiments in which we varied key hyperparameters.

In both VAE and CVAE models, the dimensionality of latent variables is a critical hyperparameter as it determines the representational capacity for data compression and reconstruction [22]. If the dimensionality is too low, the model may excessively compress the input information, failing to retain sufficient features. Conversely, if the dimensionality is too high, the model is prone to overfitting and unstable training behavior [29]. Therefore, selecting an appropriate dimensionality is essential for optimizing the quality of the generated data.

In our additional experiments, we focused on the dimensionality of the latent variables as the target hyperparameter. Specifically, we tested six dimensionality settings: {8, 16, 32, 64, 128, 256}. For each setting, we generated datasets and evaluated them using ML models. The original dataset used for CVAE training and generation was MNIST, and the data generation and evaluation procedures followed the methods described in Sections IV-A and IV-B. The default dimensionality used for generating the Syn-MNIST dataset was 32.

TABLE 3. Classification accuracy and variance of ML model on generated datasets with latent dimensionalities of 8, 16, and 32 (Syn-MNIST) and 64, 128, and 256.

Latent Dimensionality	Accuracy	Variance
8	0.996	0.000018
16	0.968	0.000225
32	0.926	0.000682
64	0.925	0.000797
128	0.950	0.000373
256	0.969	0.000227

Figure 14 plots the distribution of classification accuracy across ML models on the generated datasets, while Table 3 summarizes the average accuracy and variance. When the latent dimensionality was set to 32 or 64, the generated datasets exhibited higher variance in classification accuracy compared with the original MNIST dataset described in Section IV-B. These findings suggest that, with an appropriate dimensionality, the CVAE model can preserve critical features such as class separability and difficulty, while introducing meaningful diversity.

When the latent dimensionality was either relatively small (8 or 16) or relatively large (128 or 256), classification accuracies were more tightly clustered, indicating a diminished ability of the generated datasets to effectively assess classifiers.

For relatively small latent dimensionality (8 or 16), the generated images lacked diversity and frequently exhibited similar structures. This limitation arises from insufficient latent capacity, which prevents the model from effectively encoding the essential characteristics of the input data, resulting in overly compressed representations. Zhao et al. [30] pointed out that when the latent space is too small, the learned representations become limited, failing to capture the variability of the input data and thus producing overly uniform outputs. Consistent with this, our experiments suggest that low-dimensional latent spaces lead to simplified images that are easily classified, thereby reducing the dataset's capacity for effective assessment.

This work is licensed under a Creative Commons Attribution 4.0 License. For more information, see https://creativecommons.org/licenses/by/4.0/

This article has been accepted for publication in IEEE Access. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2025.3582016

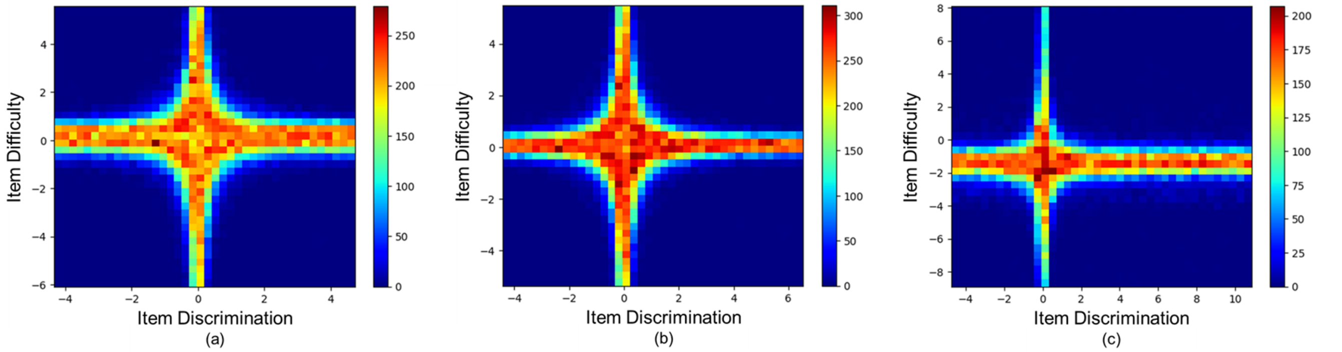


FIGURE 10. Distributions of item discrimination and item difficulty for each image, generated to follow a normal distribution of accuracy probabilities per image: (a) Syn-MNIST, (b) Syn-Fashion-MNIST, (c) Syn-FER-2013.

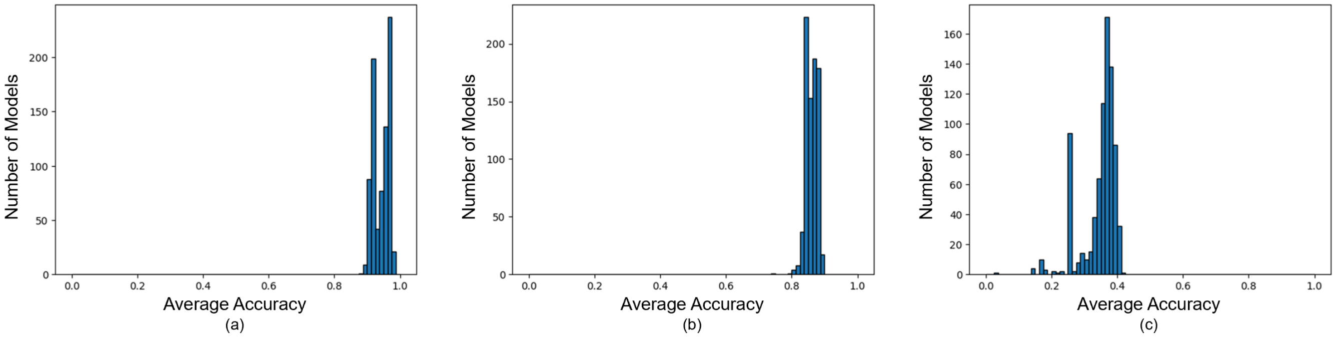


FIGURE 11. Average accuracy distribution across ML models on original datasets: (a) MNIST, (b) Fashion-MNIST, (c) FER-2013.

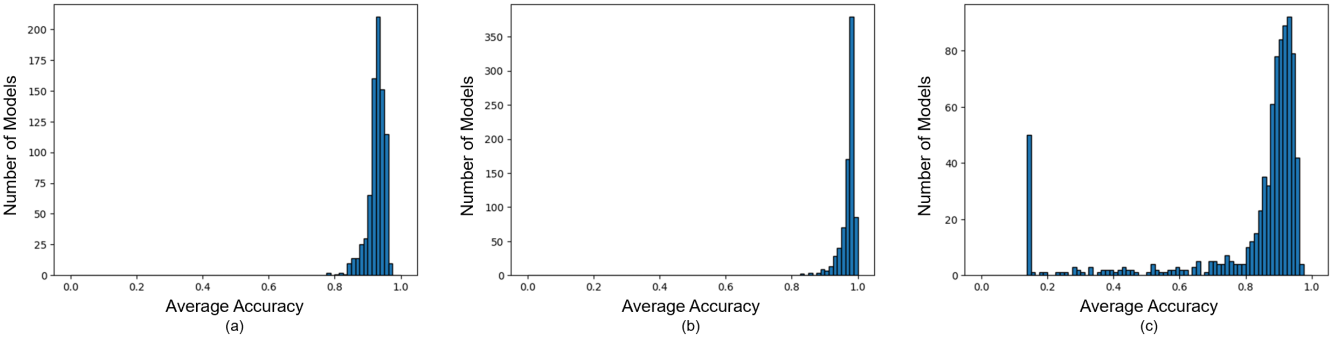


FIGURE 12. Average accuracy distribution across ML models on generated datasets: (a) Syn-MNIST, (b) Syn-Fashion-MNIST, (c) Syn-FER-2013.

In contrast, when the latent dimensionality was relatively large (128 or 256), the generated images contained substantial noise. This is likely due to redundant degrees of freedom in the latent space, which introduce irrelevant variations that do not reflect the underlying semantic structure of the data—such as handwriting styles or digit contours. Higgins et al. [29] observed that excessive latent dimensions in VAE models can result in some latent variables not contributing meaningfully to data generation, thereby increasing the risk of introducing noise. Moreover, according to Dai and Wipf [31], high-dimensional latent spaces can destabilize training and hinder the model from capturing the true latent structure, which can lead to visually ambiguous images and reduced variation in classifier evaluations.

In summary, a relatively small latent dimensionality leads to the loss of critical features and monotonous image generation, whereas a relatively large dimensionality introduces noisy and less meaningful variations. Therefore, selecting an appropriate latent dimensionality is essential for ensuring high-quality data generation while maintaining the evaluative utility of the generated datasets for ML models.

3) Comparison of Generation Models

To evaluate the effect of differences in generative models within the proposed framework, we conducted an additional experiment in which we replaced the CVAE model with a

This work is licensed under a Creative Commons Attribution 4.0 License. For more information, see https://creativecommons.org/licenses/by/4.0/

This article has been accepted for publication in IEEE Access. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2025.3582016

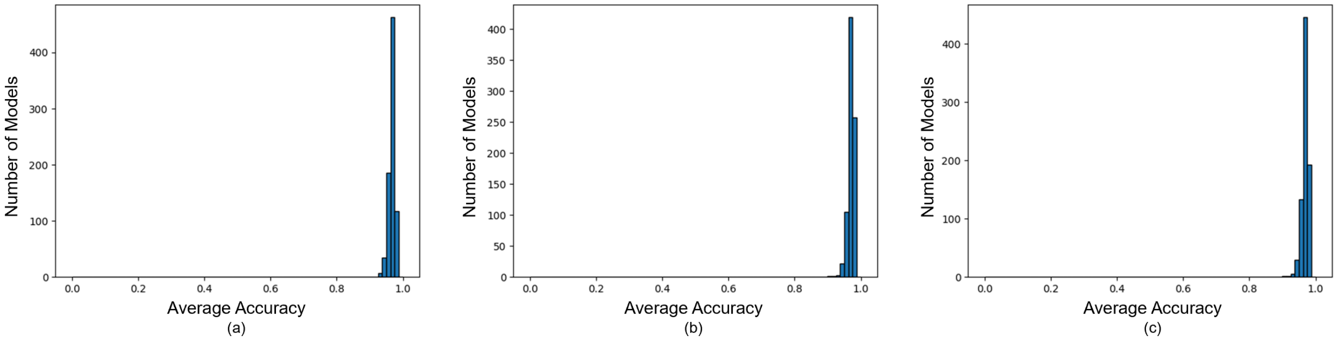


FIGURE 13. Average accuracy distribution across ML models on generated datasets with CVAE inputs of (a) {discrimination}, (b) {difficulty}, and (c) {accuracy}.

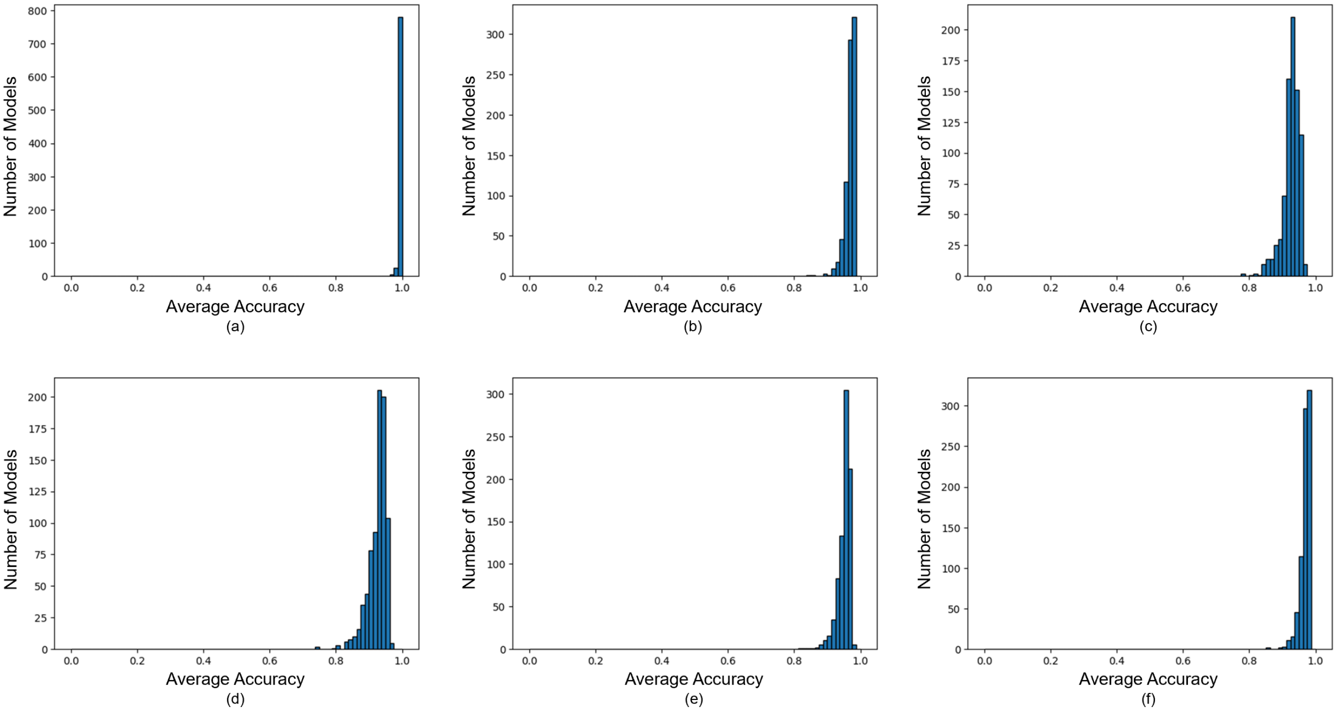


FIGURE 14. Classification accuracy distributions of ML models on generated datasets with latent dimensionalities of (a) 8, (b) 16, (c) 32 (Syn-MNIST); (d) 64, (e) 128, and (f) 256.

CDCGAN model.

In the original framework, the CVAE model was trained using images, image labels, discrimination, difficulty. In this experiment, we trained the CDCGAN model using the same input structure and the same training data as that used for the CVAE training described in Section IV-A. The experimental procedure followed the steps outlined in Sections IV-A and IV-B. The MNIST dataset served as the data source.

Figure 15 plots the distribution of accuracy across ML models for the CDCGAN-generated dataset. Despite using the same training data, significant differences were observed due to the choice of generative model. For the Syn-MNIST dataset generated by the CVAE model, the accuracy distribution exhibited a bell-shaped curve, whereas for the dataset generated by the CDCGAN model, the accuracy values were skewed toward higher values. A possible cause of this bias is mode collapse, a common issue with GAN-based models. When mode collapse occurs, the generator tends to produce highly similar images, failing to sufficiently reproduce the diversity of the original dataset.

Furthermore, the average accuracy and variance for the CDCGAN-generated dataset were 0.973 and 0.004620, respectively. The variance was higher than for the original MNIST dataset, which can be attributed to a few ML models exhibiting relatively low accuracy. As a result, the overall evaluation performance did not improve.

These results suggest that while CDCGAN is capable of generating realistic images, it has limitations in applications where maintaining dataset balance is crucial. Therefore, the selection and adjustment of the generative model in the pro-

This work is licensed under a Creative Commons Attribution 4.0 License. For more information, see https://creativecommons.org/licenses/by/4.0/

This article has been accepted for publication in IEEE Access. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2025.3582016

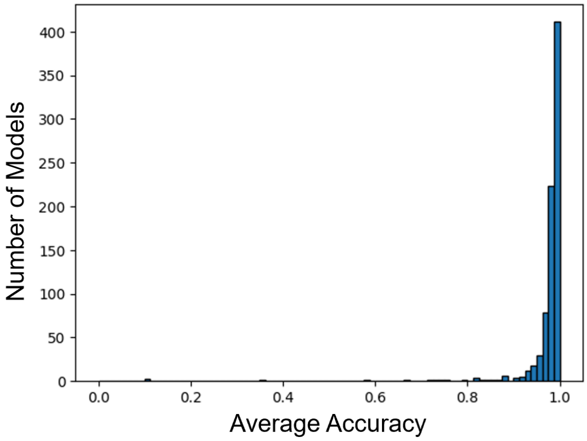


FIGURE 15. Average accuracy distribution across ML models on CDCGAN-generated dataset with MNIST dataset as data source.

posed framework play a vital role. Compared with the CDCGAN model, which is susceptible to mode collapse, the CVAE model demonstrated more stable dataset generation with higher assessment reliability.

D. PRACTICAL EVALUATION: COMPETITION-BASED VALIDATION

As discussed in Sections IV-B and IV-C, the assessment capabilities of the original dataset and the generated dataset were compared exclusively through training and testing using a fixed set of 810 ML models. To further evaluate these capabilities, we conducted a similar comparison in an experiment involving nine students majoring in information engineering, simulating an actual competition using MNIST and Syn-MNIST. All participants provided informed consent prior to participating in the experiments.

1) Experimental Procedure

Syn-MNIST and MNIST datasets were provided to the participants as "DatasetA" and "DatasetB," respectively. Each participant followed the experimental procedure outlined below, repeating Steps 1 and 2 a total of 30 times for each dataset.

Step1) Preparation of Machine Learning Model

Participant selects either DatasetA or DatasetB and prepares an ML model, including the hyperparameter selection, to maximize test accuracy on the chosen dataset.

Step2) Training and Testing of Machine Learning Model

Using the training and test data from the selected dataset, participant trains and tests the prepared model. After testing, the participant records the results and checks the test accuracy.

2) Experimental Results

The highest accuracies achieved by each participant are shown in Table 4. Syn-MNIST exhibited greater variance than MNIST, suggesting that it provides better support for difficulty adjustment.

TABLE 4. Highest test accuracy for each participant and variance.

Participant	MNIST	Syn-MNIST
1	0.974	0.987
2	0.804	0.781
3	0.975	0.984
4	0.980	0.991
5	0.921	0.869
6	0.976	0.988
7	0.973	0.987
8	0.978	0.990
9	0.963	0.984
Variance	0.0029	0.0050

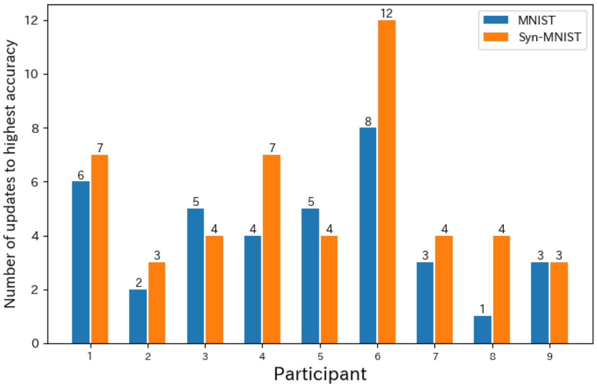


FIGURE 16. Comparison of number of updates to highest test accuracy for each participant.

To further analyze the test results and their variations, we compared the number of times the highest test accuracy was updated (i.e., improved) and the range of these updates across 30 trials for both Syn-MNIST and MNIST. The results are shown in Figures 16 and 17 and suggest that Syn-MNIST better reflects the trial-and-error process of each participant in their learning progression.

As shown in Figure 16, six of the nine participants exhibited a greater number of updates when using Syn-MNIST. As shown in 17, seven of the nine participants exhibited a greater range of updates when using Syn-MNIST. These results indicate that Syn-MNIST facilitates learning progression more effectively.

V. CONCLUSION

Our proposed framework constructs datasets suitable for evaluating machine learning model performance in competitions in which students and engineers studying ML participate. The item response probabilities from a two-parameter logistic model are used to generate item discrimination and item difficulty parameters, which are then passed to the decoder of a trained conditional variational autoencoder model, which generates the images for the target dataset.

The effectiveness of our framework depends on the provision of response data that accurately reflects the target competition level. For example, organizers may supply customized response data tailored to a specific competition, or, for regularly held competitions, utilize response data from past it-

This work is licensed under a Creative Commons Attribution 4.0 License. For more information, see https://creativecommons.org/licenses/by/4.0/

This article has been accepted for publication in IEEE Access. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2025.3582016

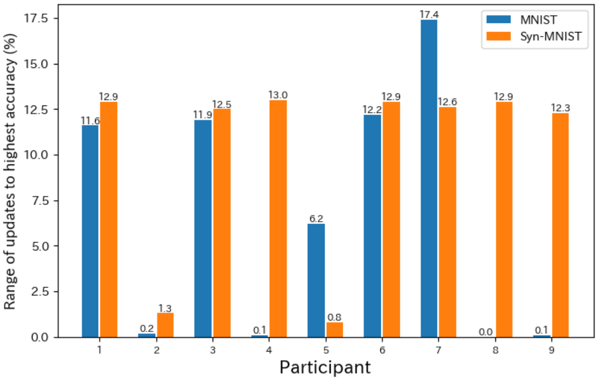


FIGURE 17. Comparison of range of updates to highest test accuracy for each participant.

erations. This approach ensures the generation of datasets that are highly effective for competency assessment and well-aligned with the anticipated skill level of the participants.

While our framework greatly enhances dataset generation for educational ML competitions, several avenues for future research remain. One promising direction is the refinement of the dataset generation process by incorporating a test information function to optimize the selection of item parameters, ensuring even more precise competency assessments. Additionally, a current limitation of our framework is the quality of the generated images, which tend to be blurrier compared with the original images. To address this limitation, integrating a more advanced generative model, such as a diffusion model, could improve the visual quality of individual images. Moreover, such integration could enable the generation of richer and more realistic datasets that better align with the skill distribution of learners across the dataset as a whole.

From a practical perspective, automating dataset generation through the integration of IRT and CVAE reduces the manual effort required to curate educational datasets. This advancement facilitates the broader adoption of AI-generated data in educational settings beyond ML competitions. For example, by ensuring that datasets are appropriately structured for competency assessment, our framework supports the development of personalized intelligent tutoring systems.

APPENDIX. HYPERPARAMETERS IN EXPERIMENTS

Table 5 shows the hyperparameters used for training the CVAE model.

TABLE 5. Hyperparameters used for training CVAE model.

Hyperparameter	Value
Batch size	256
Hidden dimension	64
Latent dimension	32
Number of epochs	30
Learning rate	1×10^{-3}

Table 6 shows the hyperparameter values used in our experiments (Section IV) for the MLPClassifier in scikit-learn. A total of 810 models were trained using diverse combinations from the listed values.

TABLE 6. Hyperparameter settings used in our experiments (Section IV) for MLPClassifier in scikit-learn.

Hyperparameter	Values
Alpha (L2 penalty term)	0.01, 0.001
Batch size	60, 120, 180
Initial learning rate	0.0001, 0.001, 0.01
Maximum no. of iterations	50, 100, 200
Activation function	identity, logistic, relu
Hidden layer sizes	(20), (40), (60), (80), (100)

REFERENCES

[1] Y. Baba, T. Takase, K. Atarashi, S. Oyama, and H. Kashima, "Data analysis competition platform for educational purposes: lessons learned and future challenges," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 32, no. 1, 2018.

[2] V. Kuleto, M. Ilić, M. Dumangiu, M. Ranković, O. M. Martins, D. Păun, and L. Mihoreanu, "Exploring opportunities and challenges of artificial intelligence and machine learning in higher education institutions," *Sustainability*, vol. 13, no. 18, p. 10424, 2021.

[3] H.-T. Chang and C.-Y. Lin, "Applying competition-based learning to stimulate students' practical and competitive AI ability in a machine learning curriculum," *IEEE Transactions on Education*, 2024.

[4] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proceedings of the IEEE (IEEE:1998)*, vol. 86, no. 11, pp. 2278–2324, 1998.

[5] H. Xiao, K. Rasul, and R. Vollgraf, "Fashion-MNIST: a novel image dataset for benchmarking machine learning algorithms," *arXiv preprint arXiv:1708.07747*, 2017.

[6] R. A. Fisher, "The use of multiple measurements in taxonomic problems," *Annals of eugenics (Ann. Eugen.:1936)*, vol. 7, no. 2, pp. 179–188, 1936.

[7] F. M. Lord, *Applications of item response theory to practical testing problems*. Routledge, 2012.

[8] K. Sohn, H. Lee, and X. Yan, "Learning structured output representation using deep conditional generative models," *Advances in neural information processing systems (NeurIPS:2015)*, vol. 28, 2015.

[9] L. Crocker and J. Algina, *Introduction to classical and modern test theory*. ERIC, 1986.

[10] N. Metropolis, A. W. Rosenbluth, M. N. Rosenbluth, A. H. Teller, and E. Teller, "Equation of state calculations by fast computing machines," *The journal of chemical physics (J. Chem. Phys.:1953)*, vol. 21, no. 6, pp. 1087–1092, 1953.

[11] H. J. Weerts, A. C. Mueller, and J. Vanschoren, "Importance of tuning hyperparameters of machine learning algorithms," *arXiv preprint arXiv:2007.07588*, 2020.

[12] J. N. Van Rijn and F. Hutter, "Hyperparameter importance across datasets," in *Proceedings of the 24th ACM*

This work is licensed under a Creative Commons Attribution 4.0 License. For more information, see https://creativecommons.org/licenses/by/4.0/

This article has been accepted for publication in IEEE Access. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2025.3582016

SIGKDD International Conference on Knowledge Discovery & Data Mining (KDD:2018), 2018, pp. 2367–2376.

[13] F. Hutter, H. Hoos, and K. Leyton-Brown, "An efficient approach for assessing hyperparameter importance," in *International conference on machine learning (ICML:2014)*. PMLR, 2014, pp. 754–762.

[14] L. Yang and A. Shami, "On hyperparameter optimization of machine learning algorithms: Theory and practice," *Neurocomputing (Neurocomputing:2020)*, vol. 415, pp. 295–316, 2020.

[15] S. Falkner, A. Klein, and F. Hutter, "Bohb: Robust and efficient hyperparameter optimization at scale," in *International conference on machine learning (PMLR:2018)*. PMLR, 2018, pp. 1437–1446.

[16] M. Uto and M. Ueno, "Item response theory for peer assessment," *IEEE Transactions on Learning Technologies*, vol. 9, no. 2, pp. 157–170, 2016.

[17] F. Samejima, "Estimation of latent ability using a response pattern of graded scores 1," *ETS Research Bulletin Series (ETS:1968)*, vol. 1968, pp. 1–169, 1968.

[18] M. Uto, Y. Tomikawa, and A. Suzuki, "Difficulty-controllable neural question generation for reading comprehension using item response theory," in *Proceedings of the 18th Workshop on Innovative Use of NLP for Building Educational Applications (BEA 2023)*, 2023, pp. 119–129.

[19] P. De Boeck and M. Jeon, "An overview of models for response times and processes in cognitive tests," *Frontiers in psychology rm(Front.Psychol:2019)*, vol. 10, p. 102, 2019.

[20] Z. Wang, W. Yan, C. Zeng, Y. Tian, and S. Dong, "A unified interpretable intelligent learning diagnosis framework for learning performance prediction in intelligent tutoring systems," *International Journal of Intelligent Systems (Int.J.Intell.Syst.:2023)*, vol. 2023, no. 1, p. 4468025, 2023.

[21] M. Uto, I. Aomi, E. Tsutsumi, and M. Ueno, "Integration of prediction scores from various automated essay scoring models using item response theory," *IEEE Transactions on Learning Technologies (IEEE:2023)*, vol. 16, no. 6, pp. 983–1000, 2023.

[22] D. P. Kingma and M. Welling, "Stochastic gradient vb and the variational auto-encoder," in *Second international conference on learning representations, (ICLR:2014)*, vol. 19, 2014, p. 121.

[23] P. Esser, R. Rombach, and B. Ommer, "Taming transformers for high-resolution image synthesis," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition (CVPR:2021)*, 2021, pp. 12 873–12 883.

[24] K. Preechakul, N. Chatthee, S. Wizadwongsa, and S. Suwajanakorn, "Diffusion autoencoders: Toward a meaningful and decodable representation," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition (CVPR:2022)*, 2022, pp. 10 619–10 629.

[25] N. Chen, H. Tu, X. Duan, L. Hu, and C. Guo, "Semisupervised anomaly detection of multivariate time series based on a variational autoencoder," *Applied Intelligence (Appl.Intell.)*, vol. 53, no. 5, pp. 6074–6098, 2023.

[26] A. Razavi, A. Van den Oord, and O. Vinyals, "Generating diverse high-fidelity images with vq-vae-2," *Advances in neural information processing systems (NeurIPS:2019)*, vol. 32, 2019.

[27] M. Suzuki, K. Nakayama, and Y. Matsuo, "Joint multimodal learning with deep generative models," *arXiv preprint arXiv:1611.01891*, 2016.

[28] I. J. Goodfellow, D. Erhan, P. L. Carrier, A. Courville, M. Mirza, B. Hamner, W. Cukierski, Y. Tang, D. Thaler, D.-H. Lee *et al.*, "Challenges in representation learning: A report on three machine learning contests," in *Neural information processing: 20th international conference, ICONIP 2013, Daegu, Korea, November 3-7, 2013. Proceedings, Part III 20 (ICONIP:2013)*. Springer, 2013, pp. 117–124.

[29] I. Higgins, L. Matthey, A. Pal, C. Burgess, X. Glorot, M. Botvinick, S. Mohamed, and A. Lerchner, "beta-vae: Learning basic visual concepts with a constrained variational framework," in *International conference on learning representations (ICLR:2017)*, 2017.

[30] S. Zhao, J. Song, and S. Ermon, "InfoVAE: Balancing learning and inference in variational autoencoders," in *Proceedings of the AAAI conference on artificial intelligence (AAAI:2019)*, vol. 33, no. 01, 2019, pp. 5885–5892.

[31] B. Dai and D. Wipf, "Diagnosing and enhancing VAE models," *International conference on learning representations (ICLR:2019)*, 2019.

This work is licensed under a Creative Commons Attribution 4.0 License. For more information, see https://creativecommons.org/licenses/by/4.0/

This article has been accepted for publication in IEEE Access. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2025.3582016

TAKEAKI SAKABE received the B.Eng. degree from the Nagoya Institute of Technology, in 2024, where he is currently pursuing the master's degree in computer science. His research interests include machine learning and human computation.

YUKO SAKURAI received her M.M.S. and Dr.Erg. degrees from Nagoya University and Kyushu University in 1997 and 2006, respectively. She is currently a professor in Department of Computer Science, Nagoya Institute of Technology . Her research interests include multi agent systems and game theory.

EMIKO TSUTSUMI received a Ph.D. degree from the University of Electro-Communications in 2023. She has been a project assistant professor at the Hosei University since 2024. Her research interests include e-learning, e-testing, machine learning, and data mining.

SATOSHI OYAMA received his B.Eng., M.Eng., and Ph.D. degrees from Kyoto University in 1994, 1996, and 2002, respectively. He is currently a professor in the Graduate School of Data Science, Nagoya City University. His research interests include machine learning and human computation. He is a member of IEEE.

This work is licensed under a Creative Commons Attribution 4.0 License. For more information, see https://creativecommons.org/licenses/by/4.0/