Progress in Microbes and Molecular Biology

HH PUBLISHER

Method

Support Vector Machine – Recursive Feature Elimination (SVM-RFE) for Feature Selection on Multi-omics Lung Cancer Data.

Ji Tong Lin [1], Azurah A Samah [2]

[1]School of Computing, Faculty of Engineering, Universiti Teknologi Malaysia, Johor Bahru, 81310, Malaysia, jitonglin1998@gmail.com

[2]Artificial Intelligence and Bioinformatics Group (AIBIG), School of Computing, Faculty of Engineering, Universiti Teknologi Malaysia, Johor Bahru, 81310, Malaysia, azurah@utm.my

Article History

Received: XX Month 20XX;

Received in Revised Form: XX Month 20XX;

Accepted: XX Month 20XX;

Available Online: XX Month 20XX

Abstract: Biological data obtained from sequencing technologies are growing at an exploding rate. The curse of dimensionality is an inevitable obstruction in studying biological data such as the classification of omics data. The study focuses on mitigating the curse of dimensionality by implementing SVM-RFE as the selected feature selection method in the lung cancer (LUSC) omics dataset comprises of genomics, transcriptomics and epigenomics. In this research, the LUSC datasets first undergo data pre-processing including checking for missing value, normalization and removing zero variance features. The cleaned LUSC datasets are then integrated to form a multi-omics dataset. Feature selection is performed on the LUSC multi-omics data using SVM-RFE to select the several optimal feature subsets. The 5 smallest feature subsets (FS) are used in classification using SDAE and VAE neural networks to assess the quality of the feature subsets. The results show that all 5 VAE models are able to obtain an accuracy and AUC score of 1.000 while only 2 out of 5 SDAE models (FS 1000 & 4000) are able to do so. 3 out of 5 SDAE models have an AUC score of 0.500, indicating zero capability in separating the binary class labels. The study concludes that for this specific study, a fine-tuned supervised learning VAE model has better capability in classification tasks compared to SDAE models, and FS 1000 and 4000 are the two most optimal feature subsets selected by the SVM-RFE algorithm as both the SDAE and VAE models built with these feature subsets are achieving the best classification results.

Keywords: Multi-omics Analysis, Support Vector Machine – Recursive Feature Elimination (SVM-RFE), Stacked Denoising Autoencoder (SDAE), Variational Autoencoder (VAE)

1. Introduction

Lung cancer is among the fatal cancers in the world. In 2018, it accounted for 11.6% of total cancer cases and is expected to take the lives of 1.8 million people, making it the leading cause of cancer death worldwide [1, 2]. Fortunately, the advancement of artificial intelligence technology came into rescue by being applied in the form of machine learning and deep learning models in multi-omics data, which has brought improvement to the current prognosis of lung cancer [1].

The advancement of technology has come a long way, making the internet a vast space of information. Such vast information is considered as big data. In biology, the advanced sequencing techniques caused an explosion in the rate and the amount of data being extracted known as omics data. Omics refers to several field of studies in life sciences that focus on large amount of information to understand life [3]. Multi-omics on the other hand is formed when two or more omics types are combined to form a single big picture. The integration of multiple omics allows the study of biological phenomenon in a more holistic way. This has in turn improved the prognosis and predictive accuracy of disease phenotype, allowing a better treatment and prevention of cancers to be facilitated [4].

Nowadays biological data can be extracted easily with the latest sequencing technologies as mentioned previously however, it still requires an enormous amount of effort to truly extract insightful information from the vast amount of raw data. In multi-omics disciplinary field, insightful information could be obtained by studying the more complex mechanism across molecular layers [5]. Even if massive benefits could potentially be obtained from the studies of multi-omics, it is still a puzzle to the researchers in which effective multi-omics integration methods are still immature. Historically, many analytical tools and experimental designs are carefully developed for single omics disciplinary field and they are not capable of appropriate comparison or intelligent integration across multiple omics discipline [6].

Multi-omics data analysis usually involves the three major challenges: 1) the curse of dimensionality, 2) difference in scale, sampling and bias for each omics, and 3) effectively extract insightful information via appropriate multi-omics integration methods [7]. The study primarily focuses on the curse of dimensionality of multi-omics data, or also known as the large *p* small *n* problem, whereby the multi-omics dataset has a small number of samples (*n*) with large number of features (*p*) [8]. The nature of multi-omics data analysis that requires the researchers to merge multiple omics data into one usually limits the number of observations for the multi-omics data [7], as the integration process requires the data from the same individual or patient to exist in every omics type involved in the study [9].

To address the curse of dimensionality challenge, the study employs the use of Support Vector Machine – Recursive Feature Elimination (SVM-RFE) as the feature selection algorithm. With large dimensionality, the multi-omics data prone to contain redundant or low-predictive power features that might hinders the machine learning or deep learning methods to produce optimal classification result. The study aims to use SVM-RFE to select only the relevant features from the multi-omics data of lung cancer for the development of better deep learning classifiers.

2. Materials and Methods

The experimental workflow of the research is summarized in Figure 1. In general, the procedure of the study starts with data acquisition, followed by data cleaning, multi-omics integration, feature selection, and classification. The results and findings of the study is then discussed.

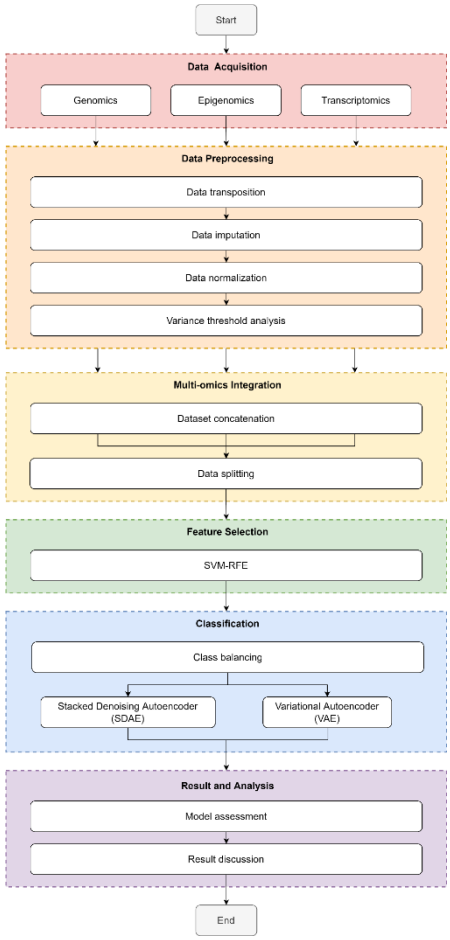


Figure 1. The experimental workflow of the research.

2.1 Dataset

The lung cancer omics dataset used in this study is retrieved from an open source website http://acgt.cs.tau.ac.il/multi_omic_benchmark/download.html. The dataset comes in a package of 4 different files, which are the 3 omics datasets (i.e. gene expression, DNA methylation expression & miRNA expression) and 1 clinical dataset. All 4 datasets contain the patient ID, which is important for column concatenation in multi-omics integration. A quick summary of the dimensions of each dataset is tabulated in Table 1.

Table 1. Summary of the dimensions of the raw lung cancer omics datasets

Data / omics type	Number of instances	Number of features
Gene expression (genomic)	20531	553
DNA methylation expression (epigenomics)	5000	413

miRNA (Transcriptomics)	1046	388
Clinical data	626	127

The class labels contained in the clinical dataset is binary, whereby it contains one positive outcome (has lung cancer) and one negative outcome (no lung cancer). The positive outcome is denoted as "Primary Tumour" while the negative outcome is denoted as "Solid Tissue Normal".

2.2 Data Preparation

The acquired datasets undergo data cleaning process to prepare the data for further analysis. The 2 type of datasets (i.e. omics dataset & clinical dataset) are cleaned differently. The omics datasets are cleaned by performing data transposition, data imputation, data normalization and variance threshold analysis. The patient ID and the class label are extracted from the clinical data.

2.2.1 Omics Data Cleaning

Despite being labeled as pre-processed, the omics datasets still contain certain caveats which requires further processing. First of all, the rows and columns of the data of the raw omics dataset are inverted and misleading. Data transposition is performed on all 3 omics datasets to correct the orientation of the data to be represented. After data transposition, the rows now represent the samples/instances corresponds to the patient ID, while the columns now represent the omics expression values. The dimensions of the omics datasets before and after data transposition is summarized in Table 2.

Table 2. The dimensions of the omics datasets before and after data transposition

Dataset	**Dimension (row, column)**	
	Raw dataset	**After Data Transposition**
Gene expression	(20531, 552)	(552, 20531)
DNA Methylation	(5000, 412)	(412, 5000)
miRNA	(1046, 387)	(387, 1046)

Next, data imputation is performed. The steps involved are checking for duplicated rows and missing values (or NaN), and impute them with value zero as missing values tend to reduce the statistical power of the study and produces biased estimations [10]. The result of the checking shows that all 3 omics datasets contain neither duplicated rows nor missing values.

Next, the omics datasets undergo data normalization. The values of each feature in each omics datasets are adjusted and scaled between 0 and 1 to improve the quality of both the data and the machine learning model [11].

The data cleaning phase ends with variance threshold (VT) analysis. Zero variance features (i.e. features with only one unique value or the value for each sample in a particular feature are the same) are dropped as they do not provide any predictability to the output class [12]. A total of 287 and 160 zero variance features are removed from the gene expression and

miRNA expression omics data respectively. The DNA methylation expression dataset on the other hand contains no zero variance feature. The summary of the VT analysis is shown in Table 3.

Table 3. The dimensions of the omics datasets before and after variance threshold analysis

Dataset	**Dimension (row, column)**	
	Before VT	**After VT**
Gene expression	(552, 20531)	(552, 20244)
DNA Methylation	(412, 5000)	(412, 5000)
miRNA	(387, 1046)	(387, 886)
Clinical Data	(626, 127)	(626, 127)

2.2.2 Clinical Data Cleaning

The cleaning of the clinical dataset is as simple as extracting the patient ID column along with the class label column. The patient ID is necessary to attach the omics expression data and the class label together.

2.2.3 Attaching Class Label

With the cleaned omics datasets and clinical dataset, the two need to be combined. The cleaned omics dataset only contains the omics expression data without the class label. Therefore, the class label column from the clinical data is attached to each omics datasets according to the patient ID.

2.3 Multi-omics Integration

In multi-omics integration, the columns from each cleaned dataset are concatenated by using the patient ID as an index. Meaning, the integrated multi-omics dataset will only contain the information of the patients whose information are present in all 4 datasets. The summary of the datasets before and after data preparation and multi-omics integration is shown in Table 4.

Table 4. Summary of the datasets before and after data preparation and multi-omics integration

Dataset	**Dimension of the Dataset (row, column)**			
	Raw dataset	**Data Transposition**	**Variance Threshold**	**Multi-omics Integration**
Gene expression	(20531, 552)	(552, 20531)	(552, 20245)	(344, 26131)
DNA Methylation	(5000, 412)	(412, 5000)	(412, 5000)	
miRNA	(1046, 387)	(387, 1046)	(387, 886)	
Clinical Data	(626, 127)	(626, 127)	(626, 1)	

It is worth noting that the class label distribution before and after multi-omics integration has changed drastically. Table 5 summarizes the class label distribution for each omics

dataset including the integrated multi-omics data. Before integration, each single omics dataset has a class label distribution of around 90:10 for Primary Tumour and Solid Tissue Normal. The distribution changed into 99:1 when the multi-omics data is integrated. The multi-omics data is now severely imbalanced.

Table 5. Class label distribution with percentage for each omics dataset

Omics	Primary Tumour	Solid Tissue Normal	Total Sample
Gene expression	501 (90.8%)	51 (9.2%)	552
DNA methylation	370 (89.8%)	42 (10.2%)	412
miRNA expression	342 (88.4%)	45 (11.6%)	387
Multi-omics Data	**341 (99.1%)**	**3 (0.9%)**	**344**

2.4 Data Splitting

The integrated multi-omics dataset undergoes data splitting as shown in Figure 2. First, the multi-omics data is split into train-test set with ratio of 70:30, which empirically produces the best result [13]. For feature selection with SVM-RFE (Figure 2(a)), the train set is used in stratified 2-fold cross validation (CV). For classification with SDAE and VAE (Figure 2(b)), the train set is further split into train and validation set with 70:30 ratio.

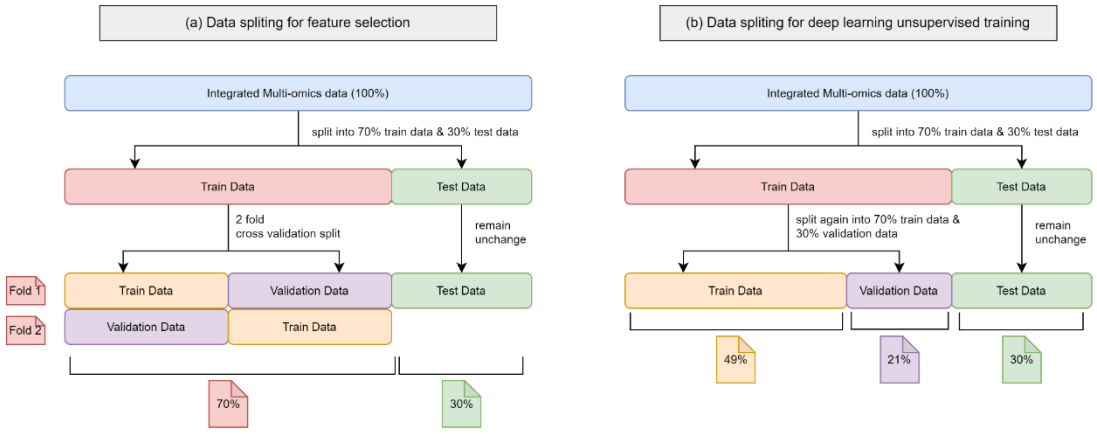


Figure 2. The class distribution for the multi-omics data before (**a**) and after (**b**) SMOTE

2.5 Feature Selection

With the integrated multi-omics data, the study proceeds with feature selection using SVM-RFE. SVM-RFE is responsible for selecting the n most relevant features, whereby n is the number of features to be selected. The study mains to select several subsets of features to assess the most optimal number of features to be selected specific to this study. A total of 20 feature subsets (abbreviated as FS from now on) are selected, which ranges from 20000, 19000, 18000, …, 1000.

The most optimal set of hyperparameters for SVM-RFE is determined using grid search. The hyperparameter grid to be used in the search is summarized in Table 6 [14, 15]. The "C" parameter controls the tradeoff between the correctly classified instances and the capability of the hyperplane to separate instances. "linear" kernel is the only kernel that produces feature importance as one of its output for the RFE algorithm to rank the feature. "step" on the other

hand is the hyperparameter for RFE, whereby it decides the number of features to remove in each iteration.

Table 6. Hyperparameter grids used for SVM-RFE

Hyperparameter	Values
C	0.1, 1, 10, 100
Kernel	linear
Step	1, 2, 3

To obtain early insight regarding the 20 selected feature subsets, an SVM model with the similar set of hyperparameters shown in Table 6 is used to perform classification on each feature subset. 2-fold CV is employed to obtain a more generalized result. The omics composition is also observed for each feature subset.

Ultimately, the output of the SVM-RFE algorithm is the 20 selected feature subsets, which are used as inputs for the deep learning models for classification.

2.6 SMOTE

The issue with data imbalance for the integrated multi-omics data addressed in Chapter 2.3 is handled here. The study employs a data oversampling technique via the synthesis of new data based on existing data, namely SMOTE, on the training set. The hyperparameter chosen for SMOTE is listed in Table 7. "sampling_strategy" is set to 1 so that the newly synthesized instances with minority class label (Solid Tissue Normal) will match the number of the instances with "Primary Tumour". "k_neighbors" decides the number of nearest data points to use as references to synthesize new data points. It is forced to set to 1 since "k_neighbour" has to be smaller than the number of minority class. "random_state" is set to 42 to allow reproducible result.

Table 7. Hyperparameters used for SMOTE on train data

Hyperparameters	Settings
sampling_strategy	1
k_neighbors	1
random_state	42

The result of SMOTE on the training set is depicted in Figure 3. Now, the training set for the multi-omics data is balanced with equal number of samples on either class label. However, the testing set is still severely imbalanced. Figure 4 shows the comparison of the class distribution between the training and testing set.

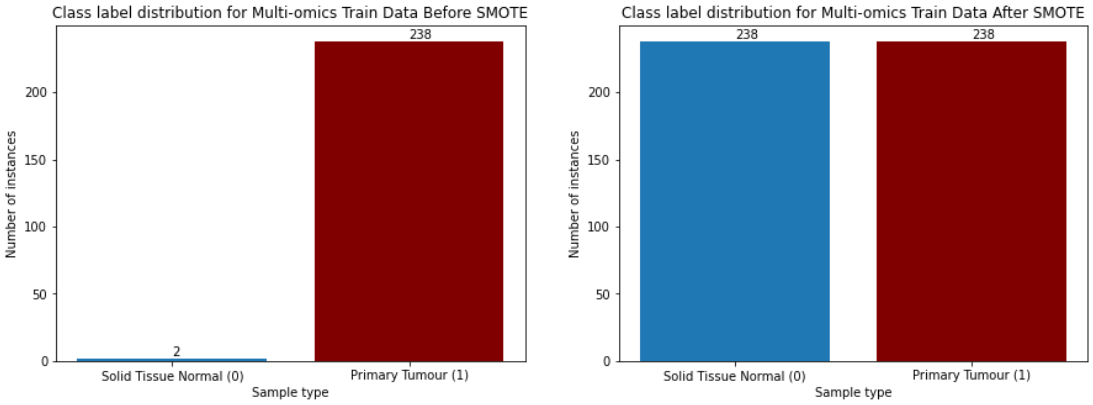


Figure 3. The class distribution for the multi-omics data before and after SMOTE

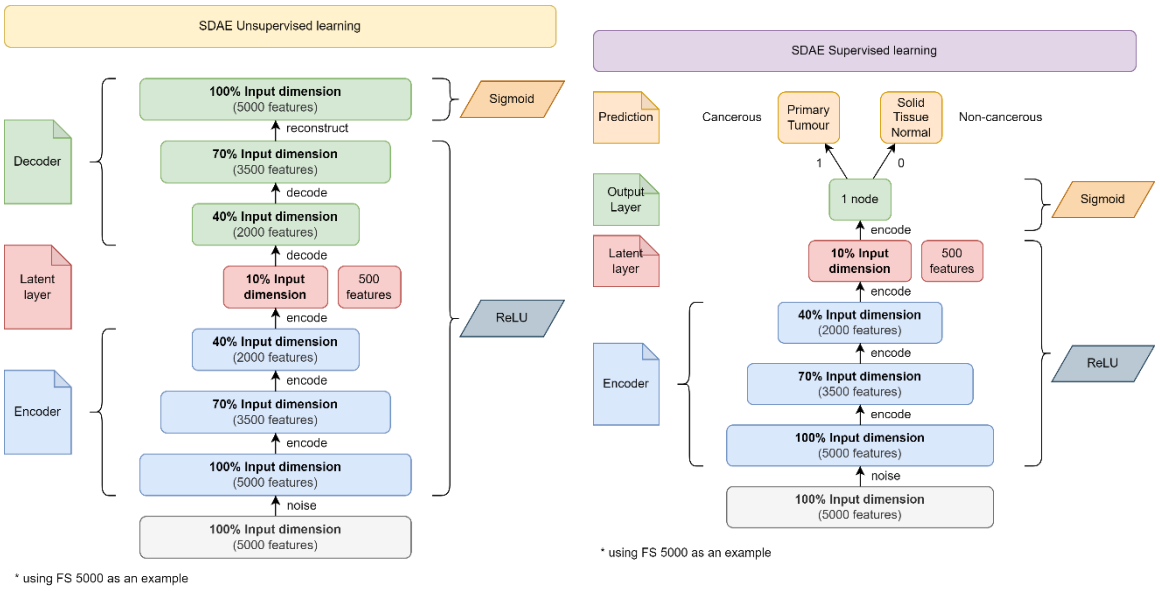
Figure 4. The comparison of class distribution for the training and testing set of the multi-omics data

2.7 Deep Learning Models

The study includes two deep learning models: SDAE and VAE to validate and assess the feature subsets selected by SVM-RFE in Chapter 2.4. In previous study, smaller feature subsets produced better classification results [15]. At the same time, due to hardware constraint, whereby the memory for the GPU used is insufficient for large neural network, the study only incorporates FS 5000, 4000, 3000, 2000 and 1000 in classification. The setup for the experiment using the two models are specified in their respective subchapters.

2.7.1 SDAE

The SDAE model building starts with unsupervised learning. The main for unsupervised learning is to train the SDAE model to learn the important features from each feature subset and by encoding them into a smaller dimension (latent layer). The model loss during the unsupervised learning phase is recorded to assess the capability of the model to reconstruct the given inputs. Figure 5(a) shows the neural network of the SDAE model during unsupervised learning.



(**a**) (**b**)

Figure 5. The neural network structure of the SDAE model for (**a**) unsupervised training and (**b**) supervised training

The hyperparameters used for the SDAE model during unsupervised training is tabulated in Table 8. There is a total of 8 layers in the neural network, including the gaussian noise layer. Each layer in the encoder part of the model reduces the dimension of the input by 30%. At the latent layer, the original input is encoded into 10% of its original dimension. For the decoder part, each layer increases the dimension by 30%. The dimension is restored to 100% at the output layer, in which the original input is attempted to be reconstructed. The epoch and batch size are set to 50 and 16 respectively, which is observed to allow the model to minimize the model loss to a converging point [16]. With reference to [17], the activation functions used the hidden layers and the output layer are set to "ReLU" and "sigmoid" respectively. "adam" optimizer is used as it is the most recommended optimizer and is being adapted as the benchmark for deep learning models [18]. Binary cross entropy is used as the loss function as the objective of the SDAE model is classification [19]. The gaussian noise layer introduced at the input layer uses 10% of dropout rate to aid the model learning [20].

Table 8. Hyperparameters used for SDAE during unsupervised and supervised training

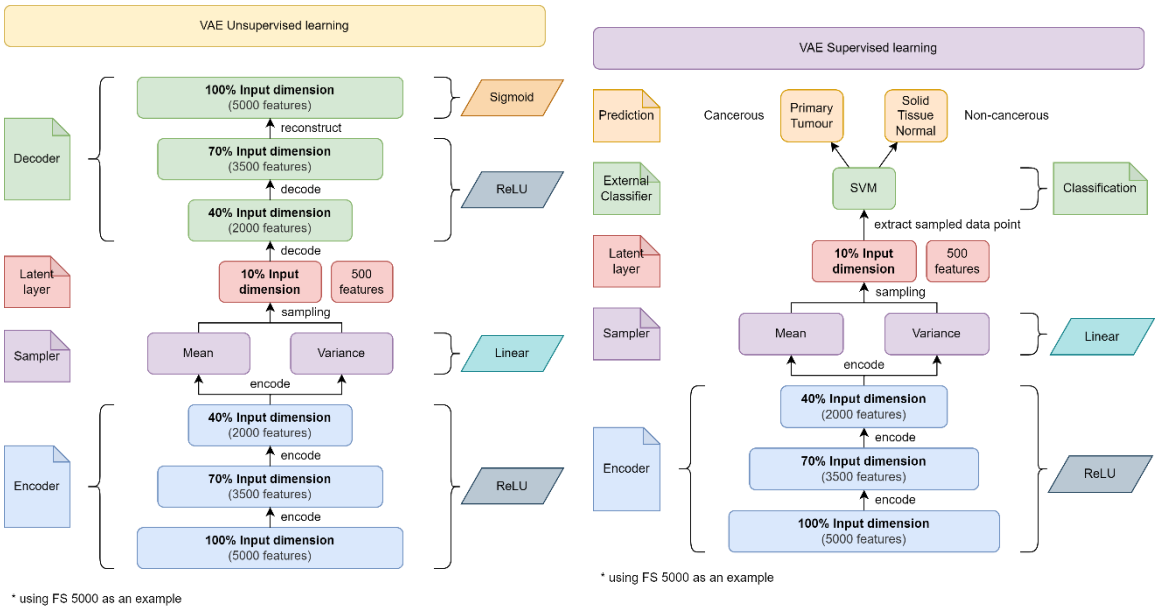
Hyperparameters	**Unsupervised Learning**	**Supervised Learning**
Layers	*5000, 5000 (noisy), 3500, 2000, 500, 2000, 3500, 5000* 100%, 100% (noisy), 70%, 40%, 10%, 40%, 70%, 100%	*5000, 5000 (noisy), 3500, 2000, 500, 1* 100%, 100% (noisy), 70%, 40%, 10%, 1
Epoch	50	50
Batch size	16	16
Optimizer	adam	adam

Activation functions	ReLU – Hidden layers Sigmoid – Last layer (output layer)	ReLU – Hidden layers Sigmoid – Last layer (output layer)
Loss function	binary cross entropy	binary cross entropy
Gaussian Noise Dropout Rate	10%	10%

The unsupervised learning SDAE model is then fine-tuned into a supervised learning model. This is done by replacing the decoder part of the model with a new layer that contains only 1 node with sigmoid activation function as shown in Figure 5(b). This layer acts as the new output layer and allows the SDAE model to output values between 0 and 1 to represent the binary classes (Solid Tissue Normal & Primary Tumour), which turns the model into a supervised learning model capable of performing classification. The hyperparameters used during the unsupervised training phase are kept unchanged except for the layers.

2.7.2 VAE

The VAE model building follows a similar fashion. It also starts with unsupervised learning to learn the features from each feature subset and encodes them into a smaller dimension. A sampler is incorporated to generate new data points according to the mean and variance learnt from the previous layers. The model then tries to reconstruct the original input according to the sampled data. Similarly, the VAE model is assessed based on the model loss Figure 6 shows the neural network of the VAE model during unsupervised learning.



(**a**) (**b**)

Figure 6. The neural network structure of the VAE model for (**a**) unsupervised training and (**b**) supervised training

Table 9 shows the hyperparameters used for the unsupervised training of the VAE model. The hyperparameters used are very similar to those used in the unsupervised learning of the SDAE model. however, the difference between the two models lies in the activation functions and the loss functions. With reference to [17] in their work, the activation function for the

sampler is set to "linear". Two different loss function are used for the VAE model. The generative loss measures the overall reconstruction loss of the model, while the Kullback-Leibler (KL) loss measures the difference between the two probability distributions. The total loss of the VAE model is the combination of both the loss functions [17].

Table 9. Hyperparameters used for VAE during unsupervised training

Hyperparameters	**Unsupervised Learning**
Layers	*5000, 3500, 2000, 500, 2000, 3500, 5000* 100%, 70%, 40%, 10%, 40%, 70%, 100%
Epoch	50
Batch size	16
Optimizer	adam
Activation functions	ReLU (Rectified Linear Unit) – Hidden layers Linear – Bottleneck layer (latent layer) Sigmoid – Last layer (output layer)
Loss function	Generative loss Kullback-Leibler (KL) loss

Classification cannot be done directly by the VAE model itself by using the similar fine-tuning method in the supervised learning of the SDAE model. This is due to the fact that the accuracy produced fluctuates around 50%, which is not in line with the baseline accuracy of the data. An external classifier is used to aid the VAE model in classification. This is done by extracting the data points sampled by the sampler in the latent layer, and feeding them to the external classifier. In this study, SVM is the chosen external classifier. To keep it simple, the hyperparameters of the SVM model is kept as default as shown in Table 10.

Table 10. Hyperparameters used for SVM as external classifier

Hyperparameters	**Description**
C	1.0
Kernel	linear

3. Results

The output of the SVM-RFE algorithm is shown here. Besides that, the SDAE and VAE models are built with the selected feature subsets by the SVM-RFE and the classification results are shown.

3.1. SVM-RFE

The With the grid search implemented to determine the best set of hyperparameters for the SVM-RFE, it has been determined that the most optimal set of hyperparameter are $C =$

0.1, linear kernel and $step = 1$ as shown in Table 11. The total computation time for the SVM-RFE to finish selected 20 feature subsets is recorded at 3 hours and 9 minutes.

Table 11. The selected set of hyperparameters for the feature selection using SVM-RFE for each feature subset

Feature Subset	Computation Time
C	0.1
kernel	linear
step	1

The classification result on the 20 feature subsets is shown in Figure 7. It is observed that from FS 20000 to 14000, the mean accuracy is recorded at 0.996. The mean accuracy for FS 13000 to 1000 on the other hand is recorded at 1.000.

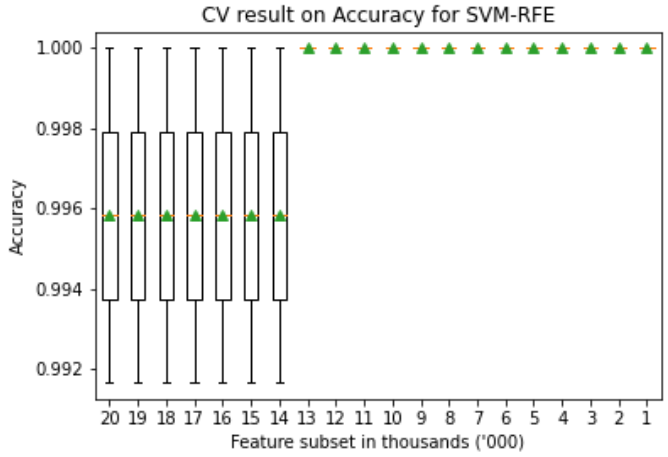


Figure 7. Boxplot for the CV result (accuracy) for each feature subset

The omics composition for each feature subset is depicted in Figure 8. The general trend observed is that the composition of gene expression omics goes down (75.08% to 70.50%) as the size of feature subset reduces, while the composition of DNA methylation expression rises (22.77% to 27.94%). This trend is observed until FS 5000, in which the trend is observed to go the opposite way whereby the composition of gene expression goes up while the composition of DNA methylation goes down. The initial composition of miRNA expression at 2.15% on the other hand fluctuates as the size of feature subset goes down until it settles at 1.8%

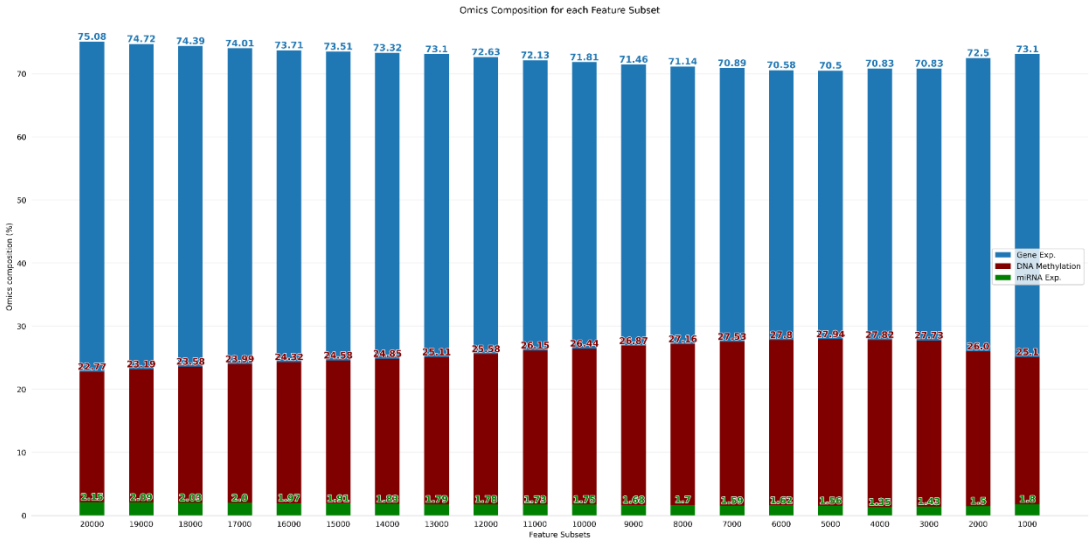


Figure 8. Omics composition after SVM-RFE represented in bar chart

3.2 SDAE

The model loss of the developed SDAE models using FS 1000 to 5000 during the unsupervised learning phase is recorded in Figure 9. Generally, the model loss for each SDAE models are low at below 0.5. It is noted that only the model loss for the validation in FS 5000 resembles closer to the training set compared to the SDAE models built with other feature subsets.

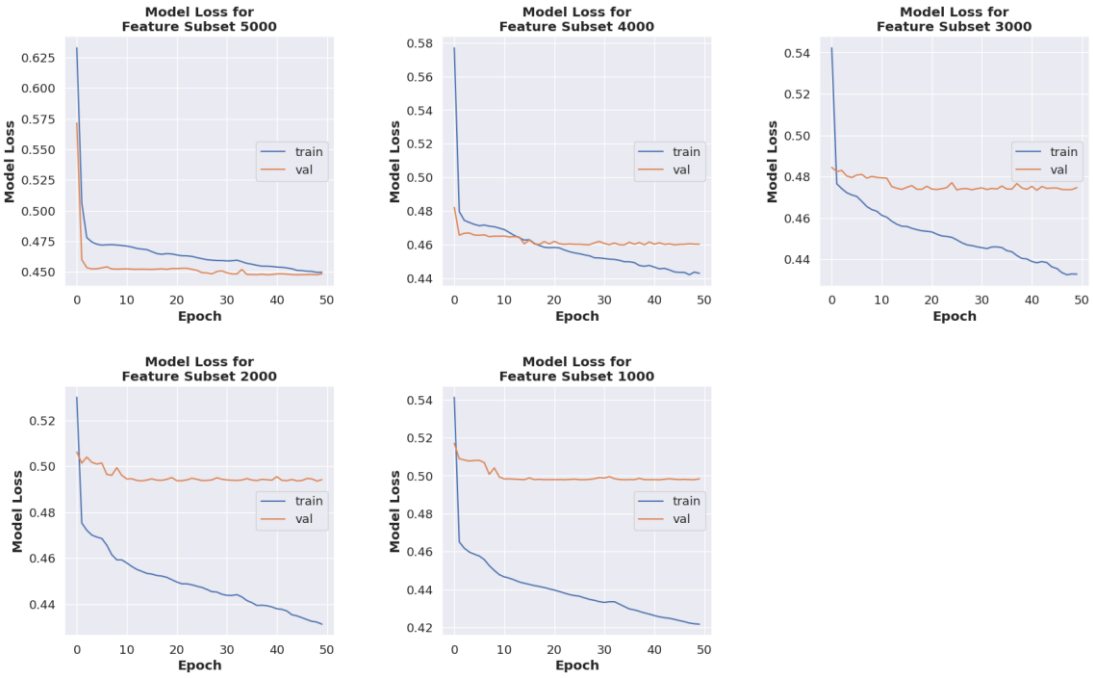
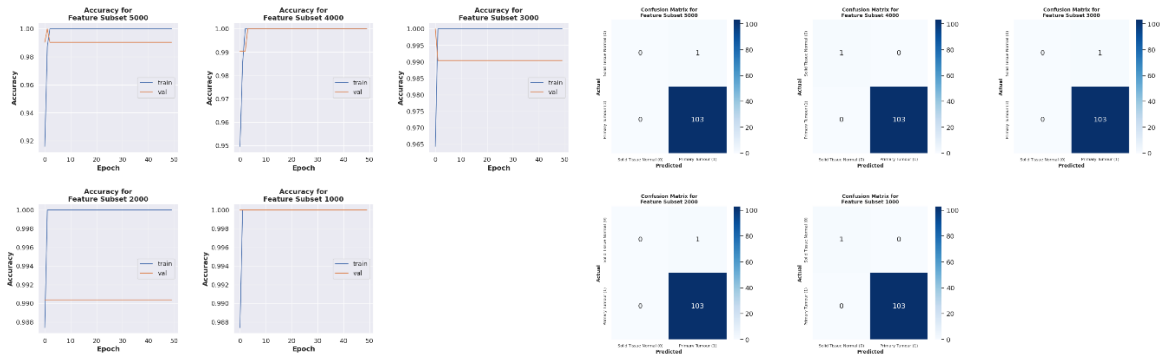


Figure 9. Model loss for the unsupervised learning for the SDAE model

The classification result of the fine-tuning supervised learning SDAE models are recorded. Figure 10(a) shows the accuracy of the SDAE models with respect to epoch, while Figure 10(b) shows the confusion matrix obtained from the classification result of

the last epoch. It is observed that FS 1000 and 4000 are able to correctly classify all the instances correctly, while the FS 2000, 3000 and 5000 are unable to classify the negative class label correctly, resulting in one false positive. With the confusion matrix, the accuracy, AUC score, precision, recall and F1 score are tabulated in Table 12.



(**a**) (**b**)

Figure 10. The classification result for the fine-tuned supervised learning SDAE model. (**a**) The accuracy for supervised learning for the SDAE model. (**b**) The confusion matrix from the classification using the fine-tuned SDAE model

Table 12. Metrics obtained from the classification result of SDAE model

Feature Subset	Accuracy	AUC score	Precision	Recall	F1 score
5000	0.9904	0.5000	0.9904	1.0000	0.9951
4000	1.0000	1.0000	1.0000	1.0000	1.0000
3000	0.9904	0.5000	0.9904	1.0000	0.9951
2000	0.9904	0.5000	0.9904	1.0000	0.9951
1000	1.0000	1.0000	1.0000	1.0000	1.0000

3.3 VAE

The unsupervised learning VAE models are developed using FS 1000 to 5000. The total model loss of the model is obtained using the combination of the generative loss and the KL loss and shown in Figure 11. Generally, the total model loss for the VAE models decreases as the size of feature subset used to develop the VAE models decreases. It is observed that both the total model loss for the training and validation sets are able to converge, but fluctuation is also observed for the validation sets.

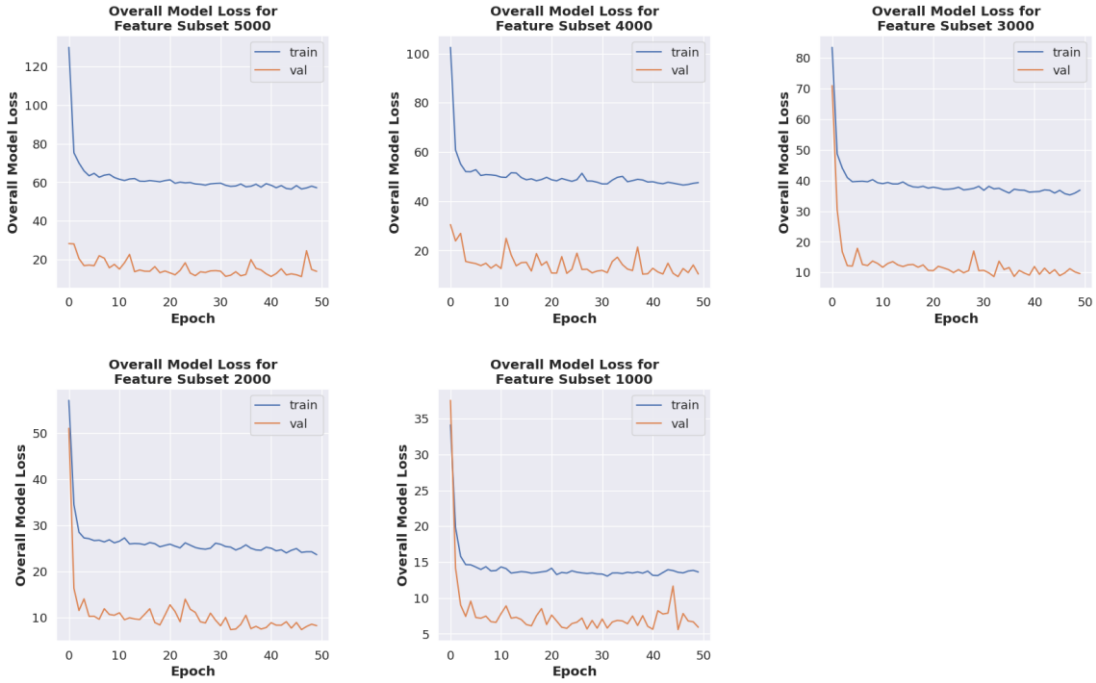


Figure 11. The overall loss of the VAE model during unsupervised learning phase

The classification of the VAE model involves the use of an external classifier, which is an SVM model. The study first built a supervised learning VAE model using the same method applied for the fine-tuning of the SDAE model. However, it is observed that the accuracy produced by the fine-tuned VAE models is around 50%, which is far below the baseline accuracy for this testing set at 99.04% (refer Table 13). This leads to the use of an external classifier for the classification of the VAE model.

To achieve this, the sampled data by the sampler in the latent space for each VAE model are extracted and fed to the SVM classification model. The hyperparameters used for the SVM model are kept as default at C = 1 with linear kernel. The classification of the sampled data by the VAE model using the SVM classifier is shown in the form of confusion matrix in Figure 12.

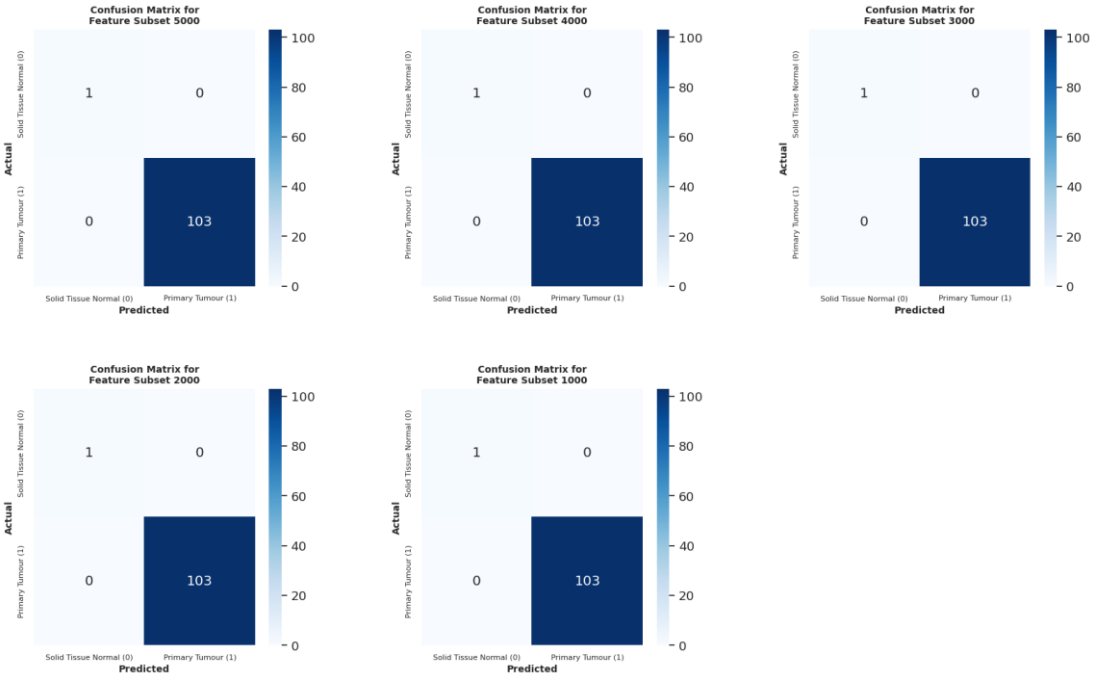


Figure 12. Confusion matrices produced using the classification results of SVM as the external classifier on the encoded inputs in the VAE model

The accuracy, AUC score, precision, recall and F1 score are calculated from the confusion matrix. The results are tabulated in Table 13. In Table 13, the metrics from the fine-tuned VAE model and the SVM model are displayed side-by-side to demonstrate the difference in performance between the two classification methods.

Table 13. Comparison between the metrics obtained from the classification result of the fine-tuned supervised learning VAE model and the classification using SVM

Feature Subset	**Accuracy**		**AUC (ROC)**		**Precision**		**Recall**		**F1 score**	
	FT	**SVM**	**FT**	**SVM**	**FT**	**SVM**	**FT**	**SVM**	**FT**	**SVM**
5000	0.500	1.000	0.252	1.000	0.981	1.000	0.505	1.000	0.667	1.000
4000	0.481	1.000	0.738	1.000	1.000	1.000	0.476	1.000	0.645	1.000
3000	0.462	1.000	0.728	1.000	1.000	1.000	0.456	1.000	0.627	1.000
2000	0.529	1.000	0.267	1.000	0.982	1.000	0.534	1.000	0.692	1.000
1000	0.471	1.000	0.238	1.000	0.980	1.000	0.476	1.000	0.605	1.000

4. Discussion

According to the boxplot in Figure 7, it could be understood that, the SVM-RFE has removed more less relevant features as the size of the feature subsets decreases. Therefore, the smaller feature subsets contain higher density of relevant features, which allows the SVM classifier to obtain 100% accuracy.

As for the omics composition after feature selection, the decline of the composition of gene expression and the rise of the composition of DNA methylation expression from FS 20000 to 5000 could indicate that, as the feature subset becomes smaller, the SVM-RFE algorithms removed more less relevant features from gene expression omics while keeping the more relevant features from DNA methylation expression omics. As the size of the feature subsets continues to decline, the opposite is observed, whereby the gene expression features become more relevant than that of the DNA methylation expression, causing the SVM-RFE to remove more features from DNA methylation expression, which results in the decline in composition.

The baseline accuracy for the testing set used is 0.9904 or 99.04%. This is due to the fact that the testing set with 104 instances only have 1 instance with the negative class label (Solid Tissue Normal). Therefore, by simply predicting only the positive class (Primary Tumour), an accuracy of 0.9904 can be achieved. From the classification results of the SDAE models built with FS 1000 to 5000, it is observed that only FS 1000 and 4000 are able to achieve 100% accuracy with all correctly classified instances. FS 2000, 3000 and 5000 on the other hand failed to predict the negative class label correctly, resulting in 1 false positive prediction. Despite that, these models still achieved an accuracy of 0.9904. In this case, accuracy might not be a useful metric as the concern here is to correctly predict the minority negative class. The AUC score on the other hand is a more useful metric here since it scores according to the predicted outcome for both class labels. The AUC score for FS 1000 and 4000 are recorded at 1.000 since all the instances are correctly classified. On the other hand, FS 2000, 3000 and 5000 achieved 0.5000 for their AUC score, indicating zero capability in separating the class labels.

According to the classification result on the extracted sampled data from the VAE models for each feature subset using the SVM model as an external classifier, each feature subset is able to achieve 1.000 accuracy. With all correctly classified instances, the rest of the metrics are also measured at 1.000.

When comparing the classification results of the SDAE and VAE models, it could be deduced that the VAE models are more capable at learning the useful feature of each feature subset during the encoding process. FS 1000 and 4000 are the two feature subsets which allowed both the SDAE and VAE models to achieve the score of 1.000 for each metrics. This could potentially indicate that FS 1000 and 4000 are the two most optimal feature subset selected by the SVM-RFE algorithm.

5. Conclusions

The study has employed the use of SVM-RFE for feature selection to extract the most relevant features from the multi-omics data with large dimension. The output obtained from the SVM-RFE are the 20 most optimal feature subsets selected by the algorithm. The 5 feature subsets with the smallest size are then used in cancer classification using the fine-tuned supervised learning SDAE and VAE deep learning models. The result suggests that FS 1000 and 4000 are the two most optimal feature subsets selected by the SVM-RFE algorithm

as both the SDAE and VAE classifiers are able to correctly classify all the instances using the testing set.

Funding: Please add: "No external funding was provided for this research" or "This work was funded by the XX with grant number XXX".

Acknowledgments: In preparing this thesis, I was in contact with many people, researchers, academicians, and practitioners. They have contributed towards my understanding and thoughts. In particular, I wish to express my sincere appreciation to my main thesis supervisor, Dr. Azurah binti A Samah, for encouragement, guidance, critics and friendship. I am also very thankful to Mr. Hairudin bin Abdul Majid for his guidance, advices and motivation. Without their continued support and interest, this thesis would not have been the same as presented here. My fellow student should also be recognised for their support. My sincere appreciation also extends to all my colleagues and others who have aided me at various occasions. Their views and tips are useful indeed. Unfortunately, it is not possible to list all of them in this limited space. I am grateful to all my family member.

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

References

1. Gao, Y., Zhou, R., & Lyu, Q. (2020). Multiomics and machine learning in lung cancer prognosis. Journal of thoracic disease, 12(8), 4531-4535. doi:10.21037/jtd-2019-itm-013

2. Xu, Y., She, Y., Li, Y., Li, H., Jia, Z., Jiang, G., . . . Duan, L. (2020). Multi-omics analysis at epigenomics and transcriptomics levels reveals prognostic subtypes of lung squamous cell carcinoma. Biomedicine & Pharmacotherapy, 125, 109859. doi:https://doi.org/10.1016/j.biopha.2020.109859

3. Yadav, S. P. (2007). The wholeness in suffix -omics, -omes, and the word om. Journal of biomolecular techniques : JBT, 18(5), 277-277. Retrieved from https://pubmed.ncbi.nlm.nih.gov/18166670

4. Subramanian, I., Verma, S., Kumar, S., Jere, A., & Anamika, K. (2020). Multi-omics Data Integration, Interpretation, and Its Application. Bioinformatics and biology insights, 14, 1177932219899051-1177932219899051. doi:10.1177/1177932219899051

5. Ulfenborg, B. (2019). Vertical and horizontal integration of multi-omics data with miodin. BMC Bioinformatics, 20(1), 649. doi:10.1186/s12859-019-3224-4

6. Pinu, F. R., Beale, D. J., Paten, A. M., Kouremenos, K., Swarup, S., Schirra, H. J., & Wishart, D. (2019). Systems Biology and Multi-Omics Integration: Viewpoints from the Metabolomics Research Community. Metabolites, 9(4), 76. doi:10.3390/metabo9040076

7. El-Manzalawy, Y., Hsieh, T.-Y., Shivakumar, M., Kim, D., & Honavar, V. (2018). Min-redundancy and max-relevance multi-view feature selection for predicting ovarian cancer survival using multi-omics data. BMC Medical Genomics, 11(3), 71. doi:10.1186/s12920-018-0388-0

8. Huynh, P.-H., Nguyen, V. H., & Do, T.-N. (2020). Improvements in the Large p, Small n Classification Issue. SN Computer Science, 1(4), 207. doi:10.1007/s42979-020-00210-2

9. Taguchi, Y. h., & Turki, T. (2022). Novel feature selection method via kernel tensor decomposition for improved multi-omics data analysis. BMC Medical Genomics, 15(1), 37. doi:10.1186/s12920-022-01181-4

10. Kang, H. (2013). The prevention and handling of the missing data. Korean J Anesthesiol, 64(5), 402-406. doi:10.4097/kjae.2013.64.5.402

11. Singh, D., & Singh, B. (2020). Investigating the impact of data normalization on classification performance. Applied Soft Computing, 97, 105524. doi:https://doi.org/10.1016/j.asoc.2019.105524

12. Kuhn, M., & Johnson, K. (2019). Feature Engineering and Selection: A Practical Approach for Predictive Models (1st ed.). Chapman and Hall/CRC.

13. Gholamy, A., Kreinovich, V., & Kosheleva, O. (2018). Why 70/30 or 80/20 Relation Between Training and Testing Sets: A Pedagogical Explanation.

14. Huang, M.-L., Hung, Y.-H., Lee, W. M., Li, R. K., & Jiang, B.-R. (2014). SVM-RFE Based Feature Selection and Taguchi Parameters Optimization for Multiclass SVM Classifier. The Scientific World Journal, 2014, 795624. doi:10.1155/2014/795624

15. Guyon, I., Weston, J., Barnhill, S., & Vapnik, V. (2002). Gene Selection for Cancer Classification using Support Vector Machines. Machine Learning, 46(1), 389-422. doi:10.1023/A:1012487302797

16. Keskar, N. S., Mudigere, D., Nocedal, J., Smelyanskiy, M., & Tang, P. T. P. (2016). On Large-Batch Training for Deep Learning: Generalization Gap and Sharp Minima. CoRR, abs/1609.04836. Retrieved from http://arxiv.org/abs/1609.04836

17. Hira, M. T., Razzaque, M. A., Angione, C., Scrivens, J., Sawan, S., & Sarkar, M. (2021). Integrated multi-omics analysis of ovarian cancer using variational autoencoders. Scientific Reports, 11(1), 6265. doi:10.1038/s41598-021-85285-4

18. Gupta, A. (2021). A Comprehensive Guide on Deep Learning Optimizers. Retrieved from https://www.analyticsvidhya.com/blog/2021/10/a-comprehensive-guide-on-deep-learning-optimizers/

19. Ruby, U., & Yendapalli, V. (2020). Binary cross entropy with deep learning technique for Image classification. International Journal of Advanced Trends in Computer Science and Engineering, 9. doi:10.30534/ijatcse/2020/175942020

20. Gondara, L. (2016). Medical Image Denoising Using Convolutional Denoising Autoencoders. 2016 IEEE 16th International Conference on Data Mining Workshops (ICDMW), 241-246.