Progress in Microbes and Molecular Biology

HH PUBLISHER

Original Research Article

Support Vector Machine – Recursive Feature Elimination for Feature Selection on Multi-omics Lung Cancer Data

Nuraina Syaza Azman[1], Azurah A Samah[1*], Ji Tong Lin[1], Hairudin Abdul Majid[1], Zuraini Ali Shah[1], Nies Hui Wen[1], Chan Weng Howe[1]

Article History

Received: 10 March 2023;

Received in Revised Form: 28 March 2023;

Accepted: 30 March 2023;

Available Online: 4 April 2023

[1]Faculty of Computing, Universiti Teknologi Malaysia, Johor Bahru, 81310, Johor, Malaysia; nsyaza7@graduate.utm.my (NSA); jitonglin1998@gmail.com (JTL); hairudin@utm.my (HAM); aszuraini@utm.my (ZAS); huiwennies@utm.my (NHW); cwenghowe@utm.my (CWH)

*Corresponding author: Azurah A Samah; Faculty of Computing, Universiti Teknologi Malaysia, 81310 Johor Bahru, Johor, Malaysia; azurah@utm.my (AAS)

Abstract: Biological data obtained from sequencing technologies is growing exponentially. Multi-omics data is one of the biological data that exhibits high dimensionality, or more commonly known as the curse of dimensionality. The curse of dimensionality occurs when the dataset contains many features or attributes but with significantly fewer samples or observations. The study focuses on mitigating the curse of dimensionality by implementing Support Vector Machine – Recursive Feature Elimination (SVM-RFE) as the selected feature selection method in the lung cancer (LUSC) multi-omics dataset integrated from three single omics dataset comprising genomics, transcriptomics and epigenomics, and assess the quality of the selected feature subsets using SDAE and VAE deep learning classifiers. In this study, the LUSC datasets first undergo data pre-processing, including checking for missing values, normalization, and removing zero variance features. The cleaned LUSC datasets are then integrated to form a multi-omics dataset. Feature selection was performed on the LUSC multi-omics data using SVM-RFE to select several optimal feature subsets. The five 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 can obtain an accuracy and AUC score of 1.000, while only 2 out of 5 SDAE models (FS 1000 & 4000) can 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 a fine-tuned supervised learning VAE model has better capability in classification tasks compared to SDAE models for this specific study. Additionally, 1000 and 4000 are the two most optimal feature subsets selected by the SVM-RFE algorithm. The SDAE and VAE models built with these feature subsets achieve 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

Despite the availability of various cancer treatments, cancer continues to be a leading cause of death worldwide. Among the fatal types of cancer is lung cancer [1]. In 2020, it accounted for 11.4% of total cancer cases and took the lives of around 1.8 million people [2]. Omics refers to several fields of study in life sciences that focus on much information to understand life [3]. Multi-omics, on the other hand, is formed when two or more omics types are combined to allow the study of the biological phenomenon in a more holistic way, which in turn improves the prognosis and predictive accuracy of disease phenotype, allowing a better treatment and prevention of cancers to be facilitated [4]. The study primarily focuses on the curse of dimensionality of multi-omics data, also known as the large *p* small *n* problem, whereby the multi-omics dataset has a small number of samples (*n*) and a large number of features (*p*) [5]. 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 [6], as the integration process requires the data from the same individual or patient to exist in every omics type involved in the study [7].

The study employs Support Vector Machine – Recursive Feature Elimination (SVM-RFE) as the feature selection algorithm to address this problem. The study aims to use SVM-RFE to select only the relevant features from the lung cancer multi-omics data to develop better deep-learning classifiers. Next, Stacked Denoising Autoencoder (SDAE) and Variational Autoencoder (VAE) are used in the binary classification of the selected feature subsets. The objectives of the study include: 1) to study and understand the algorithm of SVM-RFE, SDAE, and VAE, 2) to determine suitable parameters for the selected algorithms and apply appropriate fine-tuning methods to them, and 3) to validate and verify the performance of SVM-RFE using SDAE and VAE.

The remaining sections of this paper are structured as follows: Section II provides a comprehensive review of relevant literature for this study, Section III outlines the methods and procedures employed to conduct the analysis, Section IV presents the results and subsequent discussion of the model performance, and finally, Section V provides the conclusion.

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 are then discussed.

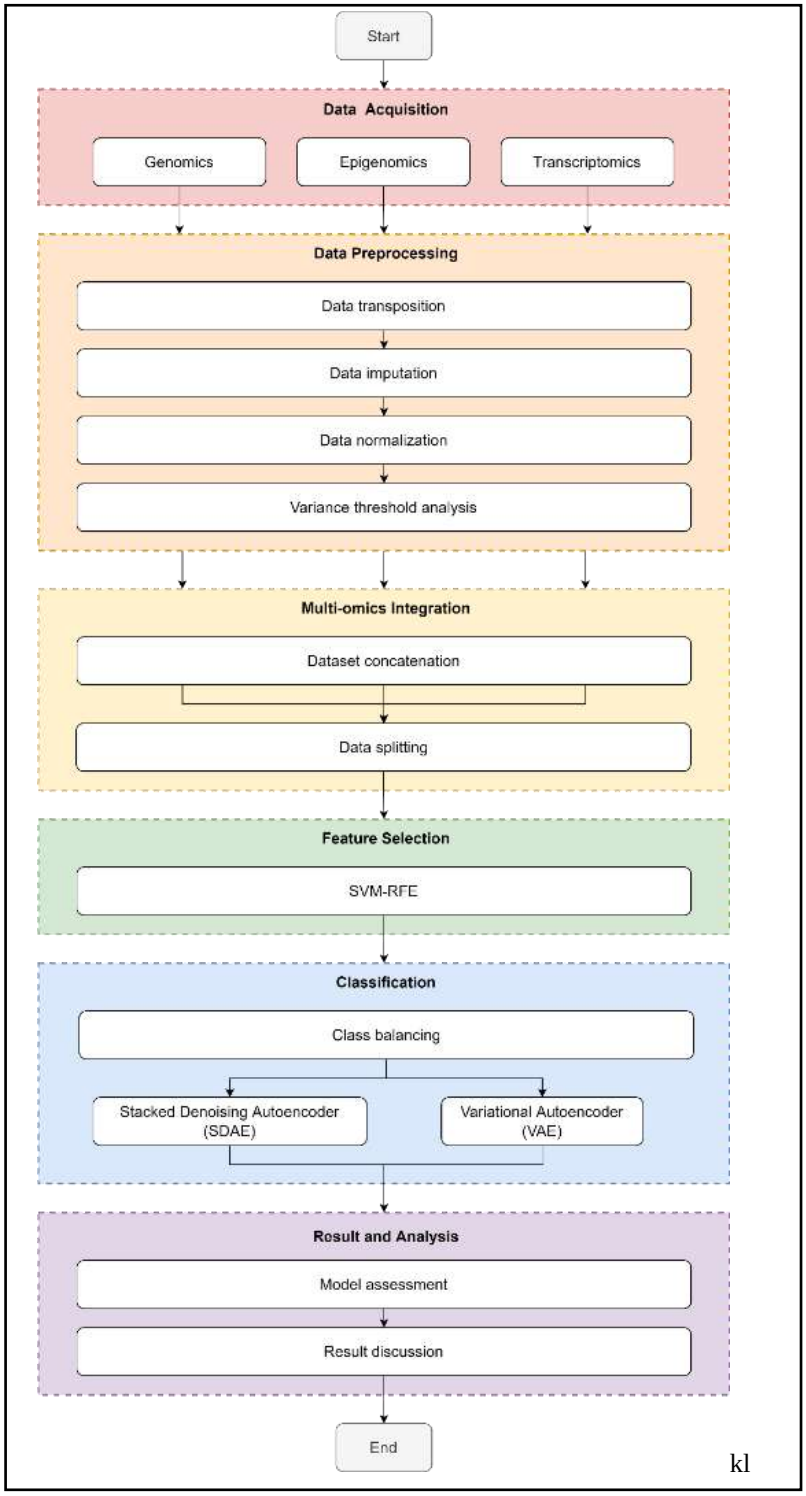

Figure 1. The experimental workflow of the research.

2.1. Data Acquisition

This study's lung cancer omics dataset 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 files: the 3 omics datasets (i.e., gene expression, DNA methylation expression & miRNA expression) and 1 clinical dataset. All 4 datasets contain the patient ID, vital 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 field	Num. of features	Num. of patients
Gene expression	Genomic	20531	553
DNA methylation	Epigenomics	5000	413
miRNA	Transcriptomics	1046	388
Clinical data	-	626	127

The class labels in the clinical dataset are binary, containing 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 Preprocessing

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

2.2.1. Data Transposition

Despite being labeled as pre-processed, the omics datasets still contain certain caveats which require further processing. First, 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 corresponding 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)

2.2.2. Data Imputation

The next step was data imputation. These steps involved checking for duplicated rows and missing values (or NaN) and imputing them with a value zero, as missing values tend to reduce the study's statistical power and produce biased estimations [8]. The result of the checking shows that all 3 omics datasets contain neither duplicated rows nor missing values.

2.2.3. Data Normalization and Variance Threshold Analysis

The omics datasets then undergo data normalization. The values of each feature in each omics dataset are adjusted and scaled between 0 and 1 to improve the data quality and the machine learning model [9]. 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 [10]. 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 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)

2.3. Multi-omics Integration

In multi-omics integration, the columns from each cleaned single omics 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 is 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 to 99:1 when the multi-omics data was 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

The integrated multi-omics dataset undergoes data splitting, as shown in Figures 2a and 2b. First, the multi-omics data is split into the train-test set with a ratio of 70:30, which empirically produces the best result [11]. 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 sets with a 70:30 ratio. The class distribution of the train and test set after data splitting are shown in Figure 3.

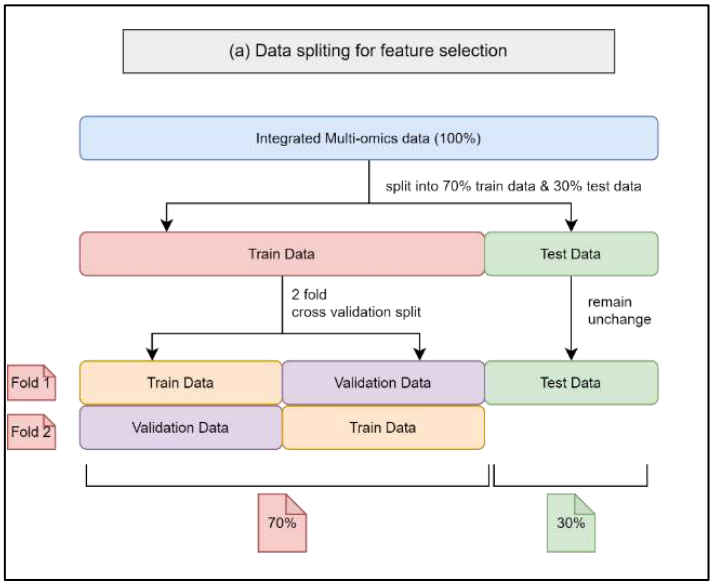

Figure 2a. The data splitting of the multi-omics data for feature selection.

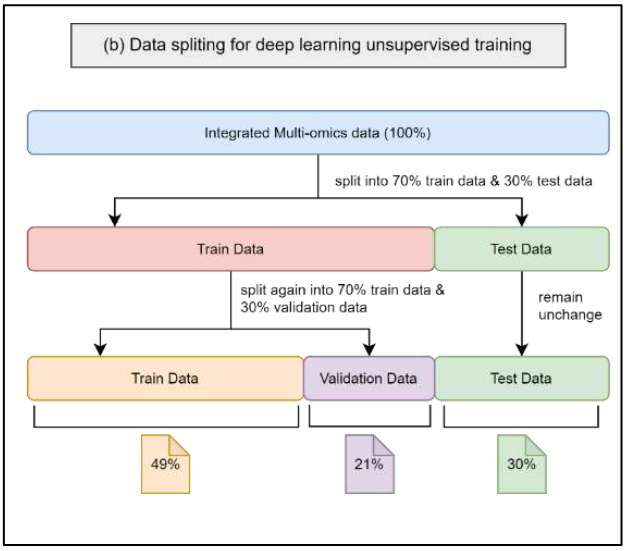

Figure 2b. Data splitting for deep learning unsupervised training.

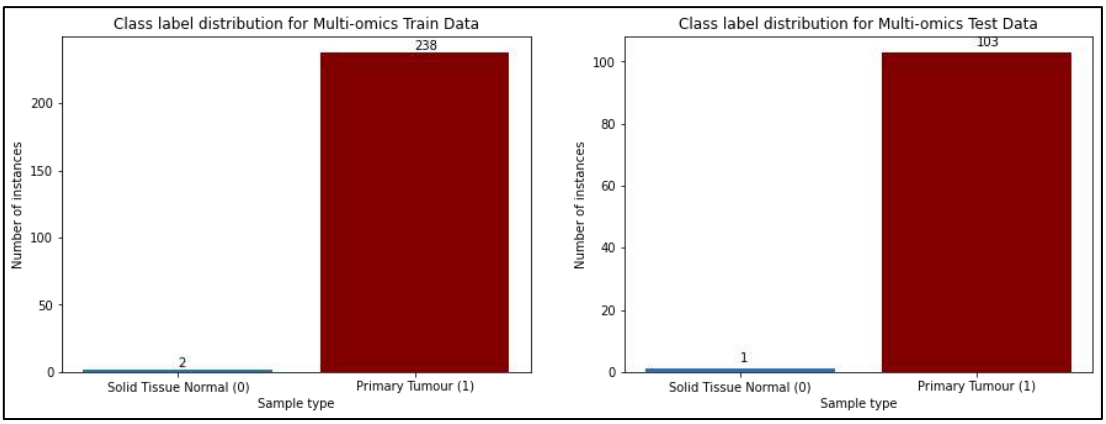

Figure 3. The class distribution of the multi-omics data for the train and test set.

2.4. Feature Selection

With the integrated multi-omics data, the study proceeds with a wrapper feature selection SVM-RFE. Wrapper methods use a classification algorithm to assess the significance of data features. The classifier is encapsulated in a search algorithm to identify the optimal subset of features [12]. SVM-RFE is responsible for selecting the n most relevant features, whereby n is the number of features to be selected. The study aims to select several subsets of features to assess the most optimal number of features to be explicitly selected for this study. A total of 20 feature subsets (abbreviated as FS from now on) are selected, which range from 20000, 19000, 18000, …, 1000.

The most optimal set of hyperparameters for SVM-RFE is determined using grid search. The hyperparameter grid used in the search is summarized in Table 6 [13]. 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 outputs for the RFE algorithm to rank the feature. "step" 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 a similar set of hyperparameters shown in Table 6 is used to classify each feature subset. A 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.

2.5. SMOTE

The issue with data imbalance for the integrated multi-omics data, addressed in 2.3 is handled here. The study employs a data oversampling, namely SMOTE, on the training set. SMOTE creates synthetic examples to oversample the minority class instead of replacing them with unfamiliar samples [14]. 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 classes. "random_state" is set to 42 to allow reproducible results.

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 4. Now, the training set for the multi-omics data is balanced with the equal number of samples on either class label. However, the testing set is still severely imbalanced. Figure 5 compares the class distribution between the training and testing set.

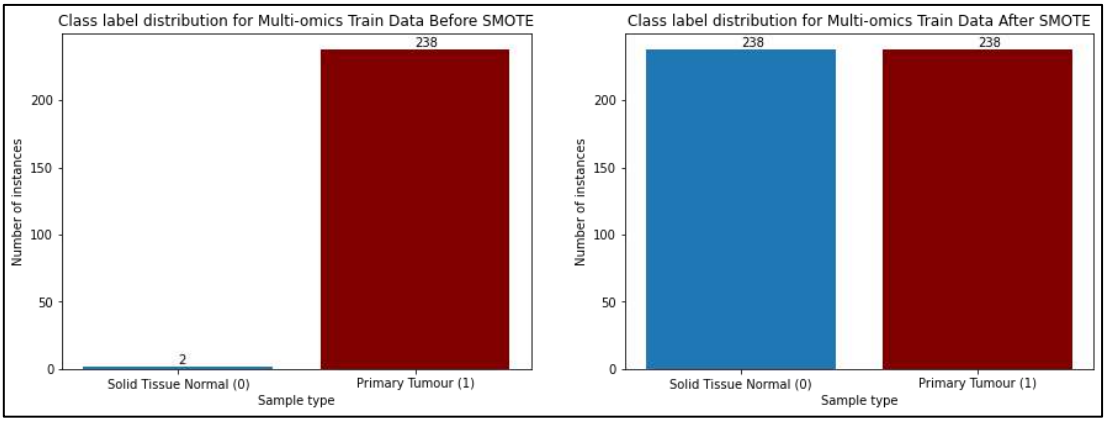

Figure 4. The class distribution for the multi-omics train set before and after SMOTE.

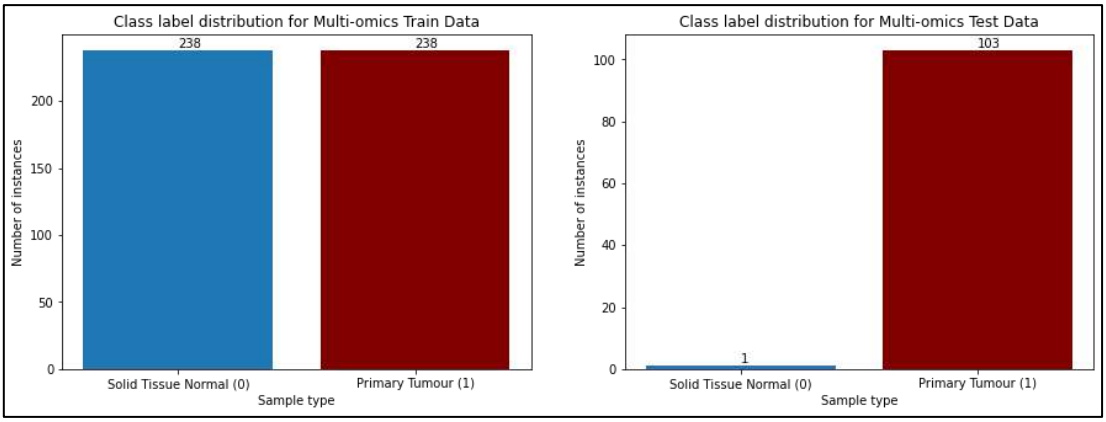

Figure 5. The comparison of class distribution for the training and testing set of the multi-omics data.

2.6. Deep Learning Models

Deep learning, which involves using artificial neural networks structured in layers to enable learning, has found application in cancer research [15-17] and other medical fields such as dental research and molecular biology [18–24]. The study includes two deep learning models: SDAE and VAE, to validate and assess the feature subsets selected by SVM-RFE in 2.4. At

the same time, due to hardware constraints, whereby the memory for the GPU used is insufficient for large neural networks, the study only incorporates FS 5000, 4000, 3000, 2000, and 1000 in classification. The setup for the experiment using the two models is specified in their respective subchapters.

2.6.1. SDAE

The SDAE model building starts with unsupervised learning. The main function for unsupervised learning is to train the SDAE model to learn the important features from each feature subset 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 6(a) shows the neural network of the SDAE model during unsupervised learning.

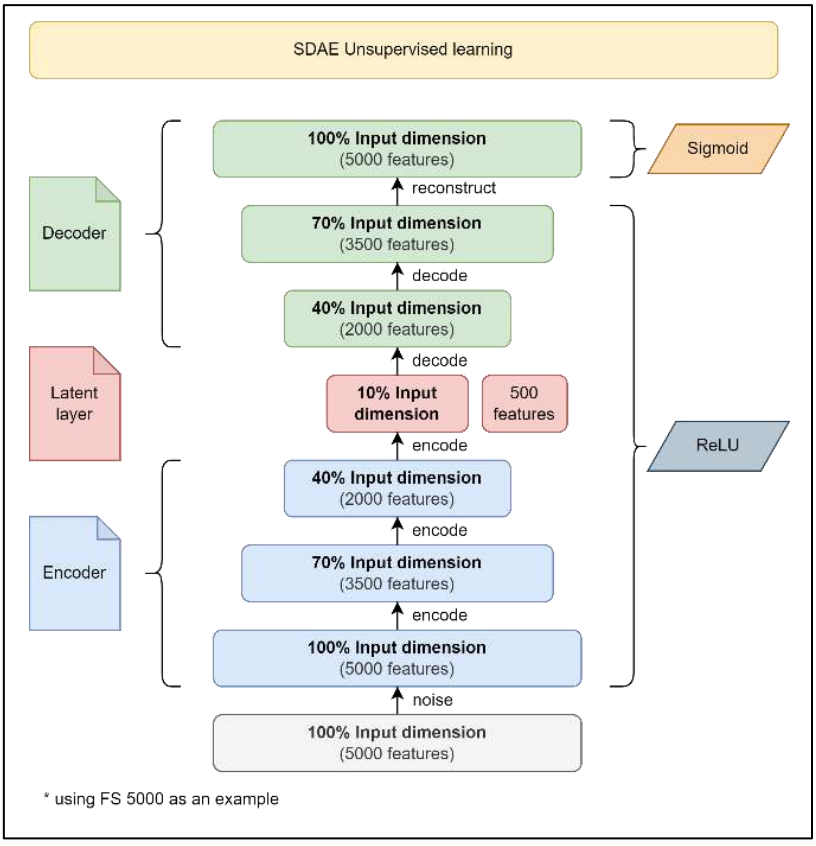

Figure 6a. The neural network structure of the SDAE model for unsupervised training.

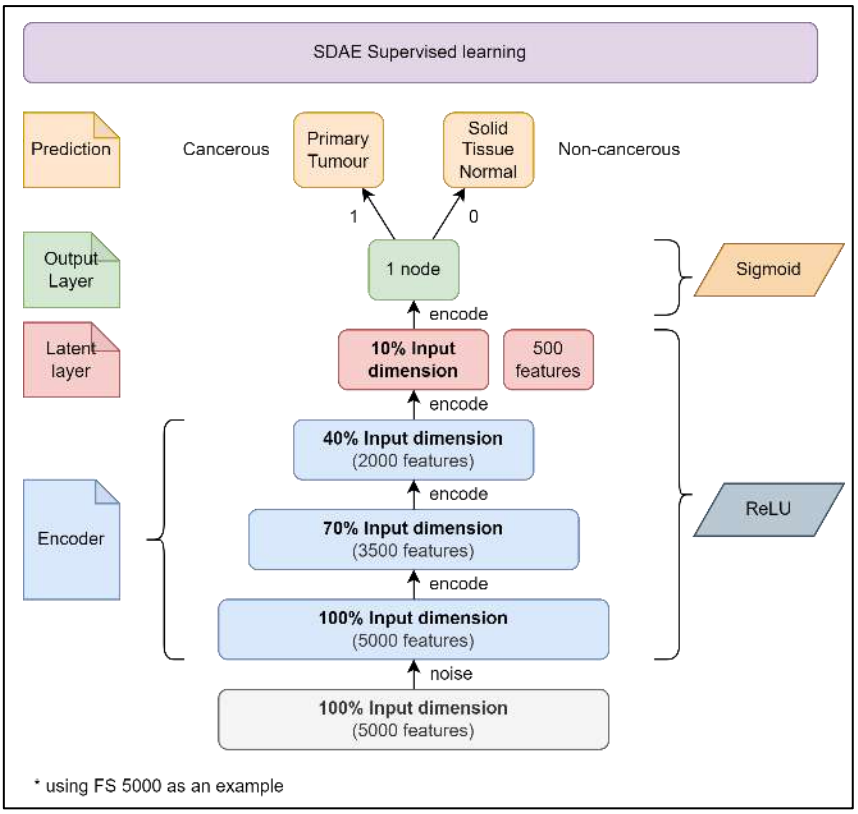

Figure 6b. The neural network structure of the SDAE model for supervised training.

The hyperparameters used for the SDAE model during unsupervised training is tabulated in Table 8. There are 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%. The original input is encoded into 10% of its original dimension at the latent layer. 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 [25]. With reference to [26], the activation functions used the hidden layers, and the output layer is 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 [27]. Binary cross entropy is used as the loss function as the objective of the SDAE model is classification [28]. The gaussian noise layer introduced at the input layer uses 10% of the dropout rate to aid the model learning [29].

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 a sigmoid activation function, as shown in Figure 6(b). This layer acts as the new output layer. It 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.

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	5000, 5000 (noisy), 3500, 2000, 500, 1
	100%, 100% (noisy), 70%, 40%, 10%, 40%, 70%, 100%	100%, 100% (noisy), 70%, 40%, 10%, 1
Epoch	50	50
Batch size	16	16
Optimizer	Adam	Adam
Activation functions	ReLU – Hidden layers	ReLU – Hidden layers
	Sigmoid – Last layer (output layer)	Sigmoid – Last layer (output layer)
Loss function	binary cross entropy	binary cross entropy
Gaussian Noise Dropout Rate	10%	10%

2.6.2. VAE

The VAE model building follows a similar fashion. It also starts with unsupervised learning to learn the features from each feature subset and encode them into a smaller dimension. A sampler is incorporated to generate new data points according to the mean and variance learned from the previous layers. The model then reconstructs the original input according to the sampled data [30]. Similarly, the VAE model is assessed based on the model loss. Figure 7a shows the neural network of the VAE model during unsupervised learning.

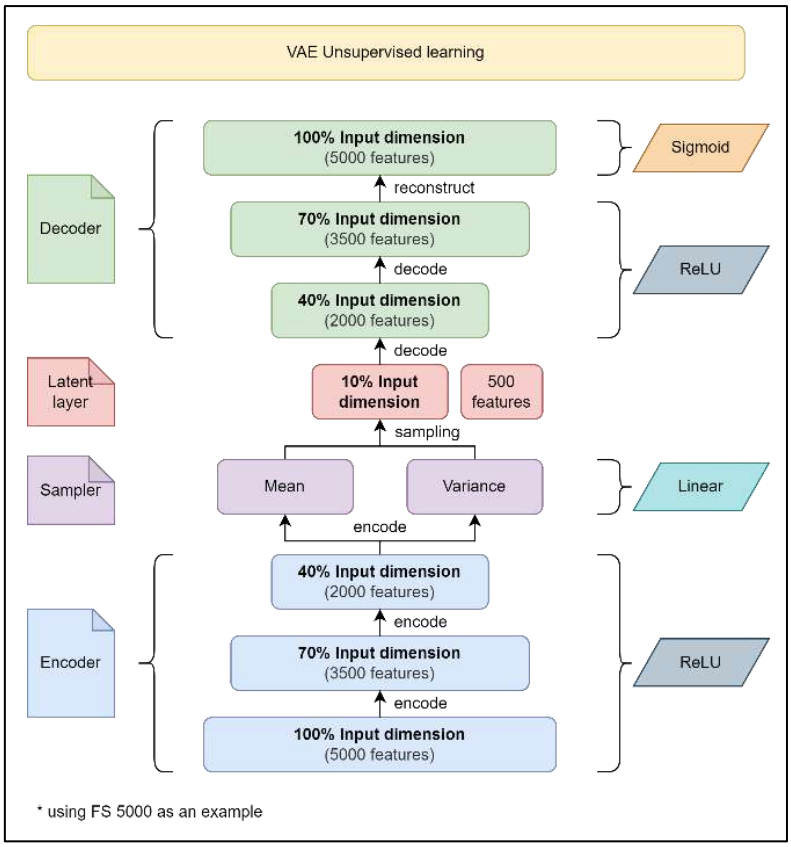

Figure 7a. The neural network structure of the VAE model for unsupervised training.

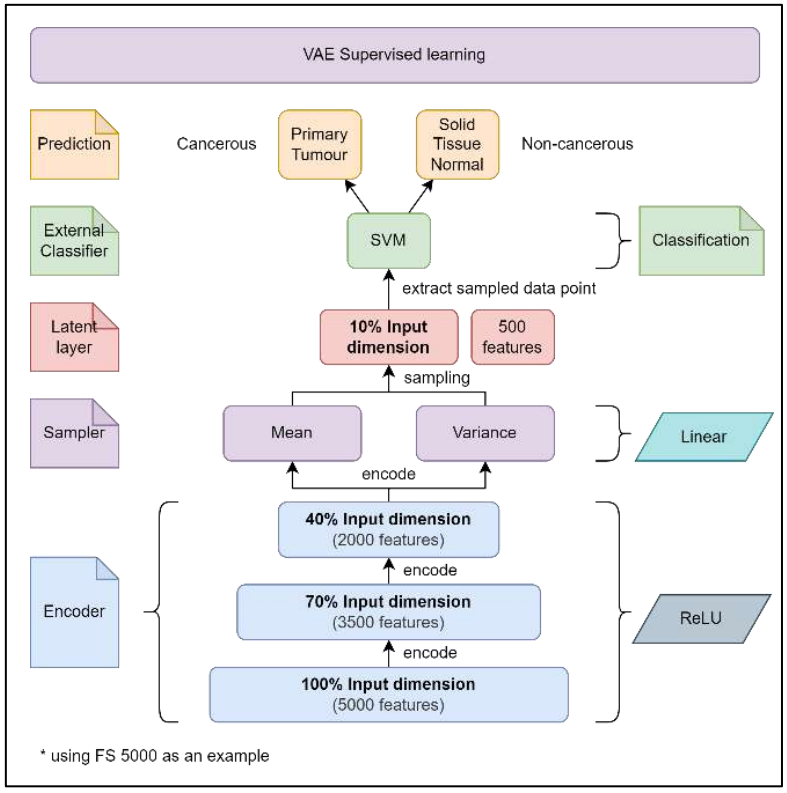

Figure 7b. The neural network structure of the VAE model for supervised training.

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 and loss functions. With reference to [29] in their work, the activation function for the sampler is set to "linear". Two different loss functions 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 loss functions [29].

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 a similar fine-tuning method in the supervised learning of the SDAE model. This is because 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 are 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

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 selecting 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 initial classification result on the 20 feature subsets of multi-omics data is shown in Figure 8. It is observed that from FS 20000 to 14000, the mean accuracy is recorded at 0.996. While the mean accuracy for FS 13000 to 1000 is recorded at 1.000.

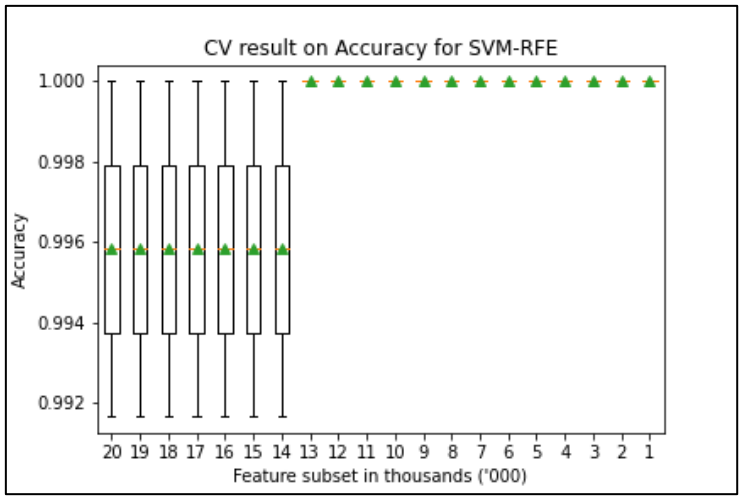

Figure 8. Boxplot for the CV result (accuracy) for each feature subset.

The omics composition for each feature subset is depicted in Figure 9. The general trend observed is that the composition of gene expression omics goes down (75.08% to 70.50%) as the size of the 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% fluctuates as the size of the feature subset goes down until it settles at 1.8%.

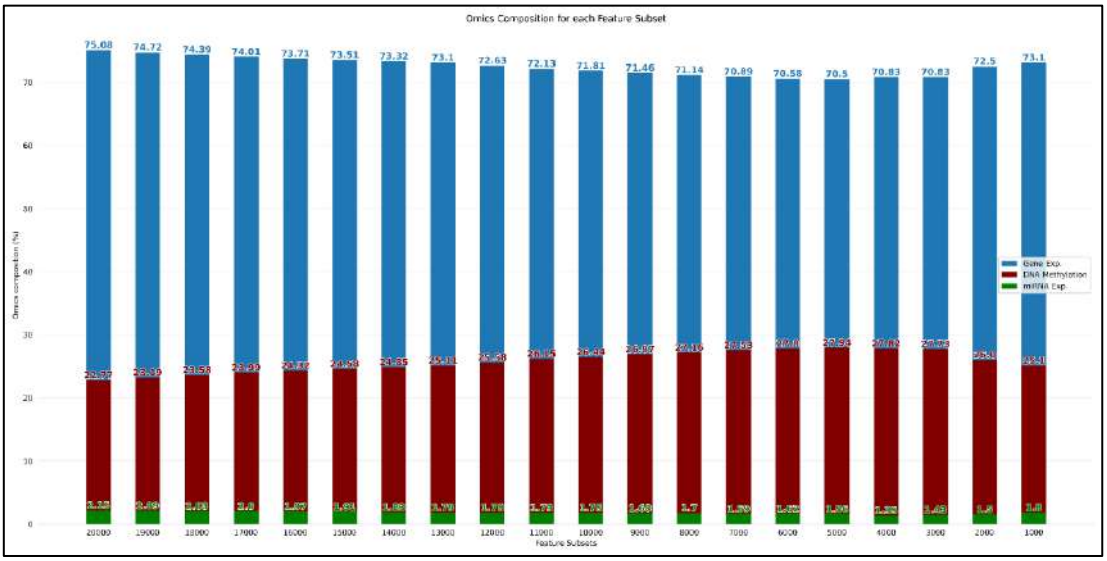

Figure 9. Omics composition after SVM-RFE represented in bar chart.

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 10. Generally, the model loss for each SDAE model is 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.

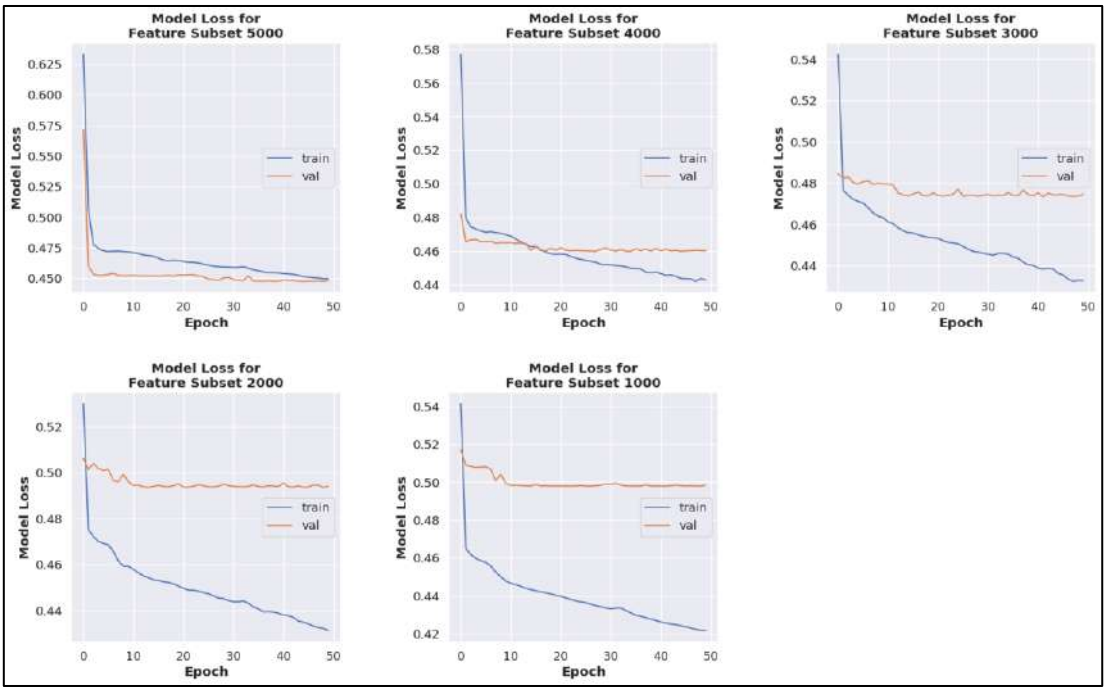

Figure 10. Model loss for the unsupervised learning for the SDAE model.

The classification results of the fine-tuning supervised learning SDAE models are recorded. Figure 11 shows the accuracy of the SDAE models with respect to epoch, while Figure 12 shows the confusion matrix obtained from the classification result of the last epoch. It is observed that FS 1000 and 4000 can correctly classify all the instances, while FS 2000, 3000, and 5000 cannot classify the negative class label correctly, resulting in one false positive. The confusion matrix tabulates the accuracy, AUC score, precision, recall, and F1 score in Table 12.

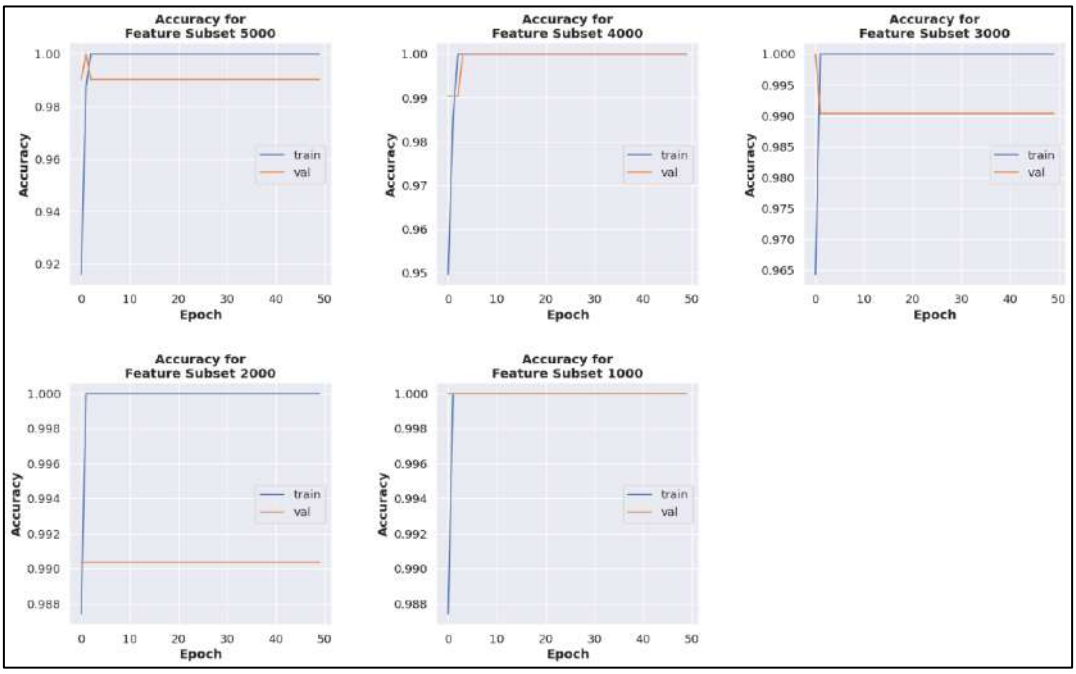

Figure 11. The classification result for the fine-tuned supervised learning SDAE model. The accuracy of supervised learning for the SDAE model.

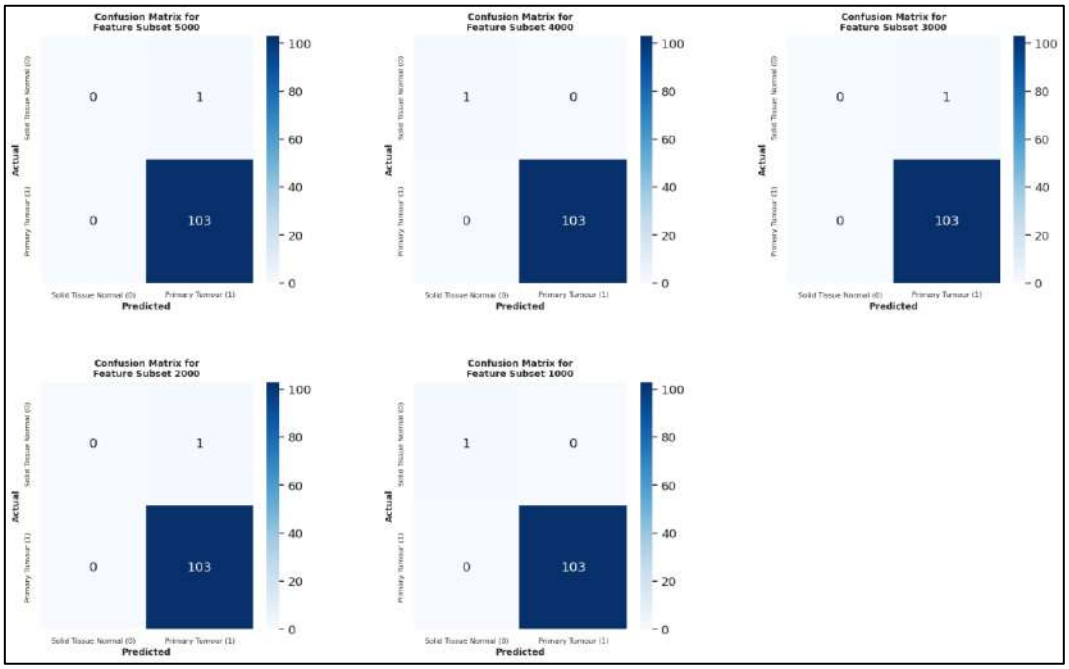

Figure 12. The confusion matrix from the classification using the fine-tuned SDAE model.

Table 12. Metrics obtained from the classification result of the 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 (Kullback-Leibler) KL loss, as shown in Figure 13. Generally, the total model loss for the VAE models decreases as the size of the feature subset used to develop the VAE models decreases. It is observed that both the total model loss for the training and validation sets can converge, but fluctuation is also observed for the validation sets.

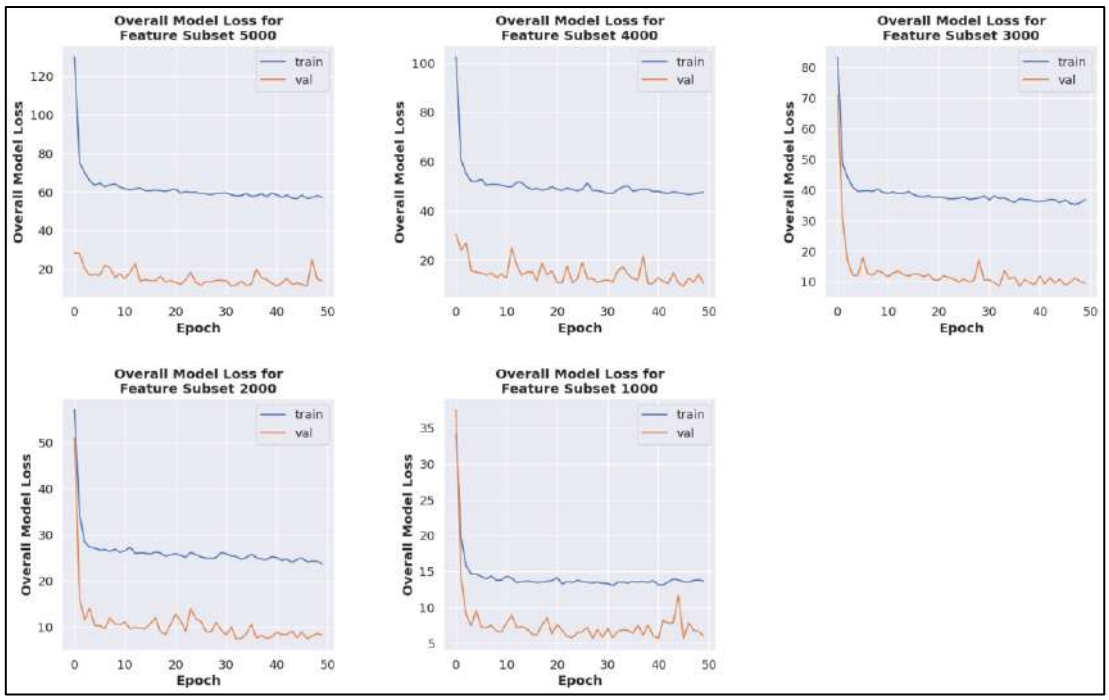

Figure 13. The overall loss of the VAE model during the unsupervised learning phase.

The classification of the VAE model involves using an external classifier, an SVM model. The study first built a supervised learning VAE model using the same method applied for fine-tuning 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 to 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 as a confusion matrix in Figure 14.

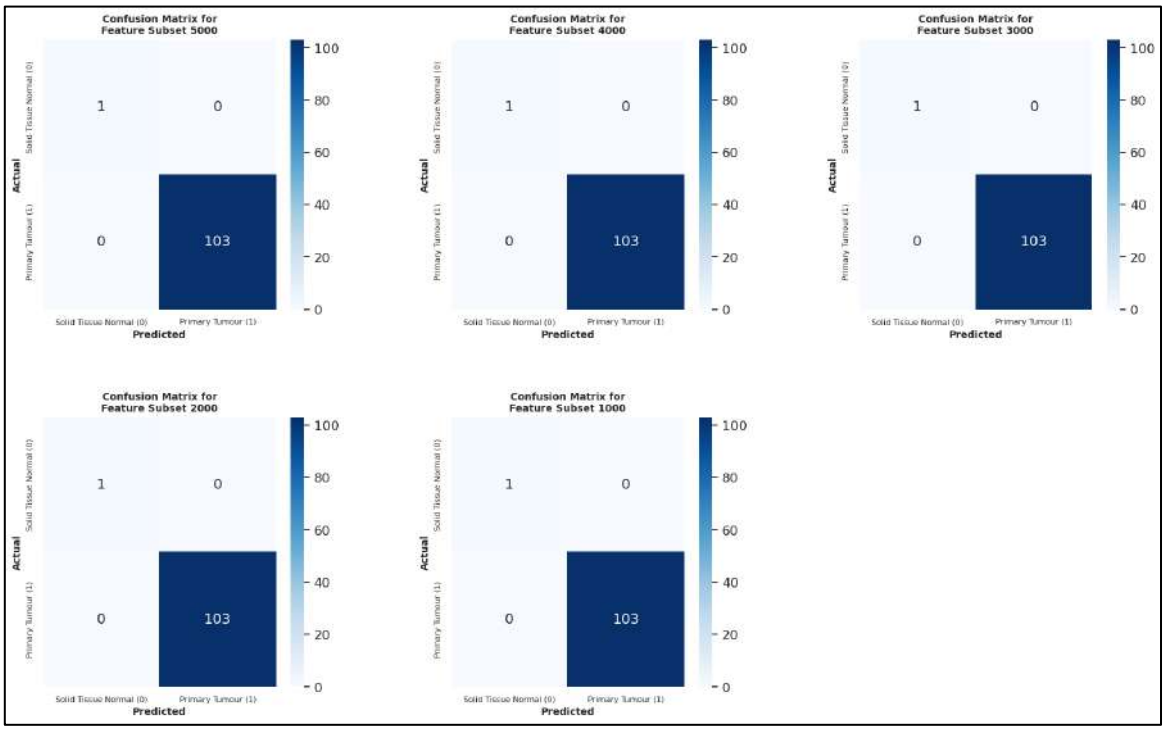

Figure 14. Confusion matrices were produced using SVM's classification results as the external classifier on the encoded inputs in the VAE model.

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 fine-tuned supervised learning VAE model's classification result and the SVM classification.

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 8, it could be understood that the SVM-RFE has removed many irrelevant features as the size of the feature subsets decreases. Therefore, the smaller feature subsets contain a 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 has removed a lot of irrelevant 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%. The testing set with 104 instances only has 1 example 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, only FS 1000 and 4000 can achieve 100% accuracy with all correctly classified instances. On the other hand, FS 2000, 3000, and 5000 failed to predict the negative class label accurately, 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 is to predict the minority negative class correctly. The AUC score is a more useful metric 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. Meanwhile, 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 can achieve 1.000 accuracies. 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 of learning the valuable feature of each feature subset during the encoding process. FS 1000 and 4000 are the two feature subsets that allowed both the SDAE and VAE models to achieve a score of 1.000 for each metric. This could indicate that FS 1000 and 4000 are the two most optimal feature subsets selected by the SVM-RFE algorithm.

5. Conclusions

The study has employed SVM-RFE for feature selection to extract the most relevant features from the multi-omics data with large dimensions. The output obtained from the SVM-RFE are the 20 most optimal feature subsets the algorithm selects. 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. The SDAE and VAE classifiers can correctly classify all the instances using the testing set.

The suggestions for future work for this study include 1) the use of new and updated multi-omics data to cater to the severe data imbalance problem and 2) the use of better hardware, including a GPU with larger memory capacity, to allow the development of the neural network of larger feature subsets (e.g., FS 20000) in deep learning classification.

Author Contributions: Conceptualization, methodology, AAS, NSA, JTL, HAM; literature search AAS, NSA, JTL, HAM; ZAS experiment, result analysis and validation, AAS, NSA, JTL, HAM; NHW writing—original draft preparation, AAS, NSA, JTL, CWH writing—review and editing, AAS, NSA; proofread, AAS, HAM, ZAS, CWH.

Funding: This work was funded by the Malaysian Ministry of Higher Education through the UTM Fundamental Research (Grant Number: Q.J130000.2551.21H71).

Acknowledgments: The authors would like to express gratitude to the Malaysian Ministry of Higher Education for the financial sponsorship of this study through the UTM Fundamental Research (Grant Number: Q.J130000.2551.21H71). The Faculty of Computing, Universiti Teknologi Malaysia also supports the study.

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

References

1. Ong YS and Tan LTH. Cancer, natural products and nanodrug delivery systems. Prog Microbes Mol Biol 2020; 3(1).
2. Ferlay J, Colombet M, Soerjomataram I, *et al*. Cancer statistics for the year 2020: An overview. Int J Cancer 2021; 149(4): 778-789.
3. Yadav SP. The wholeness in suffix -omics, -omes, and the word om. J Biomol Tech 2007; 18(5): 277.
4. Subramanian I, Verma S, Kumar S, *et al*. Multi-omics data integration, interpretation, and its application. Bioinform Biol Insights 2020; 14: 7-9.
5. Feldner-Busztin D, Firbas Nisantzis P, Edmunds SJ, *et al*. Dealing with dimensionality: the application of machine learning to multi-omics data. Bioinformatics 2023; 39(2).
6. El-Manzalawy Y, Hsieh T-Y, Shivakumar M, *et al*. Min-redundancy and max-relevance multi-view feature selection for predicting ovarian cancer survival using multi-omics data. BMC Med Genomics 2018; 11(Suppl 3): 71.
7. Taguchi YH and Turki T. Novel feature selection method via kernel tensor decomposition for improved multi-omics data analysis. BMC Med Genomics 2022; 15(1): 1-12.
8. Kang H. The prevention and handling of the missing data. Korean J Anesthesiol 2013; 64(5): 402-406.
9. Singh D and Singh B. Investigating the impact of data normalization on classification performance. Appl Soft Comput 2020; 97: 105524.

10. Kuhn M and Johnson K *Feature engineering and selection: A practical approach for predictive models*. 2019, New York: CRC Press. 1-297.
11. Gholamy A, Kreinovich V, and Kosheleva O. Why 70/30 or 80/20 relation between training and testing sets: A pedagogical explanation. Journal of Intelligent Technologies and Applied Statistics 2018; 11(2): 105-111.
12. Huang ML, Hung YH, Lee WM, *et al*. SVM-RFE based feature selection and taguchi parameters optimization for multiclass SVM Classifier. Sci World J 2014; 2014.
13. Alharbi F and Vakanski A. Machine learning methods for cancer classification using gene expression data: A review. Bioengineering 2023; 10(2): 173.
14. Azmi NS, Samah AA, Sirgunan V, *et al*. Comparative analysis of deep learning algorithm for cancer classification using multi-omics feature selection. Prog Microbes Mol Biol 2022; 5(1).
15. Hanczar B, Bourgeais V, and Zehraoui F. Assessment of deep learning and transfer learning for cancer prediction based on gene expression data. BMC Bioinformatics 2022; 23(1).
16. Ronen J, Hayat S, and Akalin A. Evaluation of colorectal cancer subtypes and cell lines using deep learning. Life Sci Alliance 2019; 2(6).
17. Stahlschmidt SR, Ulfenborg B, and Synnergren J. Multimodal deep learning for biomedical data fusion: a review. Brief Bioinform 2022; 23(2).
18. Harun MF, Samah AA, Shabuli MIA, *et al*. Incisor malocclusion using cut-out method and convolutional neural network. Prog Microbes Mol Biol 2022; 5(1).
19. Nasir SN, Ishak M, Sagap I, *et al*. Circular RNA-EPHB4 As A Potential Biotarget In Colorectal Cancer: A Preliminary Analysis. Prog Microbes Mol Biol 2022; 5(1).
20. Nazarie WFWM, Yusof AM, Tieng FYF, *et al*. Differential gene expression analysis of papillary thyroid carcinoma reveals important genes for lymph node metastasis. Prog Microbes Mol Biol 2022; 5(1): a0000269.
21. Chan PF and Hamid RA. An overview of breast cancer: Classification and related signaling pathways. Prog Microbes Mol Biol 2021; 4(1).
22. Ishak M, Baharudin R, Tan LT-H, *et al*. Landscape of HOXA genes methylation in colorectal cancer. Prog Microbes Mol Biol 2020; 3(1): a0000085.
23. Mutalib NSA, Ismail I, and Ser HL. Molecular profiling and detection methods of microRNA in cancer research. Prog Microbes Mol Biol 2020; 3(1).
24. Mohamad Yusof A, Tieng FYF, Muhammad R, *et al*. In-depth characterization of miRNome in papillary thyroid cancer with BRAF V600E mutation. Prog Microbes Mol Biol 2020; 3(1).
25. Keskar NS, Nocedal J, Tang PTP, *et al*. On large-batch training for deep learning: Generalization gap and sharp minima. 5th International Conference on Learning Representations, ICLR 2017 - Conference Track Proceedings 2016.
26. Hira MT, Razzaque MA, Angione C, *et al*. Integrated multi-omics analysis of ovarian cancer using variational autoencoders. Sci Rep 2021; 11(1).
27. Gupta P and Malhi AK. Using deep learning to enhance head and neck cancer diagnosis and classification. IEEE International Conference on System, Computation, Automation and Networking, ICSCA 2018: 1-6.
28. Yendapalli V, Ruby AU, Theerthagiri P, *et al*. Binary cross entropy with deep learning technique for Image classification A.Usha Ruby et al Binary cross entropy with deep learning technique for Image classification. Int J Adv Trends Comput Sci Eng 2020; 9(4): 5393-5397.
29. Gondara L. Medical image denoising using convolutional denoising autoencoders. IEEE International Conference on Data Mining Workshops, ICDMW 2016: 241-246.

30. Simidjievski N, Bodnar C, Tariq I, *et al*. Variational autoencoders for cancer data integration: Design principles and computational practice. Front Genet 2019; 10.