SNPs-Analysis VAE (SAVE): A Semi-supervised Learning Approach for SNP-based Bleomycin Toxicity Prediction

Beh Qian Jun[1], Emmanuel Vikran[1], Than Hui Xin[1]

Assistant Prof. Folefac Aminkeng[2], A/P Ngiam Kee Yuan[2], Bharath Shankar[1], Lim Ting Wei[1]

[1] Special Programme in Science, National University of Singapore

[2] Department of Medicine, Yong Loo Lin School of Medicine, National University of Singapore

Acknowledgment **5**
Abstract **6**
1. Introduction **7**
1.1 Past work and Research Gap 8
1.2 Significance 9
2. Background **10**
2.1 Challenges of the Dataset 11
2.2 Introduction to the Machine Learning models 12
2.3 Variational Autoencoder 12
2.4 Bleomycin-induced Toxicity 14
3. Materials & Methods **15**
3.1 Data Preprocessing 15
3.1.1 Data Collection 15
3.1.2 Data Cleaning 15
3.2 Variational Autoencoder (VAE) 18
3.2.1 Model Architecture 18
3.2.2 Regularisation 19
3.2.3 Model Training & Hyperparameter Tuning 22
3.3 Generating & Labelling of Synthetic Data for the Training of the classifer models 23
3.4 Training of Classifier Models on Different Variants of Datasets 25
4. Results **26**
4.1 Significant genetic markers obtained from the Golden HelixTree 26
4.2 Analysis of the VAE Performance 27
4.2.1 Loss Curves of VAE 27
4.2.2 Visualisation of Latent Space 28
4.3 Performance of Models 29
4.3.1 Performance of Models Trained on the Compressed Train Set 29
4.3.2 Performance of Models Trained on the Labelled Synthetic Data 30
4.3.3 Performance of Models Trained on the VAE Feature Selected Original Dataset 31
5. Discussion **33**
5.1 Possible Improvements on Regularisations of VAE 33
5.2 Evaluation of the models' performance 33
5.2.1 Evaluation of Models Trained on the Labelled Synthetic Data 33
5.2.2 Evaluation of Models Trained on the VAE Feature Selected Original Dataset 34
5.2.3 Overall Performance of Models using Synthetic Data 34
5.3 Identification of Top Genetic Variants by the Trained VAE 35
6. Future Work **36**
7. Conclusion **37**
References **38**
Supplementary Material **49**

Acknowledgment

We express our gratitude to Assistant Professor Folefac Aminkeng, our principal investigator, for granting us the chance to collaborate on this project and for his guidance throughout the process. Furthermore, we extend our appreciation to Associate Professor Ngiam Kee Yuan for introducing us to Prof. Aminkeng. Finally, we would like to acknowledge our SPS mentors, Lim Ting Wei and Bharath Shankar, for dedicating their time and effort in providing us with constructive feedback and guidance regarding the project.

Abstract

This report discusses the application of precision medicine using genomic data in predicting susceptibility to drug-induced toxicity. While conventional statistical methods have been replaced by Machine Learning (ML) techniques that can model intricate associations within genetic variants and their impact on the outcome of a disease, overfitting can occur in imbalanced datasets with limited observations, resulting in high accuracy in predicting those not susceptible to the disease and lower accuracy in predicting those who are. To address this issue, generative models such as Variational Autoencoders (VAE) can be utilised to generate synthetic data from low samples and biased datasets with high dimensionality. In this study, we introduce a novel approach called SNPs-Analysis VAE (SAVE) that effectively analyses and generates synthetic data while alleviating overfitting issues. Although there may be a tradeoff in the predictive power of the classifier model when using synthetic data, the ability to generate a significant number of synthetic data points that more accurately reflect the original population and to train the model using compressed data from the latent space can lead to a better-balanced dataset and significantly lower computational costs for model training.

1. Introduction

The rapid development of precision medicine using genomic data has been facilitated by technological advancements, where large amounts of data are easily handled and the costs of DNA sequencing have been heavily reduced (Ho et al., 2019). The differences in how individuals are susceptible to diseases or toxicity to a drug are found to be highly correlated to the genetic differences in a population. Ongoing research is being conducted on how precision medicine using machine learning (ML) models can pave the way for predicting diseases, prognosis, preventive healthcare, and pharmacogenomics based on genetic markers identified from single nucleotide polymorphism (SNP) (Zlobina et al., 2022a; Sharma & Prabha, 2021). Providing medical treatment based on individual genetic characteristics could improve medical efficacy and cost-effectiveness. In diseases such as sickle-cell anaemia or Huntington's disease, the analysis of a particular gene can determine if an individual is affected by the disease (Wu et al., 2023). These diseases are caused by risk alleles[1] from one gene that confer susceptibility to that disease (Wu et al., 2023). However, in complex and polygenic diseases such as diabetes, multiple genes could influence an individual's susceptibility to that disease (Visscher et al., 2012). Parker *et al.* demonstrated the use of ML to predict treatment outcomes of chemotherapy treatment on women with breast cancer using 50 genetic markers identified within the tumours (Dizon et al., 2014). ML has also been applied in predicting diabetes, hypertension, and kidney diseases in patients. In such diseases, not all the involved genes contribute equally to the disease. Certain genes may play a larger role as compared to other genes in polygenic diseases (Wang et al., 2022). A patient's susceptibility to a disease depends on the number of risk alleles he possesses. Furthermore, diseases could also be multifactorial, where non-genomic factors such as

[1] Alleles are different forms of genes, where in one gene, there could be an allele that confers risk or protective effect from a disease.

age and environmental conditions could affect the outcome of a disease. Conventional statistical methods such as polygenic risk scoring, Chi-squared tests, Fisher's Exact test and Regression analysis are used to determine the association between genetic markers and diseases. However, they fail to consider intricate connections among risk alleles in patients, which ML algorithms can recognise and differentiate.

1.1 Past work and Research Gap

ML is predominantly used in two methods for disease prediction using genomic data: One is where statistically relevant genetic markers are identified and fed into a model to predict the outcome as how Sarwar *et. al* (2023) and Kaklamani *et. al* (2006) demonstrated in their studies, and the second is where all the identified genetic markers are fed into the model to predict the outcome, be it statistically relevant or not (Parker 2009; Lim 2023). The former may be focused, targeted and computationally cheaper. However, this does not fully capture the complexity of associations within genes and may overlook important genetic markers. The latter method provides a much more comprehensive view of the complex disease-associated genetic patterns and thus could make accurate predictions. However, this method is computationally expensive and may capture noise instead of the underlying signal in the dataset.

Furthermore, a small, biased dataset with high dimensionality usually reduces the predictive power of an ML model, further discussed in later sections. To address these problems, generative ML models like Variational Autoencoder (VAE) could be implemented.

VAE is a type of neural network architecture that can be used to generate synthetic[2] data from small and large datasets alike. It is an unsupervised[3] learning model that learns the underlying distribution of data and generates new data by sampling from the learned distribution of the latent variables. Zhang *et al.* (2019) noted that VAE has gained considerable popularity in embedding text and image data into the latent space[4]. However, the application of VAE for analysing genomics data is relatively nascent. VAE has often been used to detect associations between genomic sequences and gene expression levels, such as inferring cell-cell communication patterns between a receptor and a ligand based on representative RNA sequences (Jia et al., 2021). However, there has been a limited number of studies that use VAE to generate synthetic SNP data to mitigate the problem of an imbalanced dataset.

1.2 Significance

In this study, we aim to develop a novel method, uniquely named SNP-analysis VAE (SAVE), that can be used to explore associations between SNPs and toxicity with minimum domain knowledge, handle imbalanced datasets with large dimensionality and predict a patient's susceptibility to a particular disease. The dataset that we have tested this tool on is from a small sample of 127 patients with Hodgekin's Lymphoma, in which some have shown toxicity after the administration of Bleomycin, which will be discussed in later sections. However, the number of patients who have experienced Bleomycin-induced toxicity (BIT), defined as cases, is of a very

[2] Data that is artificially generated which simulates the distribution of real-world data.
[3] ML algorithms that take in unlabelled data, where the output variable is not required in the dataset.
[4] A compressed and continuous lower-dimensional space that captures the underlying distribution of the input data.

small proportion as compared to those who have not, defined as controls, leading to imbalanced data. Our method aims to infer meaningful output from an imbalanced dataset through feature selection and semi-supervised learning[5], generating synthetic data to increase the number of observations, providing a more accurate representation of the population. This data is then used to train ML models to predict BIT in individuals. This tool is aimed to be affordable and comprehensive, thus allowing individuals to generate scientific insights from datasets with minimum domain knowledge and without the purchase of any software license. We used both statistically relevant genetic markers selected by Chi-squared tests and genetic markers from the whole dataset to train the VAE to compare which method better captures the relationships present in the dataset. By finding the method that captures this complexity, we can potentially predict for drug-induced toxicity and susceptibility to polygenic diseases, contributing to personalised medicine.

2. Background

Classification is a supervised[6] ML algorithm which predicts the category that the new observation belongs to after being trained on a given dataset (Tapak et al., 2023). The output variable in our dataset is the case/control column, which indicates if the patient has developed BIT.

[5] An ML approach that uses both labelled and unlabelled data to train the model.
[6] A type of ML algorithm which takes in labelled data to train the model.

2.1 Challenges of the Dataset

However, our dataset has numerous issues that make it difficult to determine which model is most ideal for our classification task (Morán-Fernández et al., 2022). Firstly, the dataset is of extremely high dimensionality, with over 700,000 features. The large number of features makes it difficult for the model to find patterns in the data (Najafabadi et al., 2015). Since the model will likely capture noise in the training process, the models are prone to overfitting[7] (López et al., 2022). Furthermore, a dataset of such high dimensionality will take up an extensive amount of computational resources that can be costly and time-consuming (Jia et al., 2022a).

In addition, the dataset only has a small number of 127 observations, which is unrepresentative of reality. The model parameters estimated will be biased, and the model will likely be overfitted. The dataset is also highly imbalanced, where there are many more patients that did not develop BIT compared to those that did, with 96 and 31 observations, respectively. The model will fit more closely to the majority class as there are more data points for the model to learn from and it will likely end up overfitting. Therefore, several classification models were employed to select the model that best overcomes the problems of our dataset.

[7] The phenomenon where the model fits the training data too closely, learning even the irrelevant features and noise of the dataset. The model will then perform extremely well on the training dataset and have a high training accuracy, but be unable to generalise to the new data, leading to a much poorer testing accuracy

2.2 Introduction to the Machine Learning models

The classification models include Logistic Regression, Random Forest classifer, eXtreme Gradient Boosting (XGBoost) classifer, Categorical Boosting (CatBoost) classifer, Light Gradient-Boosting Machine (LightGBM) classifer and an Artificial Neural Network (ANN) model. As mentioned above, our dataset poses many challenges that require us to test out the different models before selecting the most optimal model. Though some of these models do combat overfitting better than others, all models are likely to be overfitted due to the small, imbalanced dataset of high dimensionality. Therefore, to reduce the extent of overfitting, a VAE model was built to potentially solve these issues through feature selection and the generation of synthetic, as discussed in our Materials & Methods.

More details regarding the models are given in the Supplementary Material.

2.3 Variational Autoencoder

Variational Autoencoder (VAE) is an unsupervised[8] deep-learning approach. This is unlike a more common type of neural network (NN), namely the Convolutional Neural Network (CNN), which models through minimising loss of class predictions; VAE learns by data reconstruction[9] (Way & Greene, 2018).

In general, autoencoders are mainly dimensional reduction algorithms which consist of three parts: encoder, latent space, and decoder. Hence, it would be perfect for our datasets, which have

[8] An ML algorithm which takes in unlabelled data to train the model.

[9] A process which data is sampled from the latent space, and is passed through a decoder of VAE.

immensely large dimensions of over 700,000 features. Specifically in VAE, the input is encoded as a distribution over the latent space. In our model, we encode the input as a Normal distribution. Then, a point from the latent space is sampled from the distribution and is subsequently decoded in the decoder layer. Finally, the reconstruction error[10] is computed and backpropagated[11] through the network.

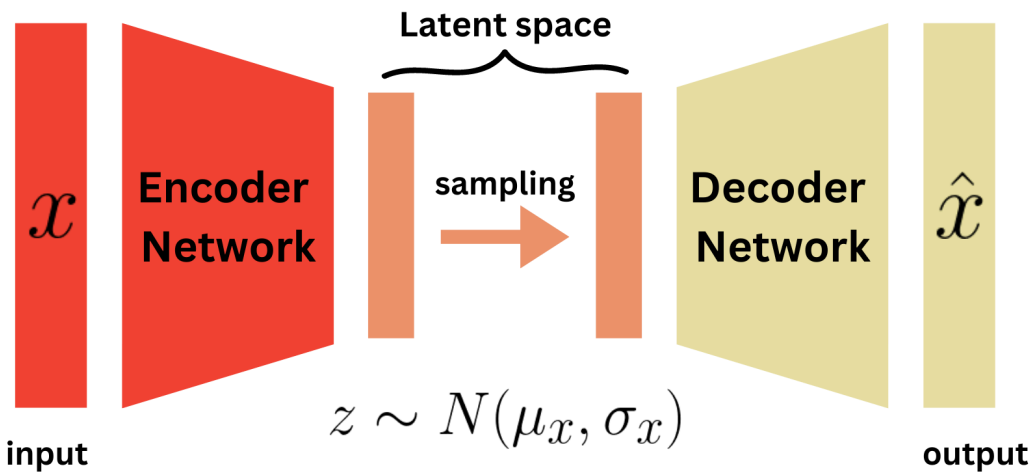


Figure 1: A simple visual representation of the conventional VAE, in which the blocks' sizes correspond to the dimensionality of the data at each stage. The input data, which can have a high dimensionality, is significantly reduced in the latent space, also known as the "bottleneck".

The loss function of VAE comprises two parts: the generative loss and the latent loss. The generative loss computes the difference between input and output. The latent loss computes the

[10] A measure of how well the decoder network can reconstruct the original input data from the learnt latent representation.
[11] A technique for NN to learn how to adjust the weights and biases such that it reduces the overall error.

differences of the latent vector with the mean and variance of the standard normal distribution $X \sim N(0,\ 1)$. More specifically, Kullback-Leibler divergence (KL divergence)[12] is used as the latent loss function.

See Supplementary Materials for more information regarding models.

2.4 Bleomycin-induced Toxicity

Bleomycin is a common drug in chemotherapy and is a major antimitotic agent that is commonly administered for Hodgekin's Lymphoma, targetting actively dividing cells (Jennane et al., 2022). Bleomycin acts by forming a complex with oxygen and iron to produce reactive oxygen species such as superoxides and hydroxyl radicals (Borzone et al., 2001). These can alter the structures of biologically important molecules including DNA, thus damaging the tumour cells(Jennane et al., 2022). About 10% of the patients administered with bleomycin develop toxicity and about 3% of them are fatally affected (Gundogan et al., 2021). Multiple genetic factors have been described to cause BIT. One of them is the Fas/ FasL pathway which is involved in apoptosis, a process for cell death (Wallach-Dayan et al., 2006). The lungs are one of the organs that express the Fas proteins, which are part of this pathway and are responsible for Fas-dependent apoptosis. When bleomycin is administered, there is the activation of the Fas/FasL pathway in the lungs, leading to fibrosis in the lungs (Wallach-Dayan et al., 2006). Other genetic factors include those that are involved in the release of Interleukin-6 (IL-6) which leads to inflammation in the lungs

[12] A measure of the difference between the distribution of the input data and the distribution of the data sampled from the latent space

and the recruitment of macrophages, leading to fibrosis. This includes the Arid5a gene that was demonstrated to upregulate IL-6 production (Masuda et al., 2013). Given that multiple genetic factors could lead to BIT, it is imperative to use ML methods that can predict BIT in patients before the administration of Bleomycin. Furthermore, newly discovered gene markers associated to BIT can be added to this tool to improve its accuracy.

3. Materials & Methods

3.1 Data Preprocessing

3.1.1 Data Collection

The genetic variants of 700,078 SNPs of 127 patients from the National University Cancer Institute, Singapore were genotyped with Illumina Global Screening arrays[13]. Other details obtained from patients include age, amount of bleomycin received and many variables totalling to about 50 factors.

3.1.2 Data Cleaning

The obtained data was cleaned by removing genetic markers that have more than 5% missing data. It was further subjected to the removal of genetic markers that displayed a minor allele

[13] A chip on which the DNA is loaded to read sequences from specific loci (positions) of the DNA.

frequency[14] of less than 1% (Wang et al., 2022). This is to correct for variations that are too insignificant to be input variables, such as loci in which all patients possess the same genotype.

To address the issue of missing values in the dataset, we employed two different techniques. For categorical variables, we assigned a new category to represent the missing values, while for numerical variables, we replaced the missing values with the average value of the corresponding feFigure 1: A simple visual representation of the conventional VAE, in which the blocks' sizes correspond to the dimensionality of the data at each stage. The input data, which can have a high dimensionality, is significantly reduced in the latent space, also known as the "bottleneck". ature, as stated by Zhang *et al.* (2019).

Chi-squared test was performed using Golden HelixTree and corrected for multiple testing corrections using Bonferroni adjustments to isolate the top ten most significant genetic markers associated to BIT. The top ten genetic markers with the lowest p-values were chosen to fit into the model. The non-genomic factors (Supplementary Table 1) were also chosen to fit into the model based on the Chi-squared test (*p-value* < 0.05**)** and a preliminary logistic regression model.

[14] The frequency at which the second most common allele occurs in a given population.

The resulting data was one-hot encoded to binary vectors for categorical variables. The numerical data were standardised to have a mean of 0 and a standard deviation of 1 to bring all numerical features on the same scale. 90% of the data was allocated to be the train set, whereas the remaining data was split equally between test and validation sets. The models are fitted on the training data to learn the aspects of the data, while the preliminary model's performance is scored on the validation set. Furthermore, the validation set is also used for hyperparameter tuning, and the final, tuned model is scored on the test data.

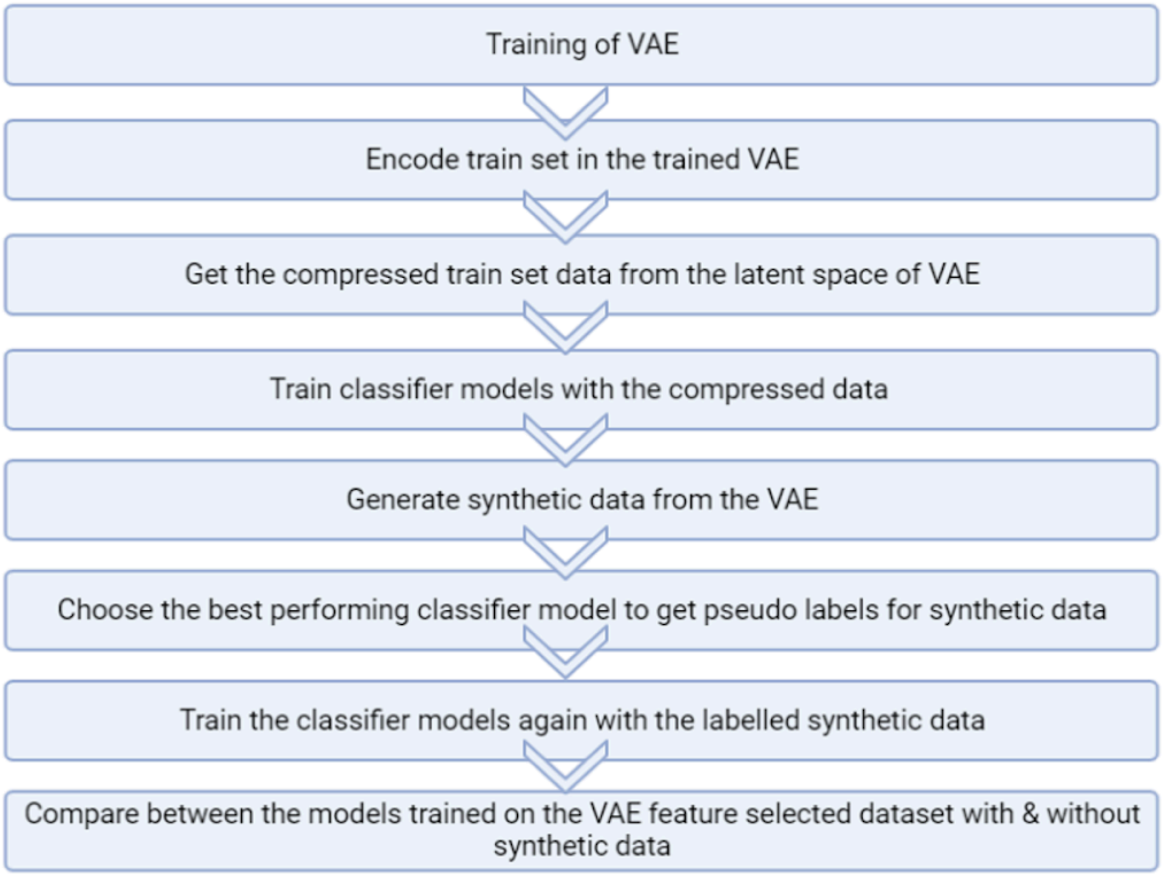


Figure 2: An overview of SAVE

3.2 Variational Autoencoder (VAE)

3.2.1 Model Architecture

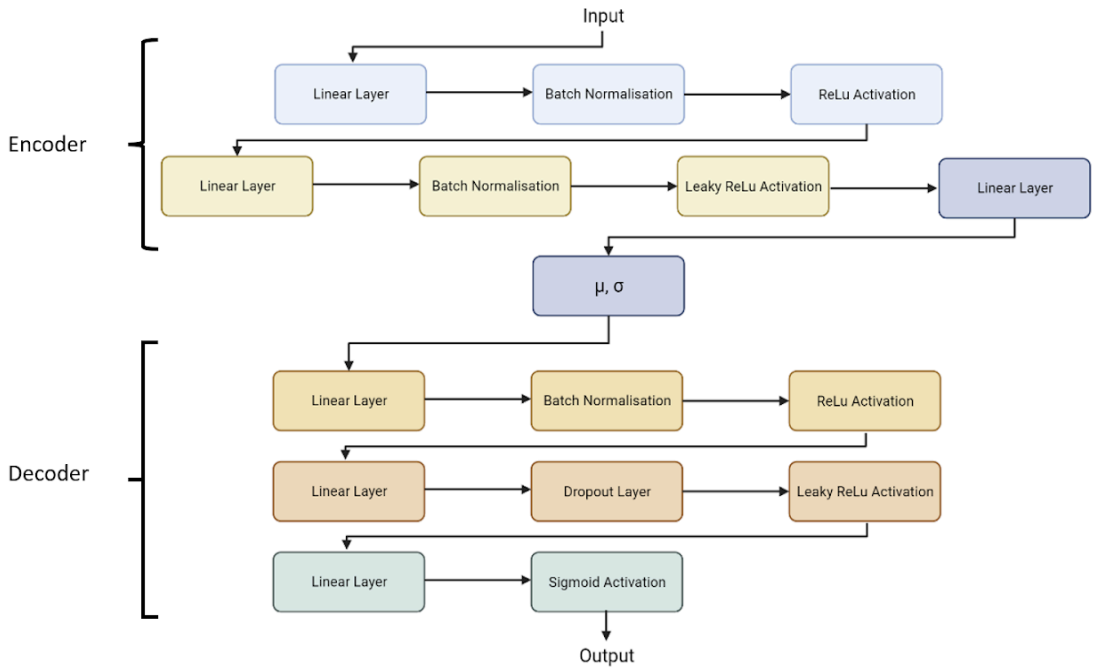


Figure 3: The model's architecture of our proposed VAE.

Our VAE model comprised 3 main layers: the encoder, the latent space, and the decoder, which is consistent with the conventional VAE model. The encoder layer incorporated Linear, LeakyRelu, Relu, and Sigmoid activation functions, as depicted in Figure 3. The selection of different activation functions was mostly empirical.

3.2.2 Regularisation

Batch Normalisation & Variational Dropout

We employed a regularisation method to improve the performance of the VAE model. Specifically, we modified certain layers in the VAE by implementing two techniques: Batch Normalisation and Variational Dropout.

Batch Normalisation is a technique used to normalise the activations of a mini-batch[15] of data by computing the mean and variance, and then applying a linear transformation. To achieve this, we utilised the ``nn.BatchNorm1d`` function from the PyTorch libraries. We opted for Batch Normalisation because it has been shown to improve the training of deep neural networks by reducing the internal covariate shift, which occurs when the distribution of input data changes during the training of the model (Ioffe & Szegedy, 2015).

In addition, we employed Variational Dropout, which is a conventional dropout technique with a user-defined probability to alleviate overfitting in deep neural networks. We set the dropout rate at 0.5, signifying that the layer has a 50% chance of dropping random neurons while training. This technique was adopted based on the work done by Kingma *et al*. (2015).

[15] A subset of the training set which contains a fixed number of samples, e.g. if the batch size is 32, that means the training set is split into 32 mini batches.

Cyclical annealing

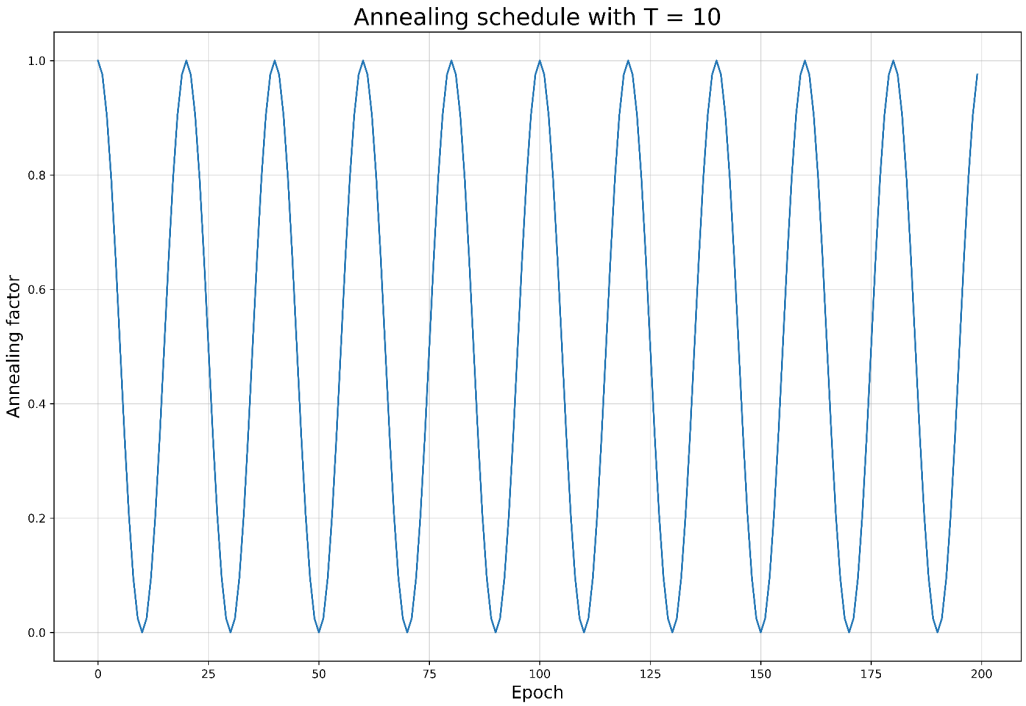


Figure 4: An illustration of the evolution of the annealing factor over epochs, with a period of T = 10.

To address the notorious KL-vanishing problem[16] in the training of a conventional VAE model, we implemented a cyclical annealing schedule[17]. To achieve this, we defined an ``anneal_function`` that took the current epoch number and annealing period T as inputs and returned a value between 0 and 1. The function used a cosine function to create a smooth transition from 0 to 1, with the returned value representing the annealing factor. We used the

[16] A phenomenon in which the KL divergence approaches zero during the training of VAE, leading to poor data reconstruction ability as the prior and posterior distributions have almost no difference.
[17] Cyclical annealing schedule in VAE involves gradually increasing the weight of the KL divergence term in the loss function during training, then reducing it, in a cyclical fashion, to encourage exploration of the latent space while preventing over-regularisation.

annealing factor to interpolate between β_{min} and β_{max} to obtain the value of β at each epoch, in which the β_{min} and β_{max} are set to be 0.1 and 1 respectively:

$$\beta = \beta_{min} + (\beta_{max} - \beta_{min}) \times annealing\ factor \quad (1)$$

To identify the best value of β for the model, we evaluated it on a validation set for each value of β. The value of β that resulted in the lowest validation loss was selected as the best β, which β is an additional term attached to the KL divergence term. This approach allowed for progressive learning of a more meaningful latent space and mitigated the KL-vanishing problem during training, as suggested by Fu *et al.* (2019).

Elastic Net

To further mitigate overfitting, we implemented elastic net regularisation to the model. Elastic net regularisation improves the model by adding both L1 and L2 regularisation terms to the loss function during training.

As mentioned earlier, L1 regularisation introduces a penalty term to the loss function. This helps the model to have sparse weights, meaning that many of the weights were set to zero, thereby reducing the complexity of the model.

L2 regularisation was also added to the loss function by incorporating a penalty term that is proportional to the square of the weights of the model. This encourages the model to have smaller weights, which helps to prevent overfitting by reducing the model's complexity and making it more generalisable to new data.

3.2.3 Model Training & Hyperparameter Tuning

To identify the optimal hyperparameters for training the model, we employed Bayesian optimisation[18] as our tuning strategy. The search space for the hyperparameters was defined as a set of intervals, within which the hyperparameters were sampled.

[18] A method for finding the best input to a function by selecting new inputs to evaluate based on previous results using a probabilistic model.

$$Search\ space = \begin{cases} Learning\_rate \in [1e-5, 1e-3] \\ Batch\_size \in [8, 123] \\ \lambda_1 \in [1e-6, 0.01] \\ \lambda_2 \in [1e-6, 0.01] \end{cases}$$

The model was then trained and evaluated using the sampled hyperparameters. The validation loss was used to determine the performance of the model for each set of hyperparameters, and the set of hyperparameters that resulted in the lowest validation loss was selected as the optimal hyperparameters. This approach allows us to efficiently explore the hyperparameter space and identify the best hyperparameters for training the model.

3.3 Generating & Labelling of Synthetic Data for the Training of the classifer models

After the model was trained, 1000 samples were randomly drawn from the learnt latent space, constituting synthetic data of dimensions 1000 x 10. The trained VAE was used to obtain a compressed train set from the latent space by passing the original train set through the VAE's encoder. The original dimensions of the train set were 114 x 1406281, whereas the compressed train set was reduced to 114 x 10. There is a significant reduction in the size of more than 100000 times. However, the generated data points lack the output variable and require subsequent labelling by using a classifer model to predict pseudo labels for the synthetic data.

We fitted the different classifier models to the train set — Logistic Regression, Random Forest, CatBoost, XGBoost, LightGBM and ANN. The same general procedure was followed for all the different models. A preliminary model was first built to evaluate the initial performance and identify potential issues, such as overfitting and underfitting. The models then underwent

hyperparameter tuning via Random Search which looks for the hyperparameter values that maximise the validation score for the model. This searching technique was chosen to tune all models as it is less computationally expensive compared to other techniques like Grid Search, which permutates through the search space. The hyperparameters tuned for each model are given in the Supplementary Material.

The LightGBM acquired the highest F1-score in predicting case observations, a metric we prioritised as a major issue we faced is being unable to predict cases correctly. The F1-score metric was chosen to determine the performance of the models as it takes into account the quality of positive predictions made, and the number of positive observations the model was able to predict. These two circumstances are crucial as they ensure we do not predict people who are controls as cases, and predict those who actually have the disease as cases. Therefore, as LightGBM performed the best amongst the other models, it was used to predict the pseudo labels for the synthetic data (Table 1).

To investigate the importance of input features in the model's output, the synthetic data was decoded into the output space. The baseline output was also obtained by decoding a tensor of zeros with the same model. Feature importance scores were calculated by subtracting the baseline output from the synthetic data samples. The mean scores were then computed by averaging the feature importance scores across the 1000 samples. The columns were then ranked in non-increasing order according to their mean importance scores.

The unique features were extracted from the top 200 columns with the highest feature scores, and these features were used to generate new synthetic data using the same approach as before.

With the synthetic data generated by the VAE, we proceeded to train the LightGBM model on the newly acquired data, where the synthetic data made up the train set, while the original data was split to form the validation and test sets, with 1000, 64 and 63 observations respectively. The original data must be used for scoring the performance of the models as it is a way to determine if the synthetic data generated managed to capture the underlying patterns of the data well, hence leading to a well-performing model. The models' hyperparameters were then tuned using Random Search.

3.4 Training of Classifier Models on Different Variants of Datasets

The classifier models were trained on two additional variants of the datasets, where the original data is feature selected using the Golden HelixTree software, and where the original data is feature selected using VAE. Both models used the same train, validation and test sets with 114, 7, and 6 observations respectively.

Therefore, we compared the performance of the classifier models trained on three different variants of the datasets: (1) Feature selection using Golden HelixTree software, (2) Feature selection using VAE on the original dataset, (3) Feature selection using VAE on the synthetic dataset. These comparisons help determine the effectiveness of our pipeline, which will be later discussed in the next section.

4. Results

4.1 Significant genetic markers obtained from the Golden HelixTree

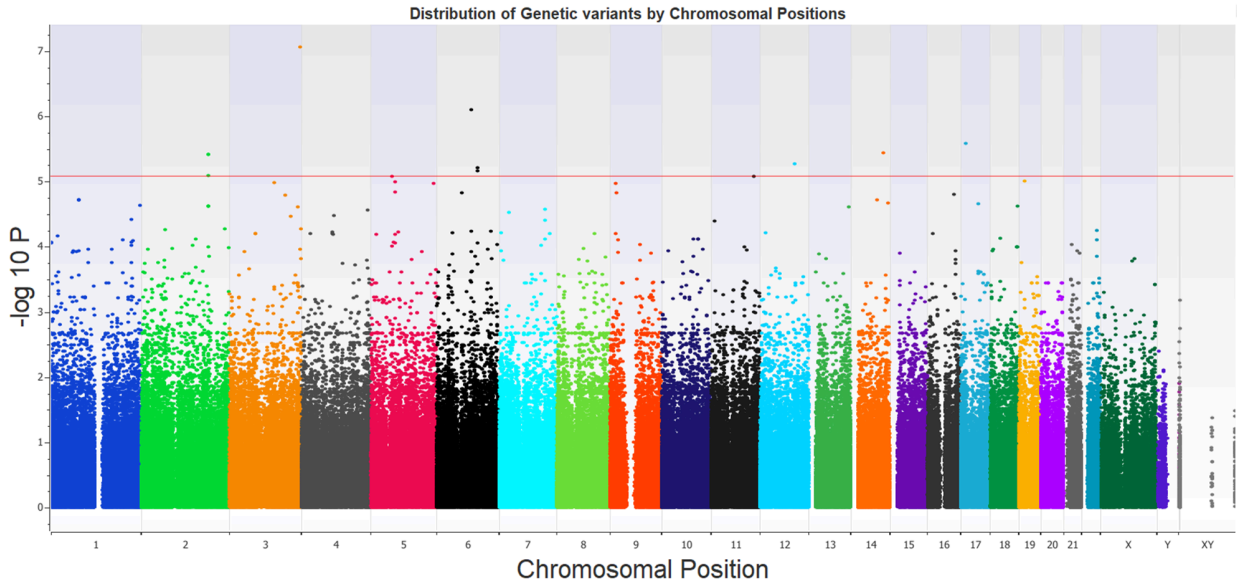


Figure 5: Manhattan plot of population, depicting the distribution of how each genetic variant contributes to the difference between cases and controls. The red line represents the threshold we have set to choose the top 10 genetic variants as they have shown the most significance.

Using Golden HelixTree, the top ten most significant genetic markers were selected based on the p-value of the Chi-squared test. All of these genetic markers have a p-value of less than 8.2 x 10^{-6}. The top ten were chosen empirically. This number varies across various studies from 5 to more than 50 genetic markers (Rocca, 2021; Parker 2009). We chose ten to only demonstrate the use of statistical methods to obtain relevant genetic markers.

4.2 Analysis of the VAE Performance

4.2.1 Loss Curves of VAE

Figure 6: A plot of training and validation loss over 60 epochs.

The training and validation losses decreased progressively as the training went on, hence it is evident that the model is not underfitting. Towards the end of the training process, the validation loss exceeded the training loss, providing evidence that the model's performance generalises well to new data. Nevertheless, it is noteworthy that the disparity between the training and validation losses noticeably grew towards the end of the training period. This phenomenon raises concerns regarding overfitting, which may occur if the training process persists, and the validation loss begins to rise.

4.2.2 Visualisation of Latent Space

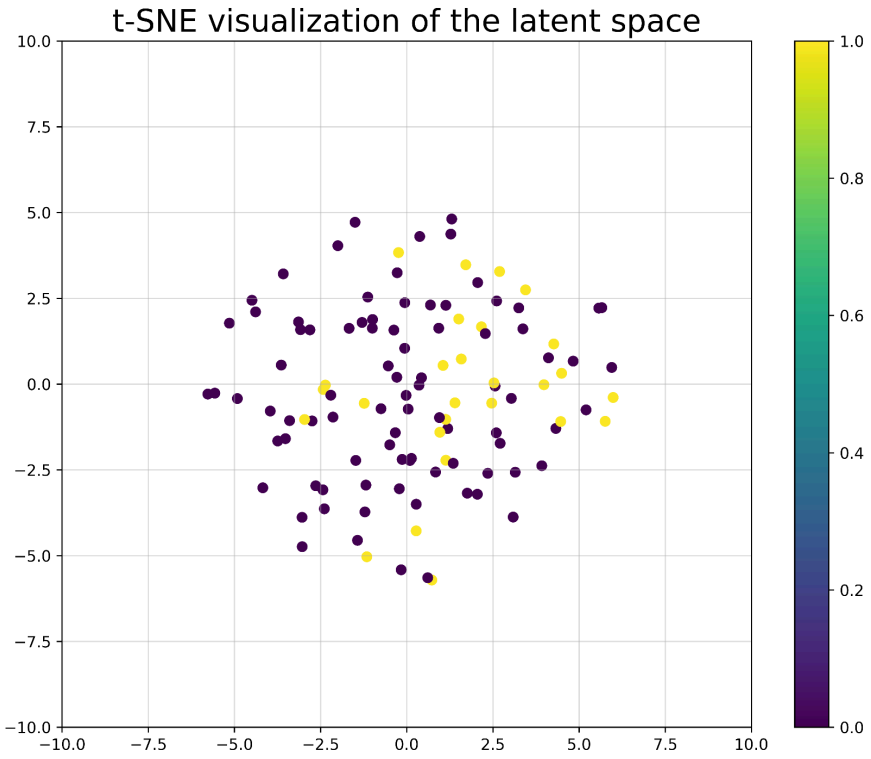


Figure 7: A t-SNE visualisation of the model's latent space trained with the hyperparameters = {Learning rate: 0.000612; Batch size: 78}. The yellow data points indicate the case, while the blue represents the control.

We employed t-distributed neighbour embedding (t-SNE)[19] to visualise the distribution of data points in the model's latent space. The t-SNE method was selected for its effectiveness in mapping high-dimensional data onto a low-dimensional manifold (Maaten & Hinton, 2008). Ideally, the data points visualised in the latent space should be clustered closely, but with clear

[19] A dimensionality reduction technique that maps high-dimensional data to a low-dimensional space while preserving the structure of the data.

distinctions and minimal overlapping between different classes. Our visualisation revealed that both case and control classes were closely clustered, which is desirable as the ideal latent space should be continuous, and the model would generate unrealistic input if they were not.

However, further examination indicated that while cases were predominantly located on the right side of the cluster, they were not very distinctly separated from the control group. This suggests that the model may not be able to differentiate between case and control with complete accuracy in the latent space representation.

4.3 Performance of Models

4.3.1 Performance of Models Trained on the Compressed Train Set

Table 1: Performance of models trained on the compressed train set.

Model	F1-score for Control	F1-score for Case	Accuracy
LightGBM	0.75	0.67	0.71
CatBoost	0.73	0.00	0.57
XGBoost	0.75	0.76	0.71
ANN	0.73	0.00	0.57
Logistic Regression	0.73	0.00	0.57
Random Forest	0.73	0.00	0.57

LightGBM was chosen as the model to get pseudo labels for synthetic data for its relatively high F1-scores and accuracy. Though XGBoost had similar F1-scores and accuracy as LightGBM, LightGBM was still chosen as the best-performing model as it computes faster than XGBoost.

4.3.2 Performance of Models Trained on the Labelled Synthetic Data

Table 2. F1-scores and Accuracies for Models Trained Using Genetic Markers Selected from VAE with the Generation of Synthetic Data.

Models	F1-score for Control	F1-score for Case	Accuracy
LightGBM	0.82	0.48	0.73
CatBoost	0.84	0.11	0.73
XGBoost	0.85	0.11	0.75
ANN	0.90	0.42	0.83
Logistic Regression	0.73	0.00	0.57
Random Forest	0.85	0.35	0.76

After acquiring the pseudo labels for the synthetic data, the classifier models were then trained on the labelled synthetic data to predict for patients with BIT. The top two performing models are the LightGBM and ANN classifiers.

4.3.3 Performance of Models Trained on the VAE Feature Selected Original Dataset

Table 3. F1-scores and Accuracies for Models Trained Using Genetic Markers Selected from VAE with the Original Dataset.

Models	F1-score for Control	F1-score for Case	Accuracy
LightGBM	0.73	0.00	0.57
CatBoost	0.73	0.00	0.57
XGBoost	0.57	0.57	0.57
ANN	0.73	0.00	0.57
Logistic Regression	0.67	0.40	0.57
Random Forest	0.80	0.50	0.71

Most models appeared to have extremely poor F1-scores for cases, demonstrating a large difference when compared to the scores of the models trained on the labelled synthetic data (Table 2).

Table 4: F1-scores and Accuracies for Models Trained Using the Top Ten Most Significant Genetic Markers Selected by Golden HelixTree With the Original Dataset.

Model	F1-score for Control	F1-score for Case	Accuracy
LightGBM	0.86	0.86	0.86
CatBoost	0.80	0.50	0.71
XGBoost	0.86	0.86	0.86
ANN	0.73	0.00	0.57
Logistic Regression	0.80	0.50	0.71
Random Forest	0.80	0.50	0.71

The performance of the models when trained on the dataset feature selected by the Golden HelixTree software appear to fit most closely to the original dataset when compared to the other dataset variants.

However, it must be acknowledged that these are not parallel comparisons as for the models that utilise synthetic data, they are validated and tested on the all the observations of the original dataset, while the others are only validated and tested on a subset of the original dataset. Hence, as more samples are tested for the models using synthetic data, it may also penalise their results to a larger extent.

5. Discussion

5.1 Possible Improvements on Regularisations of VAE

We incorporated dropout regularisation as a technique to improve the performance of our VAE model in the pipeline. Specifically, we used a dropout rate of $p = 0.5$ in the variational dropout. Previous research suggests that a dropout rate of 0.5 is often effective for various machine-learning tasks (Srivastava *et al.*, 2014). However, exploring different values of p may lead to further improvements. Unfortunately, due to limitations in computational resources and training duration, we were unable to conduct such experiments. Future studies may benefit from investigating the impact of varying dropout rates on VAE performance.

5.2 Evaluation of the models' performance

5.2.1 Evaluation of Models Trained on the Labelled Synthetic Data

The trend of having poorer F1-scores for predicting case compared to control indicates that while the model can predict patients without BIT relatively well, it still faces some difficulty in predicting patients with toxicity. This is not a surprising result as due to how imbalanced the original dataset is, the VAE model which generated the synthetic data is likely still overfitted to the majority class, and the synthetic case observations may not fully capture the patterns in the given dataset. Hence, when the classifier models were fitted with the synthetic data, the models likely did not train sufficiently on the case observations, resulting in poorer F1-scores for cases.

5.2.2 Evaluation of Models Trained on the VAE Feature Selected Original Dataset

The result presented in Table 3 likely signifies that despite the synthetic data not capturing the underlying patterns of the original dataset well, it still greatly ameliorated the model performance through the sheer number of data points, which also balanced out the case and control classes.

5.2.3 Overall Performance of Models using Synthetic Data

As mentioned above, the F1-score for predicting patients with BIT using synthetic data is not ideal due to the original dataset being imbalanced and biased, which alludes to it being a poor representation of the real-world population.

However, synthetic data generates more representative and unbiased data, which mimics the real-world better. (Draghi *et al.*, 2021). Therefore, though our models did not perform well on the biased test set that we had, they may exhibit better performance when tested on less biased data. Therefore, to precisely determine the predictive power of our models, we would have to test a more representative sample of the population which is currently difficult to obtain due to this data collection being in its early stages. This can be built on for future work.

In addition, our research demonstrates that the proposed method of using VAE to generate synthetic data for the training of the models to overcome the problem of a small, imbalanced dataset with high dimensionality may be limited. The models trained without synthetic data appear to have higher F1-scores compared to those using synthetic data, similar to what

Muñoz-Cancino *et al.* (2009) found. Nevertheless, an interesting find was made; despite the synthetic data not being representative of the original dataset, by purely increasing the dataset size and balancing the case and control classes, the performance of the models can be elevated.

5.3 Identification of Top Genetic Variants by the Trained VAE

The trained VAE selected the top 200 most weighted genetic variants (Rs2969182 (SNP) - Genes and Regulation - Homo_sapiens, n.d.; Rs4681292 (SNP) - Genes and Regulation - Homo_sapiens, n.d.; Chromosome 3: 78,470,323-78,470,423 - Region in Detail - Homo_sapiens, n.d.). Among these, rs2969182[20] on chromosome 17 was found in the shisa family member 6 gene, involved in synaptic transmission and neurotransmitter receptor trapping (Shisa6 Shisa Family Member 6 [Mus Musculus (House Mouse)] - Gene, n.d.). Another variant, rs4681292 on chromosome 3, was an intronic variant in a novel transcript highly correlated with the ST13 gene, which is highly regulated in the brain anterior region (Rs4681292 (SNP) - Genes and Regulation - Homo_sapiens, n.d.). Additionally, the intergenic variant rs426615 was important in predicting toxicity despite not directly affecting gene expression, as intergenic variants can still impact gene expression and regulatory processes (Jaura et al., 2022; Chromosome 3: 78,470,323-78,470,423 - Region in Detail - Homo_sapiens, n.d.). Notably, the heterozygous genotype for all three genetic markers has a greater impact on the predictability of BIT than homozygous genotypes (unpublished data). Identifying these genes and variants may assist in predicting toxicity in patients.

[20] To identify the SNPs in the genome, scientists generally tag the discovered SNPs with a reference SNP (Rsid) in front of each genetic variant.

6. Future Work

As previously stated, if a larger, more representative dataset is acquired, we can then verify if the models fitted with synthetic data truly captured a better representation of the real-world.

Moreover, if the chromosomal positions and the ordering can be acquired, we would be able to capture the relationships between the chromosomes more effectively by using a CNN model, which can extract meaningful aspects from such data (J. X. Wu, 2017).

The models can also be better-refined with improved hyperparameter tuning. Additionally, transfer learning can be implemented, where knowledge gained from training a model on one dataset can be applied to classifying classes in a different dataset. This may then make our pipeline more suited for small, imabalanced datasets that are omnipresent in healthcare.

7. Conclusion

To address the challenge of analysing and generating synthetic data from a low sample and biased dataset with high dimensionality, we introduced an approach called SNPs-Analysis VAE (SAVE). Our approach uses a semi-supervised method that involves training a VAE model, training classifier models with the compressed train set data, generating synthetic data using the trained VAE, obtaining pseudo labels for unlabelled synthetic data with the best-performing classifier models, decoding the labelled synthetic data using the trained VAE, and finally training the classifier model using the decoded, labelled synthetic data. We evaluated the performance of SAVE using different variants of datasets, including the feature selected dataset using the software, the VAE feature selected original dataset, and the VAE feature selected synthetic dataset. The results show that our approach effectively mitigates the overfitting issue caused by imbalanced data while producing synthetic data with reasonable accuracy.

References

Alshari, H., Saleh, A. Y., & Odabas, A. (2021, April 28). *Comparison of gradient boosting*

Babyak, M. A. (2004). What you see may not be what you get: A brief, nontechnical introduction to overfitting in regression-type models. Psychosomatic Medicine, 66(3), 411–421. https://doi.org/10.1097/00006842-200405000-00021

Borzone, G., Moreno, R., Urrea, R., Meneses, M., Oyarzún, M., & Lisboa, C. (2001). Bleomycin-Induced chronic lung damage does not resemble human idiopathic pulmonary fibrosis. American Journal of Respiratory and Critical Care Medicine, 163(7), 1648–1653. https://doi.org/10.1164/ajrccm.163.7.2006132

Cetin, I., Stephens, M., Camara, O., & González Ballester, M. A. (2023). Attri-VAE: Attribute-based interpretable representations of medical images with variational autoencoders. *Computerized Medical Imaging and Graphics*, *104*, 102158. https://doi.org/10.1016/j.compmedimag.2022.102158

Chang, A. C. (2020a). Machine and deep learning. In Intelligence-Based Medicine (pp. 67–140). Elsevier. http://dx.doi.org/10.1016/b978-0-12-823337-5.00005-6

Chromosome 3: 78,470,323-78,470,423 - Region in detail - Homo_sapiens. (n.d.). Ensembl Genome Browser 109. Retrieved April 3, 2023, from https://asia.ensembl.org/Homo_sapiens/Location/View?db=core;r=3:78470323-78470423;source=dbSNP;v=rs4266157;vdb=variation;vf=92465874;time=1680482612

Decision Tree algorithms for CPU performance. Unknown. https://www.researchgate.net/publication/351133481_Comparison_of_Gradient_Boosting_Decision_Tree_Algorithms_for_CPU_Performance#pf3

Dietterich, et al., T. (2006). *Semi-Supervised learning*. The MIT Press. http://dx.doi.org/10.7551/mitpress/9780262033589.001.0001

Dizon, D., Reynolds, K., Sarangi, S., & Bardia, A. (2014). Precision medicine and personalized breast cancer: Combination pertuzumab therapy. *Pharmacogenomics and Personalized Medicine*, 95. https://doi.org/10.2147/pgpm.s37100

Dreiseitl, S. (2002). Logistic regression and artificial neural network classification models: A methodology review. Journal of Biomedical Informatics, 35(5–6), 352–359. https://doi.org/10.1016/S1532-0464(03)00034-0

Fu, H., Li, C., Liu, X., Gao, J., Celikyilmaz, A., & Carin, L. (2019, March 25). *Cyclical annealing schedule: A simple approach to mitigating KL vanishing*. arXiv.Org. https://arxiv.org/abs/1903.10145

Gundogan, B. D., Taskinlar, S., Arikoglu, T., Balci, Y., & Citak, E. C. (2021). Bleomycin-induced pneumonitis in a child treated with nintedanib: Report of the first case in a childhood. *Journal of Pediatric Hematology/Oncology*, *44*(2), e500–e502.

https://doi.org/10.1097/mph.0000000000002266

Ho, D. S. W., Schierding, W., Wake, M., Saffery, R., & O'Sullivan, J. (2019). Machine learning SNP based prediction for precision medicine. *Frontiers in Genetics*, *10*. https://doi.org/10.3389/fgene.2019.00267

How important is data quality? Best classifiers vs best features. (n.d.). *Neurocomputing*, *470*, 365–375. https://doi.org/10.1016/j.neucom.2021.05.107

Hybrid artificial intelligent systems. (2022). Springer International Publishing. https://arxiv.org/pdf/2301.01212v1.pdf

Ioffe, S., & Szegedy, C. (2015, February 11). *Batch normalization: Accelerating deep network training by reducing internal covariate shift*. arXiv.Org. https://arxiv.org/abs/1502.03167

Jaura, R., Yeh, S.-Y., Montanera, K. N., Ialongo, A., Anwar, Z., Lu, Y., Puwakdandawa, K., & Rhee, H. S. (2022). Extended intergenic DNA contributes to neuron-specific expression of neighboring genes in the mammalian nervous system. *Nature Communications*, *13*(1), 2733. https://doi.org/10.1038/s41467-022-30192-z

Jennane, S., Ababou, M., El Haddad, M., Ait Sahel, O., Mahtat, E. M., El Maaroufi, H., Doudouh, A., & Doghmi, K. (2022a). Bleomycin-Induced lung toxicity in hodgkin's lymphoma: Risk factors in the positron emission tomography era. *Cureus*.

https://doi.org/10.7759/cureus.23993

Jia, Sun, Lian, & Hou. (2022a). Feature dimensionality reduction: A review. *Complex & Intelligent Systems*, *8*(3), 2663–2693. https://doi.org/10.1007/s40747-021-00637-x

Kingma, D. P., Salimans, T., & Welling, M. (2015, June 8). *Variational dropout and the local reparameterization trick*. arXiv.Org. https://arxiv.org/abs/1506.02557

Lee, H.-L. (n.d.). *Figure 8. Principle of a random forest (RF). Bagging is the process of...* ResearchGate. Retrieved March 29, 2023, from https://www.researchgate.net/figure/Principle-of-a-random-forest-RF-Bagging-is-the-process-of-data-sampling-from-a_fig6_351175730

Logistic regression and artificial neural network classification models: A methodology review. (n.d.-a). *Journal of Biomedical Informatics*, *35*(5–6), 352–359. https://doi.org/10.1016/S1532-0464(03)00034-0

López, M., López, M., & Crossa. (2022, January 1). *Overfitting, model tuning, and evaluation of prediction performance*. Springer International Publishing. https://link.springer.com/chapter/10.1007/978-3-030-89010-0_4

Lim, A. J. W., Tyniana, C. T., Lim, L. J., Tan, J. W. L., Koh, E. T., Ang, A. E. L., Chan, G. Y. L., Chan, M. T.-L., Chia, F. L.-A., Chng, H. H., Chua, C. G., Howe, H. S., Koh, L. W., Kong, K. O.,

Law, W. G., Lee, S. S. M., Lian, T. Y., Lim, X. R., Loh, J. M. E., … Lee, C. G. (2023). Robust SNP-based prediction of rheumatoid arthritis through machine-learning-optimized polygenic risk score. *Journal of Translational Medicine*, *21*(1). https://doi.org/10.1186/s12967-023-03939-5

Maaten, L. van der, & Hinton, G. (2008). Visualizing Data using t-SNE. *Journal of Machine Learning Research*, *9*(86), 2579–2605.

Machine learning techniques. (n.d.). ScienceDirect. Retrieved April 3, 2023, from https://www.sciencedirect.com/science/article/pii/B9780128213797000035

Mangoni, A. A., & Jackson, S. H. D. (2003). Age-related changes in pharmacokinetics and pharmacodynamics: Basic principles and practical applications. *British Journal of Clinical Pharmacology*, *57*(1), 6–14. https://doi.org/10.1046/j.1365-2125.2003.02007.x

Masuda, K., Ripley, B., Nishimura, R., Mino, T., Takeuchi, O., Shioi, G., Kiyonari, H., & Kishimoto, T. (2013). Arid5a controls IL-6 mRNA stability, which contributes to elevation of IL-6 level in vivo. *Proceedings of the National Academy of Sciences*, *110*(23), 9409–9414. https://doi.org/10.1073/pnas.1307419110

Mijwil, M. M. (2018, January 27). *Artificial neural networks advantages and disadvantages*. Unknown. https://www.researchgate.net/publication/323665827_Artificial_Neural_Networks_Advantages_

and_Disadvantages

Morán-Fernández, L., Bólon-Canedo, V., & Alonso-Betanzos, A. (2022). How important is data quality? Best classifiers vs best features. *Neurocomputing*, *470*, 365–375. https://doi.org/10.1016/j.neucom.2021.05.107

Muñoz-Cancino, Bravo, Ríos, & Graña. (2022, January 1). Assessment of Creditworthiness Models Privacy-Preserving Training with Synthetic Data. Springer International Publishing. https://link.springer.com/chapter/10.1007/978-3-031-15471-3_32

Najafabadi, Villanustre, Khoshgoftaar, Seliya, Wald, & Muharemagic. (2015). Deep learning applications and challenges in big data analytics. *Journal of Big Data*, *2*(1), 1–21. https://doi.org/10.1186/s40537-014-0007-7

Olivier Chapelle, Bernhard Schölkopf and Alexander Zien. (2006). *Semi-Supervised learning*. The MIT Press. https://direct.mit.edu/books/book/3824/Semi-Supervised-Learning

Rocca, J. (2021, March 21). Understanding variational autoencoders (vaes). *Towards Data Science*.https://towardsdatascience.com/understanding-variational-autoencoders-vaes-f70510919f73

rs2969182 (SNP) - Genes and regulation - Homo_sapiens. (n.d.). Ensembl Genome Browser 109.

https://asia.ensembl.org/Homo_sapiens/Variation/Mappings?db=core;r=17:11506644-11507644;v=rs2969182;vdb=variation;vf=104807943

rs4681292 (SNP) - Genes and regulation - Homo_sapiens. (n.d.). Ensembl Genome Browser 109. Retrieved April 3, 2023, from https://asia.ensembl.org/Homo_sapiens/Variation/Mappings?db=core;r=3:144411303-144412303;v=rs4681292;vdb=variation;vf=92680878

Sarwar, S., Shabana, Sajjad, K., & Hasnain, S. (2023). Genetic studies in the Pakistani population reveal novel associations with ventricular septal defects (VSDs). *BMC Pediatrics*, *23*(1). https://doi.org/10.1186/s12887-023-03851-3

Sharma, G., & Prabha, C. (2021). Applications of machine learning in cancer prediction and prognosis. In *Cancer Prediction for Industrial IoT 4.0: A Machine Learning Perspective* (pp. 119–135). Chapman and Hall/CRC. http://dx.doi.org/10.1201/9781003185604-8

Shisa6 shisa family member 6 [Mus musculus (house mouse)] - Gene. (n.d.). NCBI. Retrieved April 3, 2023, from https://www.ncbi.nlm.nih.gov/gene?Db=gene&Cmd=DetailsSearch&Term=380702

Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., & Salakhutdinov, R. (2014). Dropout: A simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research*, *15*(56), 1929–1958.

Tapak, L., Ghasemi, M. K., Afshar, S., Mahjub, H., Soltanian, A., & Khotanlou, H. (2023). Identification of gene profiles related to the development of oral cancer using a deep learning technique. *BMC Medical Genomics*, *16*(1). https://doi.org/10.1186/s12920-023-01462-6

The role of machine learning in advancing precision medicine with feedback control. (n.d.). *Cell Reports Physical Science*, *3*(11). https://doi.org/10.1016/j.xcrp.2022.101149

Visscher, P. M., Brown, M. A., McCarthy, M. I., & Yang, J. (2012). Five years of GWAS discovery. *The American Journal of Human Genetics*, *90*(1), 7–24. https://doi.org/10.1016/j.ajhg.2011.11.029

Wallach-Dayan, S. B., Izbicki, G., Cohen, P. Y., Gerstl-Golan, R., Fine, A., & Breuer, R. (2006). Bleomycin initiates apoptosis of lung epithelial cells by ROS but not by Fas/FasL pathway. *American Journal of Physiology-Lung Cellular and Molecular Physiology*, *290*(4), L790–L796. https://doi.org/10.1152/ajplung.00300.2004

Wang, X., Glubb, D. M., & O'Mara, T. A. (2022). 10 Years of GWAS discovery in endometrial cancer: Aetiology, function and translation. *EBioMedicine*, *77*, 103895. https://doi.org/10.1016/j.ebiom.2022.103895

Wu, G.-H., Smith-Geater, C., Galaz-Montoya, J. G., Gu, Y., Gupte, S. R., Aviner, R., Mitchell, P. G., Hsu, J., Miramontes, R., Wang, K. Q., Geller, N. R., Hou, C., Danita, C., Joubert, L.-M.,

Schmid, M. F., Yeung, S., Frydman, J., Mobley, W., Wu, C., … Chiu, W. (2023). CryoET reveals organelle phenotypes in huntington disease patient iPSC-derived and mouse primary neurons. *Nature Communications*, *14*(1). https://doi.org/10.1038/s41467-023-36096-w

Way, G. P., & Greene, C. S. (2018a). Extracting a biologically relevant latent space from cancer transcriptomes with variational autoencoders. *Pacific Symposium on Biocomputing. Pacific Symposium on Biocomputing*, *23*, 80–91.

Way, G. P., & Greene, C. S. (2018b). Extracting a biologically relevant latent space from cancer transcriptomes with variational autoencoders. *Pacific Symposium on Biocomputing. Pacific Symposium on Biocomputing*, *23*.

Zhang, X., Zhang, J., Sun, K., Yang, X., Dai, C., & Guo, Y. (2019, August 17). *Integrated multi-omics analysis using variational autoencoders: Application to pan-cancer classification*. arXiv.Org. https://arxiv.org/abs/1908.06278

Zlobina, K., Jafari, M., Rolandi, M., & Gomez, M. (2022b). The role of machine learning in advancing precision medicine with feedback control. *Cell Reports Physical Science*, *3*(11), 101149. https://doi.org/10.1016/j.xcrp.2022.101149

Supplementary Material

Supplementary explanation for the ML models

The Logistic Regression model

The logistic regression model analyses the relationship between the input variables and the output variable. Based on the input variables, the model estimates the probability that the observation belongs to a specific class. The output variable of our dataset is binary as it only contains two classes — control and case, represented by 0 and 1 respectively. The probability is modelled using the logistic function, also known as the sigmoid function, which is applied for binary classification, mapping the probability to an output value between 0 and 1. The logistic function has the following equation:

$$f(z) = \frac{e^z}{1+e^z} = \frac{1}{1+e^{-z}}$$

where z is a function of the input variables.

The variable z is a linear function of the input variables:

$$z = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + ... + \beta_k x_k$$

where X_k refers to the input variable, and β_k refers to the degree of change in the output variable for every unit change in X_k. The β parameters are the values that maximise the likelihood of observing the given data. This is a statistical method known as the Maximum Likelihood Estimation (MLE).

The Logistic Regression model was first built as it is one of the most popular and widely-used classification models for medical data (Dreiseitl, 2002). Logistic Regression is simple and easily understood, and can be computationally efficient, which is important as the large number of features of our dataset consumes computational resources extensively (Maaten & Hinton, n.d.). The model also generally handles outliers well, helping to reduce overfitting.

On the flip side, Logistic Regression lacks the ability to model more complex relationships between the input and output variables, and may not be apt if the dataset has non-linear relationships(Chang, 2020a). Alternative models such as Neural Networks may then easily outperform the Logistic Regression model (Dietterich, et al., 2006). Furthermore, when the

number of features easily exceeds the number of observations like in our dataset, a Logistic Regression model is highly likely to overfit (Babyak, 2004).

The Random Forest classifer

To aid in the understanding of a Random Forest model, we shall first introduce a Decision Tree. When a Decision Tree model is trained with aon the training dataset, it constructs test points and branches. Simply put, you can deem a Decision Tree as a cascading set of questions, where each question is asked at a test point, such as "Is the patient a male or female?", and depending on the answer, the new observation is either filtered to the "Male" or "Female" branch. This process then continues until the Decision Tree reaches a conclusion regarding which class the new observation likely fits in.

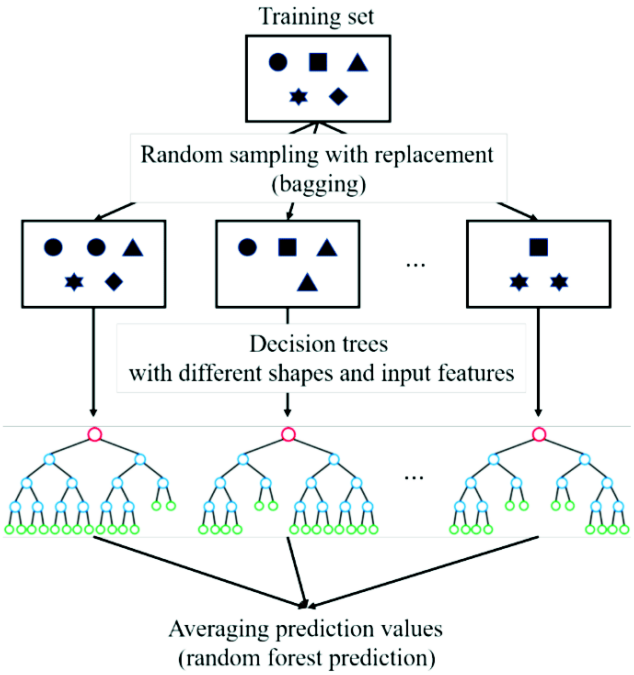
The question asked at each test point is chosen based on the attribute that returns the lowest entropy, which measures the impurity of an attribute, or the highest information gain, which measures the purity of an attribute. Given our case/control variable Y, and the two possible categorical values it can take, (y_1, y_2), representing case and control respectively, the following depicts the formula to determine the entropy:

$$D_Y = -\sum_{j=1}^{k} P(Y = y_j) log_2 P(Y = y_j)$$

where $P(Y = y_j)$ is the probability of class $Y = y_j$.

The tree is built recursively until a specific criterion is met, such as when the preset maximum depth is reached, or when the tree cannot be split further as the information gain does not exceed the minimum purity threshold set.

The Random Forest model utilises a Bagging ensemble learning algorithm, also known as Bootstrap Aggregating, where multiple different Decision Trees are trained in parallel on randomly selected samples of the dataset with replacement (Figure 1). The binary prediction of each Decision Tree is summed and the model makes the final prediction via a majority vote. If more Decision Trees predict the class of the new observation to be 0, the final prediction will be 0; while if more Decision Trees predict the class to be 1, the final prediction made will be of class 1. This ensemble algorithm hence outperforms a single Decision Tree.



Supplementary Figure 1: Each Decision Tree trains on a random subset of the data and the prediction of each tree is averaged to acquire the final prediction (Lee, n.d.)

Random Forests are less prone to overfitting than a singular Decision Tree model since randomness is introduced in the training process. Each Decision Tree in the Random Forest trains on a subset of data that is randomly selected, forcing the trees to learn different aspects of the data, and minimising the chance that the tree will learn from noise.

The Gradient Boosted Trees

Random Forests are more interpretable and less likely to overfit compared to the Gradient Boosted Trees, inclusive of the XGBoost, CatBoost and LightGBM models. However, Gradient Boosted Trees may exhibit better performance as they train multiple trees sequentially, where each sequential tree is fitted in a way that reduces the error of the previous tree. To illustrate, a single Decision Tree is first fitted to the data and used to make predictions. The errors of this first tree are then calculated, with the second tree being trained to correct these errors. The process repeats and the final prediction is computed by summing the predictions of all trees, with the prediction of each tree weighted based on its accuracy.

The CatBoost classifer is the only framework out of the three that builds a symmetric tree. For each test question asked, the next branching factor must be the same condition. This typically results in reduced overfitting and faster computation.

XGBoost grows level-wise, while LightGBM grows leaf-wise. This essentially means that LightGBM can learn the training data more closely, but is much more prone to overfitting than XGBoost. The level-wise algorithm of XGBoost keeps its trees balanced, regularising the training process. (Alshari et al., 2021)

Gradient boosted Trees are generally more accurate compared to other less robust models such as Logistic Regression, train faster on larger datasets and can also handle missing values well, which is a tremendous challenge that comes with our dataset. Though Gradient Boosted Trees are prone to overfitting, the models do offer various regularisation[21] methods to help prevent overfitting. To elucidate, the L1 regularisation method adds a penalty term to the loss function of the model, which measures how well a model predicts for the given dataset. This addition reduces the dimensionality of the dataset and simplifies the model, reducing the likelihood of overfitting. Nonetheless, these frameworks require more tuning of hyperparameters[22] to achieve optimal performance and can be computationally expensive and take a long time when training on a large dataset.

Artificial Neural Network

Artificial Neural Networks have neurons that process information and communicate with one another via having weighted connections between different neurons. The process of feeding the neural network the input data and comparing the prediction with the true output for training is iterated many times until the loss function is minimized. This process also adjusts the weights between different neurons accordingly to depict the importance of that feature in determining the final output.

ANN is good for finding complex non-linear relationships between inputs and outputs. However, they also have several disadvantages, including being difficult to interpret, requiring large amounts of data to learn effectively, being prone to overfitting, taking a long time to train, and being difficult to design and optimise (Mijwil, 2018).

Hyperparameters used for tuning the ML models

[21] A set of techniques to help prevent ML models from overfitting, e.g. L1 and L2 regularisation.

[22] Parameters that are set before training a model and cannot be learnt from the data during training.

Supplementary Table 1. Hyperparamters used to tune the different classifier models

Model	Hyperparameters Tuned
Logistic Regression	penalty, c, solver and max_iter
Random Forest	hyperparameters n_estimators, max_features, max_depth, min_samples_split, min_samples_leaf and bootstrap
CatBoost	depth, learing_rate, iterations, l2_leaf_reg, subsample, colsample_bylevel, bagging_temperature, random_strength, border_count, loss_function, and early_stopping_rounds
XGBoost	learning_rate, max_depth, min_child_weight, subsample, colsample_bytree, gamma, reg_alpha, reg_lambda, scale_pos_weight, early_stopping_rounds and n_estimators
LightGBM	learning_rate, n_estimators, num_leaves, boosting_type, max_bin, colsample_bytree, subsample, reg_alpha, reg_lambda, class_weight, bagging_fraction, bagging_freq, max_depth, feature_fraction, l2_leaf_reg
ANN	batch_size and optimiser

VAE Training Criterion

The model was trained with a loss function *L* that is composed of two terms:

$$L = L_{Reconstruction} + L_{Kl-divergence}$$

The first term is the reconstruction loss, $L_{Reconstruction}$, which is based on the Mean-squared error (MSE) between the input *X* and the reconstructed output $\hat{X}$. Again, the choice of the error function is mostly empirical. We experimented on binary cross-entropy loss (BCE), absolute error loss (L1), MSE loss, and cross-entropy loss, and chose the one with the significantly lowest loss, which is the MSE loss function.

The second term is the Kullback-Leibler divergence (KL-divergence) between the learnt prior and the posterior distributions[23], with a β attached (Cetin et al., 2023).

Overview of non-genomic data

Supplementary Table 2.

Factors	**Description**	**Type of Data**	**Reason for being chosen**
Ethnicity variance	A numerical value that determines the "biological race" deduced by several	Numerical	Significantly affects outcome of BIT in preliminary Logistic Regression

[23] Prior distributions refer to the distributions of the original data, while posterior distributions refer to the distributions of data sampled from the latent space.

	genomic regions.		
Age	Age of patients obtained from their date of births	Numerical	Age has significant ffect on the immune system and drug-induced toxicity (Mangoni & Jackson, 2003)
Age at treatment	Age where the patient received treatment	Numerical	Age has significant effect on the immune system and drug-induced toxicity (Mangoni & Jackson, 2003)
Gender	-	Categorical	Gender can significantly affect immune system
Race	There were people from various races such as Chinese, Malays, Indians, Sikh and many others	Categorical	Significantly affects outcome of BIT in preliminary Logistic Regression

Weight	-	Numerical	Weight has been found to affect the immune system (https://www-ncbi-nlm-nih-gov.libproxy1.nus.edu.sg/pmc/articles/PMC7609120/#:~:text=Obesity%20is%20considered%20as%20a,degree%20of%20obesity18%E2%80%9321.)
Stage of Diagnosis	There were people from all stages of Cancer: I, II, III, IV	Categorical	*P-value* < 0.05 in Chi-squared test
Total number of cycles of treatment	There are 1-6 cycles used in total in the treatments	Numerical	Significantly affects outcome of BIT in preliminary Logistic Regression and has a *P-value* < 0.05 in Chi-squared test

Cumulative dosage of treatment.	Total amount of Bleomycin received so far	Numerical	*P-value* < 0.05 in Chi-squared test