algorithms

MDPI

Article

AE-CTGAN: Autoencoder–Conditional Tabular GAN for Multi-Omics Imbalanced Class Handling and Cancer Outcome Prediction

Ibrahim Al-Hurani [1], Sara H. ElFar [1], Abedalrhman Alkhateeb [2,*] and Salama Ikki [1]

[1] Department of Electrical and Computer Engineering, Lakehead University, Thunder Bay, ON P7B 5E1, Canada; ialhura@lakeheadu.ca (I.A.-H.); shassan5@lakeheadu.ca (S.H.E.); sikki@lakeheadu.ca (S.I.)
[2] Department of Computer Science, Lakehead University, Thunder Bay, ON P7B 5E1, Canada
* Correspondence: aalkhate@lakeheadu.ca

Abstract

The rapid advancement of sequencing technologies has led to the generation of complex multi-omics data, which are often high-dimensional, noisy, and imbalanced, posing significant challenges for traditional machine learning methods. The novelty of this work resides in the architecture-level integration of autoencoders with Generative Adversarial Network (GAN) and Conditional Tabular Generative Adversarial Network (CTGAN) models, where the autoencoder is employed for latent feature extraction and noise reduction, while GAN-based models are used for realistic sample generation and class imbalance mitigation in multi-omics cancer datasets. This study proposes a novel framework that combines an autoencoder for dimensionality reduction and a CTGAN for generating synthetic samples to balance underrepresented classes. The process starts with selecting the most discriminative features, then extracting latent representations for each omic type, merging them, and generating new minority samples. Finally, all samples are used to train a neural network to predict specific cancer outcomes, defined here as clinically relevant biomarkers or patient characteristics. In this work, the considered outcome in the bladder cancer is Tumor Mutational Burden (TMB), while the breast cancer outcome is menopausal status, a key factor in treatment planning. Experimental results show that the proposed model achieves high precision, with an average precision of 0.9929 for TMB prediction in bladder cancer and 0.9748 for menopausal status in breast cancer, and reaches perfect precision (1.000) for the positive class in both cases. In addition, the proposed AE–CTGAN framework consistently outperformed an autoencoder combined with a standard GAN across all evaluation metrics, achieving average accuracies of 0.9929 and 0.9748, recall values of 0.9846 and 0.9777, and F1-scores of 0.9922 for bladder and breast cancer datasets, respectively. A comparative fidelity analysis in the latent space further demonstrated the superiority of CTGAN, reducing the average Euclidean distance between real and synthetic samples by approximately 72% for bladder cancer and by up to 84% for breast cancer compared to a standard GAN. These findings confirm that CTGAN generates high-fidelity synthetic samples that preserve the structural characteristics of real multi-omics data, leading to more reliable class balancing and improved predictive performance. Overall, the proposed framework provides an effective and robust solution for handling class imbalance in multi-omics cancer data and enhances the accuracy of clinically relevant outcome prediction.

Keywords: multiomics data integeration; autoencoder; GAN; CTGAN; cancer; multimodaility

Check for updates

Academic Editor: Pietro Hiram Guzzi

Received: 22 December 2025
Revised: 18 January 2026
Accepted: 20 January 2026
Published: 25 January 2026

Copyright: © 2026 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license.

1. Introduction

Technological advances in Next-Generation Sequencing (NGS) yield to generate various types of omics data, and omics platforms are becoming increasingly affordable. Continuous progress in NGS technologies has revolutionized genomics, enabling the generation of diverse omics data at unprecedented scale [1]. In general, the convergence of advances in NGS enabled comprehensive investigations into complex biological systems. Dimensionality Reduction (DR) techniques have been used to integrate multi-omics data to predict cancer outcome [2,3]. The Alghanim et al. model leverages the Synthetic Minority Oversampling TEchnique (SMOTE) to address class imbalance, combines it with a Support Vector Machine (SVM) classifier, and utilizes the Radial Basis Function (RBF) embedding model to integrate multi-omics data to predict menopausal status in breast cancer. The model applied principle component analysis (PCA) to extract the latent features of the omics, and then the model integrates the meaningful features into the SVM classifier [2]. Qattous et al. applied the Probabilistic Approximation model with Controlled Mapping (PACMAP) DR method, an embedding layer for multi-omics data integration, and PACMAP converts each omic into one colour in a 2D RGB representation. As a result of the RGB limitation, only three omics can be integrated. PCA is a linear DR technique that struggles to represent complex, nonlinear structures present in datasets. PACMAP is a probabilistic framework that aims to preserve the data's local and global structures; it involves some randomization in its algorithm, which may lead to generating different results each time it runs [3]. Al-Ghafer et al. proposed the Non-negative Matrix Factorization Gene Algorithm (NMF-GA) model that combines NMF-GA to efficiently extract meaningful patterns from high-dimensional datasets [4] to predict TMB, the total number of mutations per megabase of tumor DNA, which guides immunotherapy decisions levels in bladder cancer. The performance of the model was poor, achieving an accuracy of approximately 78%.

An autoencoder is an artificial neural network that learns efficient unsupervised data coding. Autoencoders have demonstrated superiority over traditional DR methods in certain scenarios due to their ability to capture complex, nonlinear relationships within data [5]. An autoencoder consists of two parts: the encoder and the decoder. The encoder generates reduced feature representations from initial inputs using hidden layers. The decoder reconstructs the original input from the output of the encoder by minimizing the loss function [6]. The autoencoder converts high-dimensional data to low-dimensional data. Therefore, the autoencoder is especially useful for noise removal, feature extraction, compression, and similar tasks [7]. As an application to dimensionality reduction for the classification of tumor types, Zhang et al. proposed using deep-embedded variational networks to extract the latent space from three omics, which are microRNA (miRNA) expression, Deoxyribonucleic Acid (DNA) methylation and gene expression using one Variational AutoEncoder (VAE) [8]. The latent space is the reduced dimensional space that captures the meaningful features of the original data [9].

In general, supervised learning algorithms face a challenge where one class has a large number of samples (majority class) and another class has fewer samples (minority class), which is known as class imbalance. Since prediction algorithms tend to favor the majority class, the results in such cases are questionable in terms of reliability and validity [10]. The healthcare sector is one of the domains where class imbalance is more common and has a greater impact on the overall performance of the models. Several techniques have been developed in the literature for dealing with imbalanced data. Synthetic oversampling methods generate new synthetic instances in order to balance a dataset, including SMOTE [11], ADAptive SYNthetic sampling (ADASYN) [12], GAN [13], and CTGAN [14].

SMOTE and ADASYN are data sampling algorithms based on class distributions. Both begin by identifying the minority class. SMOTE introduces class imbalance during the

k-nearest neighbor samples selection process, whereas ADASYN applies a revised version *k*-nearest neighbors samples based on a weighting scheme. Alghanim et al. compared both SMOTE and ADASYN to generate synthetic multi-omics data and concluded that SMOTE synthetic samples showed more ties to the original minority class distribution [2]. Unlike traditional data sampling algorithms, GAN learns to generate synthetic samples that resemble training data, while the discriminator learns to distinguish between real and generated samples. Through an iterative process, the generator and discriminator compete with each other, improving over time to generate more realistic samples [13]. Although GAN works well in upsampling imagery, audio, and text data, it suffers from dealing with the heterogeneous multi-sources of tabular data [15]. To be specific, when working with tabular data, we applied CTGAN to handle the various sources of omics data.

Earlier studies explored the use of deep generative models to create synthetic observations from binary omics data, such as single-nucleotide polymorphisms (SNP). The study evaluates three generative modelling approaches: VAEs, Deep Boltzmann Machine (DBM)s, and GANs [16]. While the model is based on omics data, it works only on one input of the data and has not been tested on multiple sources of data (multi-modals).

In this study, a latent space is extracted from each of the multi-omics datasets and merged into a shared latent space using an autoencoder. To handle imbalanced data classification, the minority class is up-sampled using GAN models, including standard GAN and CTGAN. The resulting shared latent space dataset feeds an artificial neural network that predicts cancer outcomes in two datasets.

2. Related Works

This adversarial process of GANs improves the quality of the generated data over time. In multi-omics research, GANs are used to generate synthetic datasets that can enhance predictive models. For example, OmicsGAN integrates multiple omics data and their interaction networks to generate synthetic data with better predictive signals [17]. However, traditional GANs can face issues such as mode collapse and training instability [18], which have been addressed by models like Wasserstein GANs. These incorporate the Wasserstein distance to enhance training stability [19]. Still, Wasserstein GANs were not specifically designed for tabular data [14]. The recent models in the literature are highlighted in Table 1.

Table 1. Comparison of generative models for cancer prediction.

Model	Year	Author	Key Contribution	Application	Performance
OmicsGAN	2022	Ahmed et al. [17]	Conditional GAN generating class-specific samples	Rare cancer subtype prediction	Improved recall/F1 vs. SMOTE/ ADASYN
CTGAN	2024	Kim et al. [20]	GTGAN applied to gene expression and survival data	12 cancer types (incl. BRCA)	Better prediction under imbalance
LASSO-MOGAT	2024	Alharbi et al. [21]	Graph-attention for multi-omics integration	Cancer outcome prediction	Handles high dimensionality and complexity
VAE-GAN	2021	Nußberger et al. [16]	Hybrid variational autoencoder and GAN model	SNP data	Effective with binary omics input
Precious2GPT	2024	Sidorenko et al. [22]	Combines diffusion models and transformers	Class imbalance scenarios	Higher quality than GANs/VAEs

VAEs are another type of generative model that encode input data into a latent space and then decode it back to the original space. This process allows VAEs to learn the underlying distribution of the data and generate new samples. In multi-omics research, VAEs are used for data integration and dimensionality reduction [23]. For instance, the Multi-Omics Variational autoEncoder (MOVE) framework integrates various omics data and clinical variables to identify cross-modal associations and improve data resolution [24]. VAEs are particularly useful for handling high-dimensional data and uncovering complex biological patterns [25,26]. These generative models offer powerful tools to overcome the challenges of complex, sparse, and high-dimensional multi-omics data, enabling new insights and advances in biological research [27].

Although recent work like the Variational AutoEncoder Generative Adversarial Network (VAE-GAN) [16] has explored the integration of autoencoders and GANs, standard GANs struggle with data that contains a mix of discrete and continuous features [14], while multiomics data has heterogeneous tabular data. This study addresses two key research gaps in the literature. First, existing approaches to class imbalance in multi-omics cancer datasets commonly rely on traditional oversampling techniques or apply generative augmentation directly in the original high-dimensional feature space, often leading to noise amplification and poor minority class representation. To address this limitation, our work employs an autoencoder to learn a compact and informative latent representation, enabling data augmentation to be performed in a reduced and denoised space. Second, the application of CTGAN to structured tabular biomedical multi-omics data, particularly within a learned latent space, remains largely unexplored. This study explicitly integrates CTGAN with autoencoder-derived latent features, leveraging its conditional sampling capability to generate high-quality synthetic minority class samples, thereby improving the effectiveness of data augmentation for imbalanced cancer outcome prediction.

Diffusion models are a newer class of generative models that simulate the process of data diffusion over time. They are particularly effective in modeling complex data distributions and generating high-quality synthetic data. In multi-omics research, diffusion models like Precious2GPT integrate pre-trained transformers with conditional diffusion to generate multi-omics, multi-species, and multi-tissue data. These models excel in generating representative synthetic data that captures tissue- and age-specific information, making them valuable for drug discovery and aging research [22]. While Precious2GPT is a powerful multimodal model, it is more generalized and not specifically optimized for tabular data.

Contributions

The main contributions of this paper are as follows:

- **Addressing class imbalance in multi-omics data:** We tackle the problem of *class imbalance*, which often biases machine learning models towards majority classes, by proposing a novel data augmentation pipeline.
- **Utilizing an Autoencoder for latent space extraction:** We employ an *autoencoder* to extract a compressed latent representation of the multi-omics datasets, effectively reducing dimensionality while preserving meaningful information.
- **Comparative analysis of GAN and CTGAN for upsampling:** We generate synthetic samples for the minority class from the latent space using both *standard GAN* and *CTGAN* to compare their effectiveness.
- **First application in cancer outcome prediction:** To the best of our knowledge, this is the *first study to apply CTGAN on autoencoder-derived latent spaces for cancer outcome prediction*, leveraging its conditional sampling ability to specifically generate high-quality minority class samples.

- **Evaluation using downstream neural network classification:** We assess the impact of generated samples by training a *neural network classifier* on the augmented data, using F1-score, precision, recall, and accuracy as performance metrics.
- **Demonstrating superior performance of AE + CTGAN pipeline:** Experimental results show that our proposed *AE + CTGAN approach significantly outperforms AE + GAN*, achieving higher F1-scores and recall, thus proving its effectiveness in improving minority class representation and classification performance.
- **Validation on two different datasets:** We validate our proposed technique on *two distinct publicly available multi-omics datasets: Breast Cancer (BRCA) (breast cancer) and Bladder Urothelial Carcinoma (BLCA) (bladder cancer)*, to demonstrate the robustness and generalizability of our approach.

3. Materials and Methods

As an initial step, we pre-processed the multi-omics dataset using an autoencoder to extract its latent space. Subsequently, we generated additional samples for the minority class to address the issue of class imbalance. A schematic representation of the proposed model is provided in Figure 1, with further details explained below as follows.

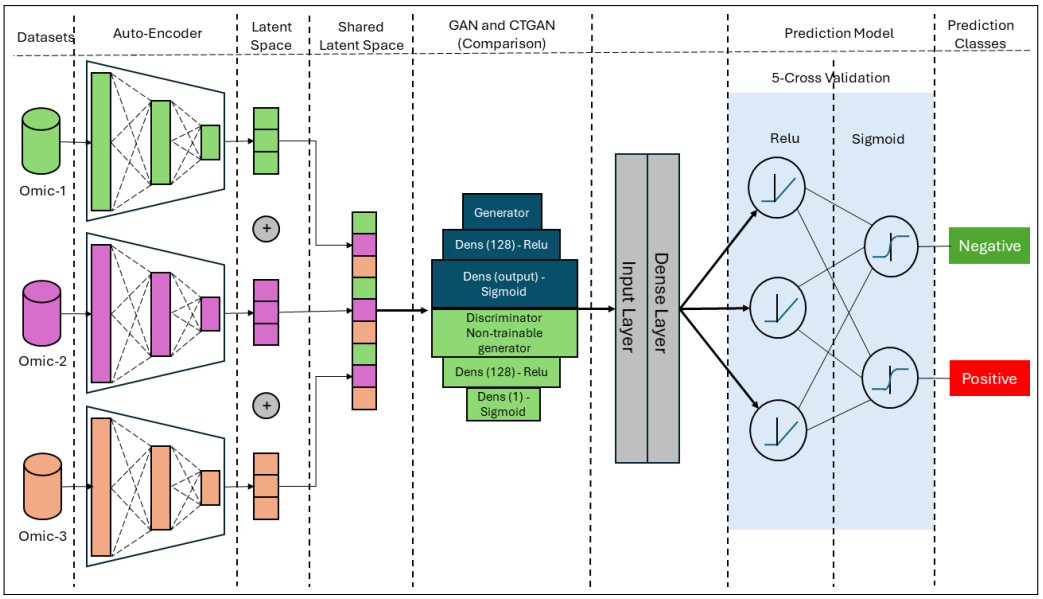


Figure 1. The workflow of the proposed model.

3.1. Materials

The proposed model was applied to two publicly available datasets. The first is a CTGAN BRCA dataset, which contains DNA methylation, Copy Number Alteration (CNA), and gene expression data [28,29] to predict the meno-pausal status (pre- versus post-). The second is a CTGAN BLCA dataset, which contains DNA methylation, CNA, and miRNA [30] to predict the TMB level (low- versus high-). From the distribution of the classes, as seen in Table 2, it shows that both datasets suffer from class-imbalance.

Table 2. Dataset metadata.

Dataset	Omics	Class Distribution
TCGA breast cancer (BRCA)	DNA methylation, CNA, and gene expression.	Pre-menopause = 89, Post-menopause = 255
TCGA bladder cancer (BLCA)	DNA methylation, CNA, and mRNA.	Low-TMB = 97, High-TMB = 107

3.2. Pre-Processing

The two dataset's features were all normalized using z-score. To select features with significant differences in means between the two classes, we performed a t-test with $p < 0.05$ (CI: 95%) for all features in the three omics. We then performed a fold-change test on the differentially expressed genes (DEG)s, a statistical method that measures how much a variable varies between two conditions or treatments. The logarithm of base two was then applied to the ratio to select features that were either doubled or halved between the two classes. The Benjamini–Hochberg test [31] was used to adjust the p-value in multiple hypothesis testing to control the FDR. In this study, the selected FDR value was 0.01, filtering out features with an adjusted p-value greater than that threshold.

3.3. Performance Measurements

To evaluate the classification models, we used the following performance metrics:

$$\text{Accuracy} = \frac{TN + TP}{TN + TP + FP + FN}, \tag{1}$$

$$\text{Recall} = \frac{TP}{TP + FN}, \tag{2}$$

$$\text{ROC} = \frac{P(x|\text{positive})}{P(x|\text{negative})}, \tag{3}$$

$$\text{Precision} = \frac{TP}{TP + FP}, \tag{4}$$

$$\text{F1} - \text{Score} = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}, \tag{5}$$

where TP is true positive, FP is false positive, TN is true negative, FN is false negative and $P(x|\text{positive})$ is probability of x given positive class.

3.4. Methods

3.4.1. Autoencoder

The encoder, implemented as a dense neural network layer with rectified linear unit (ReLU) activation function [32], compresses the input data into a lower-dimensional representation; this layer reduces the dimensionality of the input while capturing its latent space. The mathematical representation of the encoder can be written as

$$\text{Encoder} : \mathbf{h} = f_{\text{encoder}}(\mathbf{x}) = \phi(\mathbf{W}_1\mathbf{x} + \mathbf{b}_1), \tag{6}$$

where $\mathbf{x}$ is the input data, $\mathbf{h}$ is the encoded representation (latent space), f_{encoder} is the encoder function, $\mathbf{W}_1$ and $\mathbf{b}_1$ are the weights and biases of the encoder layer, respectively, and ϕ is a ReLU activation function, which returns the maximum between and 0.

The decoder, a dense neural network layer with Sigmoid activation, reconstructs the original input from the encoded representation. Hence, the mathematical representation of the decoder can be expressed as

$$\text{Decoder} : \mathbf{x}' = f_{\text{decoder}}(\mathbf{h}) = \phi(\mathbf{W}_2\mathbf{h} + \mathbf{b}_2), \tag{7}$$

where $\mathbf{x}'$ is the reconstructed output, f_{decoder} is the decoder function, and $\mathbf{W}_2$ and $\mathbf{b}_2$ are the weights and biases of the decoder layer, respectively.

Throughout training, the autoencoders minimize the MSE between the input and the output, ensuring accurate reconstruction. After training, the autoencoders are used to generate latent space representations for the respective input data types. During training,

the parameters $(\mathbf{W}_1, \mathbf{b}_1, \mathbf{W}_2, \mathbf{b}_2)$ are optimized to minimize the MSE loss between the input $(\mathbf{x})$ and the output $(\mathbf{x}')$. The MSE is calculated as

$$\mathcal{L}(\mathbf{x}, \mathbf{x}') = \frac{1}{N}\sum_{i=1}^{N}(\mathbf{x}_i - \mathbf{x}'_i)^2, \tag{8}$$

where N is the number of data points in the batch. A separate autoencoder was applied to each omic, and then the latent spaces generated from all of them were merged into one shared latent space.

3.4.2. Generative Adversarial Networks (GAN)

In this step, GAN was utilized to up-sample the minor class from the shared latent space samples. The generator G maps random noise z sampled from a prior distribution $p_z(z)$ to synthetic samples $\hat{x}$:

$$\hat{x} = G(z).$$

The discriminator D takes a sample x and outputs a probability $D(x)$ indicating the likelihood that x is a real sample. The generator network was constructed by first initializing a sequential model, that consists of

- Dense layer with 128 neurons.
- A ReLU activation function, which helps the model learn complex patterns in the data by allowing it to model non-linear relationships.
- Dense layer with a specific output dimension and a Sigmoid activation function is added to produce the final output, which generates synthetic samples.

The discriminator network was constructed using a sequential model, that consists of

- Dense layer with 128 neurons and ReLU activation function to process the input data.
- Dense layer with a single neuron and Sigmoid activation function is added to classify the input samples as real or fake.

These networks are key components of the GAN framework, where the generator creates synthetic samples, and the discriminator tries to differentiate between real and fake data. By fostering a competitive learning process, GAN boosts the generator's ability to produce alike realistic data. The generator network is defined as follows:

$$G(z) = \sigma(\phi(W_g z + b_g)), \tag{9}$$

where $G(z)$ represents the synthetic sample generated by the generator, z is the input noise vector, W_g and b_g are the weights and biases of the generator's dense layers, ReLU (ϕ) is the Rectified Linear Unit activation function, which is defined as $\phi(x) = \max(0, x)$, helping the model learn nonlinear patterns, and Sigmoid (σ) is the Sigmoid activation function, which squashes the output values between 0 and 1, suitable for generating data samples. For the discriminator network,

$$D(x) = \sigma(\phi(W_d x + b_d)). \tag{10}$$

The objective of the GAN is to train the generator to produce samples that are indistinguishable from real samples and to train the discriminator to distinguish between real and fake samples. This adversarial training process can be formulated as a minimax game:

$$\min_G \max_D V(D, G) = \mathbb{E}_{x \sim p_{\text{data}}(x)}[\log D(x)] + \mathbb{E}_{z \sim p_z(z)}[\log(1 - D(G(z)))], \tag{11}$$

where $\mathbb{E}$ denotes the expectation over real data x and noise z, $p_{\text{data}}(x)$ is the distribution of real data, and $p_z(z)$ is the prior distribution of noise.

Training Process

During training, the discriminator and generator were alternatively updated as the following:

- Discriminator Training: The discriminator is trained to maximize its ability to differentiate between real and fake samples. It was trained on both real samples x and generated samples $\hat{x}$, and its loss is calculated using binary cross-entropy:

$$\mathcal{L}_{\text{D}} = -(\log D(x) + \log(1 - D(\hat{x}))). \tag{12}$$

- Generator Training: The generator is trained to minimize the discriminator's ability to distinguish between real and fake samples. Its loss is calculated using the discriminator's output when fed with generated samples:

$$\mathcal{L}_{\text{G}} = -\log(D(G(z))). \tag{13}$$

Convergence in GAN is achieved when the generator produces data that the discriminator cannot distinguish from real data. This is theoretically when the Jensen–Shannon divergence between the real data distribution p_{data} and the generated data distribution p_G is minimized:

$$D_{\text{JS}}(p_{\text{data}} \| p_G) = \frac{1}{2}(D_{\text{KL}}(p_{\text{data}} \| M) + D_{\text{KL}}(p_G \| M)), \tag{14}$$

where $M = \frac{1}{2}(p_{\text{data}} + p_G)$, and D_{KL} is the Kullback–Leibler divergence.

3.4.3. Conditional Tabular Generative Adversarial Network (CTGAN)

CTGAN is designed to generate synthetic tabular data by addressing the challenges of imbalanced data and mixed data types. The generator G maps random noise z and conditional vector c sampled from a prior distribution $p_z(z)$ and $p_c(c)$ to synthetic samples $\hat{x}$: $\hat{x} = G(z, c)$. The discriminator D takes a sample x and outputs a probability $D(x)$ indicating the likelihood that x is a real sample: $[D(x)]$.

The generator network is constructed by first initializing a sequential model that consists of the following:

- Dense layer with 256 neurons.
- A Batch Normalization layer to stabilize and accelerate training.
- A Leaky ReLU activation function, which helps the model learn complex patterns in the data by allowing it to model nonlinear relationships.
- Dense layer with a specific output dimension and a Tanh activation function is added to produce the final output, which generates synthetic samples.

The discriminator network is constructed using a sequential model, that consists of

- Dense layer with 256 neurons and Leaky ReLU activation function to process the input data.
- Dense layer with a single neuron and Sigmoid activation function is added to classify the input samples as real or fake.

These networks are key components of the CTGAN framework, where the generator creates synthetic samples and the discriminator tries to differentiate between real and fake data. By fostering a competitive learning process, CTGAN boosts the generator's ability to produce realistic tabular data. The generator network is defined as follows:

$$G(z, c) = \tanh(\phi(W_g[z, c] + b_g)), \tag{15}$$

where $G(z, c)$ represents the synthetic sample generated by the generator, z is the input noise vector, c is the conditional vector, W_g and b_g are the weights and biases of the generator's dense layers, Leaky ReLU ϕ is the Leaky Rectified Linear Unit activation function, which is defined as $\phi(x) = \max(\alpha.x, x)$, helping the model learn nonlinear patterns with α is a tiny number and Tanh(tanh) is the Tanh activation function, which squashes the output values between -1 and 1, suitable for generating data samples.

For the discriminator network,

$$D(x) = \sigma(\phi(W_d x + b_d)). \tag{16}$$

The objective of the CTGAN is to train the generator to produce samples that are indistinguishable from real samples and to train the discriminator to distinguish between real and fake samples. This adversarial training process can be formulated as a minimax game:

$$\min_G \max_D V(D, G) = \mathbb{E}_{x \sim p_{\text{data}}(x)}[\log D(x)] + \mathbb{E}_{z \sim p_z(z), c \sim p_c(c)}[\log(1 - D(G(z, c)))], \tag{17}$$

where $\mathbb{E}$ denotes the expectation over real data x, noise z, and conditional vector c, $p_{\text{data}}(x)$ is the distribution of real data, $p_z(z)$ is the prior distribution of noise, and $p_c(c)$ is the prior distribution of the conditional vector.

Training Process

During training, the discriminator and generator are alternatively updated as the following:

- Discriminator Training: The discriminator is trained to maximize its ability to differentiate between real and fake samples. It is trained on both real samples x and generated samples $\hat{x}$, and its loss is calculated using binary cross-entropy:

$$\mathcal{L}\text{D} = -(\log D(x) + \log(1 - D(\hat{x}))). \tag{18}$$

- Generator Training: The generator is trained to minimize the discriminator's ability to distinguish between real and fake samples. Its loss is calculated using the discriminator's output when fed with generated samples:

$$\mathcal{L}\text{G} = -\log(D(G(z, c))). \tag{19}$$

CTGAN address the unique challenges of tabular data using techniques including conditional sampling and balancing utility against disclosure risk. This results in more stable and efficient convergence compared to traditional GANs. The convergence can be analyzed using the conditional Jensen–Shannon divergence:

$$D_{\text{JS}}(p_{\text{data}}(\mathbf{x}|\mathbf{c}) \| p_G(\mathbf{x}|\mathbf{c})) = \frac{1}{2}(D_{\text{KL}}(p_{\text{data}}(\mathbf{x}|\mathbf{c}) \| M(\mathbf{x}|\mathbf{c})) + D_{\text{KL}}(p_G(\mathbf{x}|\mathbf{c}) \| M(\mathbf{x}|\mathbf{c})), \tag{20}$$

where

$M(\mathbf{x}|\mathbf{c}) = \frac{1}{2}(p_{\text{data}}(\mathbf{x}|\mathbf{c}) + p_G(\mathbf{x}|\mathbf{c}))$.

3.4.4. Prediction Model

Firstly, we split the data into training and testing sets based on a 5-fold cross-validation, which was used to construct the prediction model. A neural network was built of two dense layers. The first is with 128 neurons and the ReLU activation function, and the other is with 1 neuron and the Sigmoid activation function. The model utilizes Adam optimizer [33] and binary cross-entropy loss function to optimize the accuracy. The model was trained with parameters specifying the number of training epochs of 10, batch size of 32, and a validation split of 0.2. The Sigmoid function returns the probability of the class, which represents the predicted class. To avoid overfitting and underfitting, the hyper-parameters were optimized, and the training versus validation of the models were plotted and noticed to keep monitoring the proper performance of the model. Early stoppage technique was used to stop earlier than the point where the training performance trends are different than the validation performance of the models.

4. Results

The hyper-parameters of the models were optimized and selected to select the best performance without overfitting or under fitting as listed in Table 3.

Table 3. Models and hyperparameters.

Model	Hyperparameters: Values
Omics Autoencoders	epochs: 30 batch_size: 64 latent space dimensions: 128
GAN and CTGAN	epochs: 300 batch_size: 500 α: 0.01
Prediction Model (Neural Network)	hidden layer units: 128 activation function: relu dropout rate: 0.2 output layer units: 1 optimizer: adam loss function: binary_crossentropy epochs: 30 (max, limited by early stopping) batch_size: 32 early stopping patience: 5
k-Fold Cross-Validation	n_splits: 5

Table 4 presents the performance measurements of applying AutoEncoder (AE) with CTGAN and AE with GAN for the BLCA dataset. The results demonstrate that AE with CTGAN significantly outperforms AE with GAN across all metrics. AE with CTGAN achieves an average accuracy of 0.9929, a perfect precision of 1.0000, recall of 0.9846, and an average F1-score of 0.9922. In contrast, AE with GAN achieves an average accuracy of 0.88827, with a precision of 0.85417, recall of 0.75926, and an average F1-score of 0.80392.

Table 5 presents the performance measurements of applying AE with CTGAN and AE with GAN for the BRCA dataset. The results show that both models achieve high accuracy, with AE with CTGAN demonstrating a better average accuracy of 0.95098 compared to AE with GAN. Both models exhibit perfect precision (1.0000). AE with CTGAN achieves a recall of 0.9777, while AE with GAN achieves a recall of 0.81481, suggesting that AE with CTGAN is better at identifying all true positive instances. In terms of the average F1-score,

AE with CTGAN achieves 0.9922, further supporting the superior performance of AE with CTGAN in capturing both precision and recall.

From both Tables 4 and 5, the strength of AE with CTGAN in precision in both cancer outcomes suggests that all predicted positive instances are indeed positive. Generally, predicting the positive class, often representing adverse events including relapse [34] or metastasis [35], is a key focus in cancer outcome prediction due to its significant clinical implications. The overall performance suggests that AE with CTGAN is a more effective model for generating high-quality synthetic data for the BLCAdataset.

Table 4. Performance measurements of applying AE with CTGAN and AE with GAN for the BLCA dataset.

Classifier	Average Accuracy	Precision	Recall	Average F1-Score
AE with GAN	0.8882	0.8541	0.7592	0.8039
AE with CTGAN	0.9929	1.0000	0.9846	0.9922

Table 5. Performance measurements of applying AE with CTGAN and AE with GAN for the BRCA dataset.

Classifier	Average Accuracy	Precision	Recall	Average F1-Score
AE with GAN	0.9509	1.0000	0.8148	0.8979
AE with CTGAN	0.9748	1.0000	0.9777	0.9922

Figures 2 and 3 show the area under the receiver operating characteristic (AUROC) curves for applying AE with GAN versus AE with CTGAN models on both BLCA and BRCA datasets. The comparison shows that AE with CTGAN curves tend more to the north-west, covering more areas than the other curve on multiple validation running points. This indicates a dominance in better determining true positive rates while minimizing false positives. Consequently, AE with CTGAN demonstrates superior performance in distinguishing between classes, making it a more effective model for both datasets.

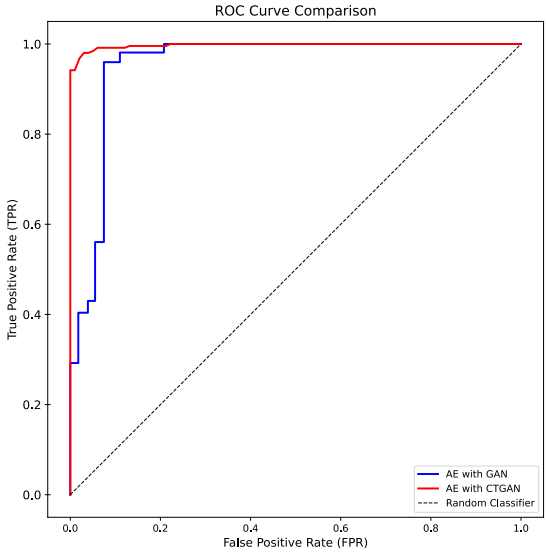


Figure 2. ROC curve for the BRCA dataset.

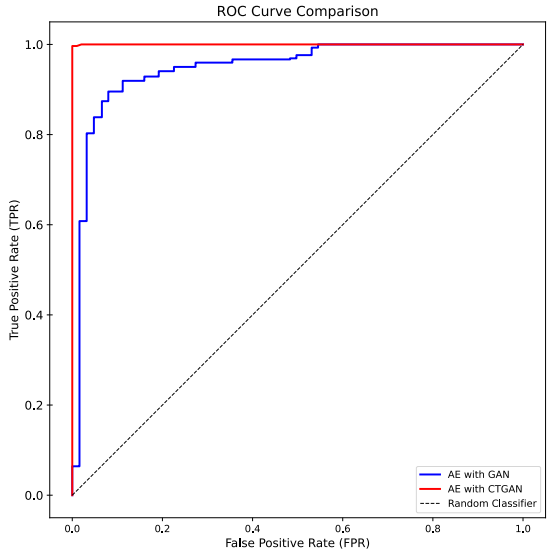


Figure 3. ROC curve for the BLCA dataset.

To validate the best performing model that is AE with CTGAN, the training versus validation loss for the model on the BLCA dataset as seen in Figure 4 and on the BRCA dataset as seen in Figure 5. The figures show a clear downward trend in both training and validation loss, which is generally a positive sign. It indicates that the model is learning and improving its performance over time. The similar trends for both training and validation curves in both figures defy overfitting. The gap between the training and validation loss is relatively small, especially in the initial epochs. This suggests that the model is generalizing well to unseen data and not overfitting significantly in the early stages.

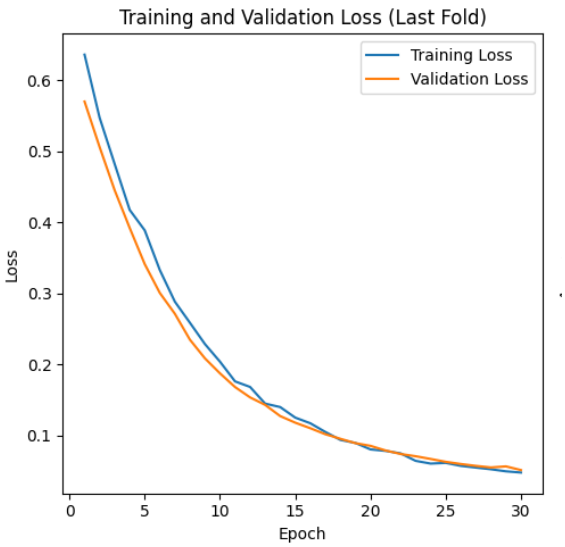


Figure 4. Training vs. validation loss for AE with CTGAN on BLCA dataset.

To evaluate the quality of the synthetic data beyond its predictive utility, we conducted a comparative fidelity analysis between the standard Generative Adversarial Network (GAN) and the Conditional Tabular GAN (CTGAN). In this context, fidelity was quantified by calculating the average Euclidean distance between the generated synthetic samples and their nearest neighbors in the original latent space. A lower Euclidean distance signifies that the synthetic data points are more representative of the real data distribution, maintaining the structural integrity of the original multi-omics features.



Figure 5. Training vs. validation loss for AE with CTGAN on BRCA dataset.

As shown in Table 6, the CTGAN architecture demonstrated a significantly higher degree of fidelity compared to the standard GAN. Specifically, for the BLCA dataset, CTGAN reduced the average Euclidean distance by approximately 72%. For the BRCA dataset, the performance gap was even more substantial, with CTGAN achieving a distance of 0.5013, outperforming the standard GAN by a factor of six.

These findings suggest that the specialized components of CTGAN, such as mode-specific normalization and the conditional generator are better equipped to navigate the non-Gaussian and sparse distributions typically found in multi-omics latent representations. The high fidelity of the CTGAN-generated samples ensures that the synthetic data used for class balancing is not merely "noise" but a high-quality approximation of the minority class. This structural similarity to real data provides a robust foundation for the downstream classifier, explaining the high precision (1.000 for the positive class) achieved in our final predictive models.

Table 6. Fidelity Comparison: Average Euclidean distance between real and synthetic samples in latent space.

Generative Model	BLCA (TMB)	BRCA (Menopausal Status)
AE with GAN	3.9600	3.1112
AE with CTGAN	1.0950	0.5013

5. Discussion

Figure 6 illustrates the analysis of latent space of AE in the BRCA data set, comprising two components: Figure 6A Correlation Heatmap displays pairwise correlations between 45 latent features (indices 0–44) in the bottleneck layer. The color gradient (from -1 to $+1$) reveals feature dependencies, where red indicates strong positive correlations, blue indicates negative correlations, and white denotes independence. Block-like patterns suggest clustered feature interactions, while sparse correlations imply disentangled representations. Figure 6B Feature Importance ranks latent features by their contribution (weight magnitude) to reconstruction. Dominant features (peaks) encode critical data patterns, while low-weight features may represent noise or redundant information. This analysis validates the AE's ability to compress input data into meaningful, non-redundant latent dimensions.

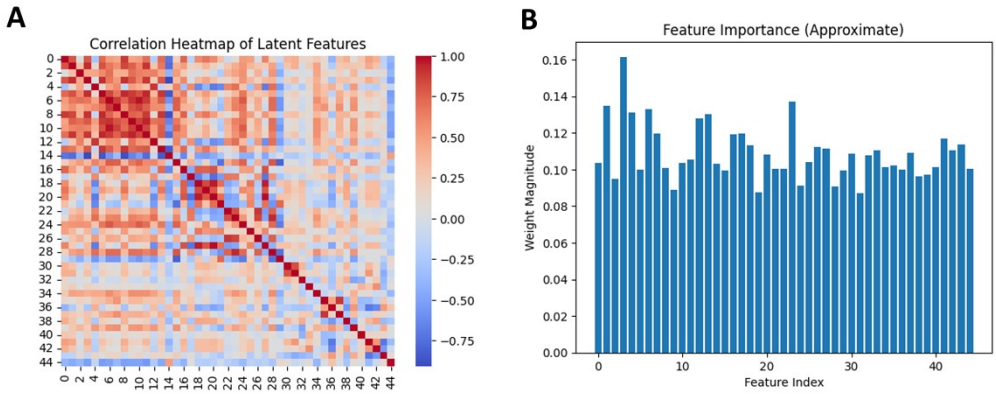


Figure 6. Applying AE to the BRCA datast: (**A**) Correlation heatmap of latent features in the Autoencoder's bottleneck layer (scale: -1 to $+1$). (**B**) Approximate feature importance ranking based on weight magnitudes for the extracted latent features.

The proposed model demonstrates superior performance when compared with previously reported machine learning approaches in the literature. For bladder cancer (BLCA), the NMF-guided feature selection and genetic algorithm-based framework utilizing a convolutional neural network proposed by Al-Ghafer et al. [4] achieved a precision of 0.7789, whereas the proposed model attained a precision of 1.0000. Similarly, for breast cancer (BRCA), the Support Vector Machine with radial basis function (RBF) kernel trained on SMOTE-synthesized samples reported by Alghanim et al. [2] achieved a precision of 0.919, compared to a precision of 1.0000 obtained by the proposed model.

GANs were initially developed for generating images and other forms of continuous data. Their architectures and training strategies are often optimized for these types of data [36]. CTGAN is designed to generate synthetic tabular data, an improvement over standard GANs [14]. CTGAN typically uses different network architectures for the generator and discriminator compared to standard GANs. These architectures are often tailored to the specific characteristics of tabular data, such as the presence of categorical features and the need to capture complex relationships between variables [14].

Both BLCA and BRCA datasets contain three omics, but applying this model on datasets with more or fewer omics is not restricted, as each omic will undergo the same workflow before integration. Meanwhile, AE with CTGAN demonstrated promising results in handling multi-omics data by extracting the latent space and generating synthetic data.

Several limitations warrant consideration. Firstly, the model's interpretability is limited. Understanding the specific features driving the model's predictions and gaining deeper biological insights remains challenging. Secondly, the model's generalizability needs further validation. Applying the model to predict various cancer outcomes and other diseases would provide a more comprehensive assessment of its performance and robustness. Finally, while the model outperforms previous models in the literature [2,4], it still needs rigorous comparisons with deep learning baseline models.

The correlation among the extracted latent space features from the BRCA dataset that can be seen in Figure 6A suggests that most of these features are uncorrelated; however, some dark-red areas suggest that some of these captured latent space features can be removed to extract only the unique latent features. While the proposed model works near perfection on both datasets, the strongly correlated latent features may affect the performance on more complex data. Addressing these limitations is crucial for future work. Techniques such as feature importance analysis and model-agnostic interpretation methods could be explored to enhance interpretability. Additionally, applying the model to diverse cancer datasets with different characteristics will broaden its applicability and

provide valuable insights into its strengths and weaknesses. Furthermore, the potential of this approach can be explored beyond the omics data. Investigating its performance on other types of tabular multi-modal data domains could reveal its broader applicability and extraction power in diverse fields.

6. Conclusions

This study demonstrates significant advancements in predictive accuracy for cancer outcome prediction by integrating feature selection, autoencoder-based dimensionality reduction, and GAN-based data augmentation into a unified pipeline, achieving average accuracies of 0.9929 and 0.9748, recall values of 0.9846 and 0.9777, and F1-scores of 0.9922 for the bladder and breast cancer datasets, respectively. Our approach begins with selecting the most discriminative features, followed by using autoencoders to generate a compact shared latent space representation of multi-omics data. which contributed to stable and consistent performance across all evaluated datasets.

A key contribution of our model is the incorporation of CTGAN for synthetic data generation. Unlike standard GANs, CTGAN leverages conditional generation, allowing it to specifically target and augment the minority class samples. This capability effectively mitigates class imbalance, which is a critical challenge in cancer datasets, and results in improved recall and F1-score for minority class predictions.

Furthermore, our approach focuses on clinically meaningful outcomes such as TMB and menopausal status, ensuring their relevance and applicability to precision medicine, where these predicted outcomes can directly support patient stratification and risk assessment. The model can be used in academic medical centers and research hospitals to analyze multi-omics data, explore biomarker associations, and evaluate predictive models before clinical deployment. The model can also be applied to different multi-model datasets in biomedical analysis and other fields. This sets our work apart from existing studies that often emphasize only technical model improvements without addressing specific clinical needs.

Experimental results on publicly available datasets demonstrate the effectiveness of our proposed pipeline, achieving superior accuracy, precision, recall, and F1-score compared to existing methods. Overall, this study shows that combining AE and CTGAN not only addresses class imbalance and high-dimensional data challenges but also enhances predictive capabilities in cancer outcome prediction, contributing to the advancement of personalized healthcare.

Author Contributions: I.A.-H. and A.A., idea conceptualization; I.A.-H., S.I. and A.A., methodology; I.A.-H., S.I. and S.H.E., validation; S.I. and A.A., supervision; A.A., management of the work; A.A., funding. All authors contributed equally in writing the manuscript. All authors have read and agreed to the published version of the manuscript.

Funding: This paper was partially supported by the Provost's Office at Lakehead University.

Data Availability Statement: The preprocessed data and the code can be accessed via the following repository: https://github.com/Ibrahimalhurani/Dataset_and_Code.git, accessed on 15 January 2025.

Acknowledgments: During the preparation of this study, the authors used ChatGPT, based on GPT-5.2, for the purposes of proofreading. The authors have reviewed and edited the output and take full responsibility for the content of this publication.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Wang, Q.; Peng, W.X.; Wang, L.; Ye, L. Toward multiomics-based next-generation diagnostics for precision medicine. *Pers. Med.* **2019**, *16*, 157–170. [CrossRef] [PubMed]
2. Alghanim, F.; Al-Hurani, I.; Qattous, H.; Al-Refai, A.; Batiha, O.; Alkhateeb, A.; Ikki, S. Machine Learning Model for Multiomics Biomarkers Identification for Menopause Status in Breast Cancer. *Algorithms* **2023**, *17*, 13. [CrossRef]
3. Qattous, H.; Azzeh, M.; Ibrahim, R.; Al-Ghafer, I.A.; Al Sorkhy, M.; Alkhateeb, A. PaCMAP-embedded convolutional neural network for multi-omics data integration. *Heliyon* **2024**, *10*, e23195 . [CrossRef]
4. Abed Al-Ghafer, I.; AlAfeshat, N.; Alshomali, L.; Shaheen, A.; Qattous, H.; Azzeh, M.; Alkhateeb, A. NMF-guided feature selection and genetic algorithm-driven framework for tumor mutational burden classification in bladder cancer using multi-omics data. *Netw. Model. Anal. Health Inform. Bioinform.* **2024**, *13*, 26. [CrossRef]
5. Berahmand, K.; Daneshfar, F.; Salehi, E.S.; Li, Y.; Xu, Y. Autoencoders and their applications in machine learning: A survey. *Artif. Intell. Rev.* **2024**, *57*, 28. [CrossRef]
6. Rumelhart, D.E.; Hinton, G.E.; Williams, R.J. Learning Internal Representations by Error Propagation, Parallel Distributed Processing, Explorations in the Microstructure of Cognition, ed. DE Rumelhart and J. McClelland. Vol. 1. 1986. *Biometrika* **1986**, *71*, 599–607.
7. Xhafa, F. *Machine Learning, Big Data, and IoT for Medical Informatics*; Academic Press: Cambridge, MA, USA, 2021.
8. Zhang, X.; Xing, Y.; Sun, K.; Guo, Y. OmiEmbed: A unified multi-task deep learning framework for multi-omics data. *Cancers* **2021**, *13*, 3047. [CrossRef]
9. Gambhir, R.; Osathapan, A.; Thaler, J. Streamlining latent spaces in machine learning using moment pooling. *Phys. Rev. D* **2024**, *110*, 074020. [CrossRef]
10. Batista, G.E.; Carvalho, A.C.; Monard, M.C. Applying one-sided selection to unbalanced datasets. In *Proceedings of the Mexican International Conference on Artificial Intelligence*; Springer: Berlin/Heidelberg, Germany, 2000; pp. 315–325.
11. Chawla, N.V.; Bowyer, K.W.; Hall, L.O.; Kegelmeyer, W.P. SMOTE: Synthetic minority over-sampling technique. *J. Artif. Intell. Res.* **2002**, *16*, 321–357. [CrossRef]
12. He, H.; Bai, Y.; Garcia, E.A.; Li, S. ADASYN: Adaptive synthetic sampling approach for imbalanced learning. In *Proceedings of the 2008 IEEE International Joint Conference on Neural Networks (IEEE World Congress on Computational Intelligence)*; IEEE: New York, NY, USA, 2008; pp. 1322–1328.
13. Goodfellow, I.; Pouget-Abadie, J.; Mirza, M.; Xu, B.; Warde-Farley, D.; Ozair, S.; Courville, A.; Bengio, Y. Generative adversarial nets. *Adv. Neural Inf. Process. Syst.* **2014**, *27*, 2672–2680.
14. Xu, L.; Skoularidou, M.; Cuesta-Infante, A.; Veeramachaneni, K. Modeling tabular data using conditional gan. *Adv. Neural Inf. Process. Syst.* **2019**, *32*, 7335–7345.
15. Ågren, W.; Sosa, V.Ú. Hierarchical Conditional Tabular GAN for Multi-Tabular Synthetic Data Generation. *arXiv* **2024**, arXiv:2411.07009. [CrossRef]
16. Nußberger, J.; Boesel, F.; Lenz, S.; Binder, H.; Hess, M. Synthetic observations from deep generative models and binary omics data with limited sample size. *Briefings Bioinform.* **2021**, *22*, bbaa226. [CrossRef]
17. Ahmed, K.T.; Sun, J.; Cheng, S.; Yong, J.; Zhang, W. Multi-omics data integration by generative adversarial network. *Bioinformatics* **2022**, *38*, 179–186. [CrossRef]
18. Habibi, O.; Chemmakha, M.; Lazaar, M. Imbalanced tabular data modelization using CTGAN and machine learning to improve IoT Botnet attacks detection. *Eng. Appl. Artif. Intell.* **2023**, *118*, 105669. [CrossRef]
19. Arjovsky, M.; Chintala, S.; Bottou, L. Wasserstein generative adversarial networks. In Proceedings of the International Conference on Machine Learning. PMLR, Sydney, Australia, 6–11 August 2017; pp. 214–223.
20. Kim, J.; Seok, J. ctGAN: Combined transformation of gene expression and survival data with generative adversarial network. *Briefings Bioinform.* **2024**, *25*, bbae325. [CrossRef] [PubMed]
21. Alharbi, F.; Vakanski, A.; Elbashir, M. K.; Mohammed, M. LASSO–MOGAT: A multi-omics graph attention framework for cancer classification. *Acad. Biol.* **2024**, *2*. 1–14.
22. Sidorenko, D.; Pushkov, S.; Sakip, A.; Leung, G.H.D.; Lok, S.W.Y.; Urban, A.; Zagirova, D.; Veviorskiy, A.; Tihonova, N.; Kalashnikov, A.; et al. Precious2GPT: The combination of multiomics pretrained transformer and conditional diffusion for artificial multi-omics multi-species multi-tissue sample generation. *Npj Aging* **2024**, *10*, 37. [CrossRef] [PubMed]
23. Zhang, X.; Zhang, J.; Sun, K.; Yang, X.; Dai, C.; Guo, Y. Integrated multi-omics analysis using variational autoencoders: application to pan-cancer classification. In *Proceedings of the 2019 IEEE International Conference on Bioinformatics and Biomedicine (BIBM)*; IEEE: New York, NY, USA, 2019; pp. 765–769.
24. CiteDrive, I. Deep Dive: Multi-Omics Variational Autoencoding. Available online: https://www.broadinstitute.org/videos/deep-dive-multi-omics-variational-autoencoding-variational-autoencoders-analysis-and (accessed on 1 January 2025).
25. Choi, Y.; Li, R.; Quon, G. siVAE: Interpretable deep generative models for single-cell transcriptomes. *Genome Biol.* **2023**, *24*, 29. [CrossRef]

26. Lee, H.S.; Hong, S.H.; Kim, G.H.; You, H.J.; Lee, E.Y.; Jeong, J.H.; Ahn, J.W.; Kim, J.H. Generative Models Utilizing Padding Can Efficiently Integrate and Generate Multi-Omics Data. *AI* **2024**, *5*, 1614–1632. [CrossRef]
27. Xin, L.; Huang, C.; Li, H.; Huang, S.; Feng, Y.; Kong, Z.; Liu, Z.; Li, S.; Yu, C.; Shen, F.; et al. Artificial intelligence for central dogma-centric multi-omics: challenges and breakthroughs. *arXiv* **2024**, arXiv:2412.12668.
28. Ciriello, G.; Gatza, M.L.; Beck, A.H.; Wilkerson, M.D.; Rhie, S.K.; Pastore, A.; Zhang, H.; McLellan, M.; Yau, C.; Kandoth, C.; et al. Comprehensive molecular portraits of invasive lobular breast cancer. *Cell* **2015**, *163*, 506–519. [CrossRef]
29. Cerami, E.; Gao, J.; Dogrusoz, U.; Gross, B.E.; Sumer, S.O.; Aksoy, B.A.; Jacobsen, A.; Byrne, C.J.; Heuer, M.L.; Larsson, E.; et al. The cBio cancer genomics portal: An open platform for exploring multidimensional cancer genomics data. *Cancer Discov.* **2012**, *2*, 401–404. [CrossRef]
30. Gao, J.; Aksoy, B.A.; Dogrusoz, U.; Dresdner, G.; Gross, B.; Sumer, S.O.; Sun, Y.; Jacobsen, A.; Sinha, R.; Larsson, E.; et al. Integrative analysis of complex cancer genomics and clinical profiles using the cBioPortal. *Sci. Signal.* **2013**, *6*, pl1. [CrossRef]
31. Benjamini, Y.; Hochberg, Y. Controlling the false discovery rate: A practical and powerful approach to multiple testing. *J. R. Stat. Soc. Ser. B (Methodol.)* **1995**, *57*, 289–300. [CrossRef]
32. Householder, A.S. A theory of steady-state activity in nerve-fiber networks: I. Definitions and preliminary lemmas. *Bull. Math. Biophys.* **1941**, *3*, 63–69. [CrossRef]
33. Kingma, D.P.; Ba, J. Adam: A method for stochastic optimization. *arXiv* **2014**, arXiv:1412.6980.
34. Mehrbakhsh, Z.; Hassanzadeh, R.; Behnampour, N.; Tapak, L.; Zarrin, Z.; Khazaei, S.; Dinu, I. Machine learning-based evaluation of prognostic factors for mortality and relapse in patients with acute lymphoblastic leukemia: A comparative simulation study. *Bmc Med. Inform. Decis. Mak.* **2024**, *24*, 261. [CrossRef] [PubMed]
35. Zhou, J.; Lu, X.; Chang, W.; Wan, C.; Lu, X.; Zhang, C.; Cao, S. PLUS: Predicting cancer metastasis potential based on positive and unlabeled learning. *PLoS Comput. Biol.* **2022**, *18*, e1009956. [CrossRef]
36. Wu, A.N.; Stouffs, R.; Biljecki, F. Generative Adversarial Networks in the built environment: A comprehensive review of the application of GANs across data types and scales. *Build. Environ.* **2022**, *223*, 109477. [CrossRef]

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.