Enhancing Cancer Detection Using Machine Learning Techniques

Isabella Chong, Lee Sin Chee, Yang Liting, Yu Zifan

National University of Singapore
21 Lower Kent Ridge Rd
Singapore 119077

1. Introduction

Cancer is one of the most common causes of death globally. (Roser, Ritchie 2024) Screening and early detection are critical for effective treatment and can greatly increase the survival rates of nearly all cancers. (Crosby et al. 2022)

Currently, numerous methods are used to classify between healthy and cancer patients, such as Gradient Boosting and Support Vector Machine (SVM). A recent study has shown that DNA from cancer patients had variations in fragment lengths at different regions compared to healthy DNA. (Cristiano et al. 2019) Several works have successfully used machine learning models, such as Gradient Boosting (Cristiano et al. 2019) and Support Vector Machine, (Liu, Chen and Wong 2021) for cancer detection. However, most of the models used are only for binary classification between cancer and healthy individuals.

In this paper, we prioritized classifying between healthy and screening stage, and healthy and early stage cancer. We implemented K-nearest Neighbors (KNN), Artificial Neural Network (ANN), Decision Tree and Random-Forest to perform multi-class classification, rather than binary classification used in present works. After this, models are compared on their effectiveness in differentiating between healthy and screening-stage cancer and differentiating between healthy and early stage cancer.

2. Dataset and Data Processing

In this dataset, there are 350 columns denoting the maximum normalized frequency of DNA fragment lengths and the last column indicates the cancer stage (healthy/screening stage cancer/early stage cancer/mid stage cancer/late stage cancer).

We preprocessed the data by checking for missing data and converting the cancer stages to numeric classes. We also found out that the dataset is class imbalanced as there are only 60 healthy observations, while there are more than 400 observations for each of the other classes. This class imbalance problem may cause models to be biased towards predicting the majority class. (Hvilshøj 2022).

To address this issue, we performed the Synthetic Minority Over-sampling Technique (SMOTE). SMOTE creates synthetic examples to over-sample the minority class, healthy observations. (Chawla et al. 2002) Since studies have suggested that a combination of SMOTE and under-sampling performs better, (Chawla et al. 2002) we combined both SMOTE and random under-sampling and applied it to our dataset.

Given that there is a large number of features, we decided to conduct feature selection on our dataset to choose the more significant features to train our models. To do so, we plotted a graph of the maximum normalized frequency of DNA fragment lengths against the feature number of the 1st sample for each of the raw, (Figure 1) normalized and standardized data. Comparing the 3 graphs, standardizing the data maximized the difference in profiles between healthy, screening and early stage cancer. We then retained the features which showed a great difference in standardized maximum normalized frequency between the 3 stages to be used for our models, as shown in Figure 2 below. The feature regions used for standardized data are the ranges 0-115, 125-230, and 265-350. Similarly, we extracted features from the normalized data based on the visualized differences between the 3 stages.

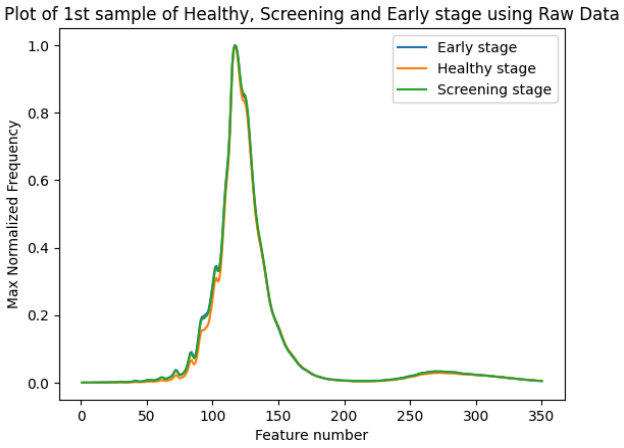

Figure 1: Plot of maximum normalized frequency against feature number using raw data for healthy, screening stage and early stage cancer.

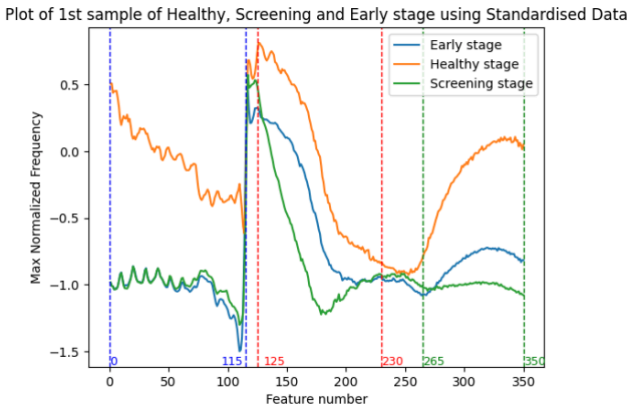

Figure 2: Plot of maximum normalized frequency against feature number using standardized data for healthy, screening stage and early stage cancer.

3. Methods

We did an 80/20 split of the raw dataset from 'Train_Set.csv' into training and validation data. The same was also done for the normalized and standardized data after we conducted feature selection. We then used the training set for the raw, normalized and standardized data to train each of our models before testing our models on the validation dataset.

The models we explored include K-Nearest Neighbours (KNN), Artificial Neural Networks (ANN), Decision Trees and Random-Forest. We trained the models and tuned the hyperparameters using code outlined by Kenjee (Kenjee. 2023). Grid search is used for KNN, Decision Tree and Random-Forest.

Grid Search systematically explores all hyperparameter combinations using a brute-force approach, evaluating each via cross-validation to select the combination with the highest accuracy.

3.1 K-Nearest Neighbours (KNN)

K-Nearest Neighbours (KNN) is implemented in cancer classification (Wu, J., and Hicks, C. 2021). Being a non-parametric, lazy learning algorithm and robust to noise (10 base pair periodicity), KNN is easier to implement and provides us with preliminary evaluations of model accuracy. We employed Grid Search to optimize the hyperparameters[1] for our KNN classifier and test it on all types of data to determine which data gives the best performance metrics.

[1] Hyperparameters of all models can be found in the appendix.

3.2 Artificial Neural Network (ANN)

Artificial Neural Network (ANN) is a machine learning technique commonly used in healthcare for clinical diagnosis and cancer prediction. (Shahid, Rappon and Berta 2019) ANNs can learn non-linear relationships, which is appropriate for our task due to our high dimensional dataset so we decided to implement ANN. (Abiodun et al. 2018) We also experimented with the hyperparameters of the models on the raw, normalized, standardized and standardized data without feature selection to ascertain which model has the best performance.

3.3 Decision Tree

Since the data is unbalanced and non-parametric, we considered the use of a decision tree. Decision tree is a hierarchical structure resembling a flowchart, where each node represents a decision based on the value of a feature, leading to possible outcomes at the leaf nodes. They are constructed through recursive partitioning, where features are selected to split the dataset into homogenous subsets. This process continues until a stopping criterion is met, resulting in a tree structure that facilitates decision-making.

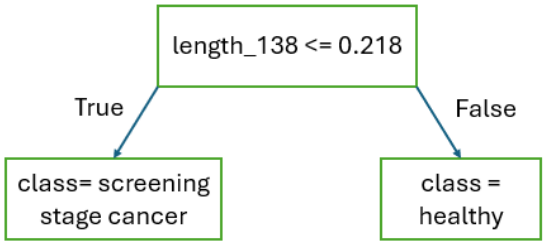

Figure 3: Example Decision Tree for Illustration

In Figure 3, this is an example decision tree with depth 1. The root node checks if the frequency of length_138 is <= 0.218 and goes to the leaf nodes. If it is true, the class is a screening stage, else it is healthy. However, trees generally tend to overfit and provide poor performance on test data. Thus, we also considered an ensemble method, the random forest.

3.4 Random-Forest

A Random-Forest is a meta estimator that fits a number of decision tree classifiers on many sub-samples of the dataset and uses averaging to improve the predictive accuracy and control over-fitting. (Scikit-learn 2024) The features in a Random-Forest are randomly selected in each decision split. The random feature selection helps to reduce the correlation between trees, which improves prediction power and efficiency. (Breiman 2001)

Past study on cancer classification research revealed that Random-Forest performed well for multiple evaluation metrics. It can also perform well on large data sets and

high dimensional data modeling, which is suitable for our task. (Ali et al., 2012)

4. Results and Discussions

To evaluate our models, the True Positive Rate (TPR) of class 2 and 3 from the classification report and the True Negative Rate (TNR) for class 0 (healthy) are suitable evaluation matrices instead of accuracy. TPR is chosen as we want to reduce the cases where people with screening and early stage cancer are incorrectly predicted as healthy, so False Negatives (FN) should be minimized to ensure patients get early treatment. Likewise, False Positives (FP) are costly when predicting for class 0 as it could result in inefficient use of medical facilities if healthy individuals are predicted to have cancer and are required to have medical attention. Hence, to minimize FP, TNR is used to evaluate the models' performance for class 0. The model with the best TPR for class 2 (screening stage) and class 3 (early stage) and TNR for class 0 will be used to classify the test data from 'Test_Set.csv'. To simplify the multiple performance metrics, we decided to use a summary score[2], which is computed by the formula defined below:

$$TPR = \frac{TP}{TP + FN}$$

$$TNR = \frac{TN}{TN + FP}$$

$$Summary = \frac{4}{\frac{1}{TPR_{2,0}} + \frac{1}{TNR_{2,0}} + \frac{1}{TPR_{3,0}} + \frac{1}{TNR_{3,0}}}$$

We obtained this formula for the summary score after drawing inspiration from the F1-score and applied it by taking the harmonic mean of all 4 TPR and TNR scores. The rationale behind this approach lies in the fact that only when both TNR and TPR are high will the sum of reciprocals be small, therefore yielding a higher summary score.

4.1 Training Process for Models

We determined the best dataset to train each model following the process below, with Random-Forest (our final model) as an example.

4.1.1 Base Random-Forest on Raw Data

When our base model was trained with raw data, we obtained a summary score of 0.33302. We decided to improve the model by addressing the class imbalance problem.

[2] The subscript in the formula means (positive class, negative class), which are used in computing the summary score.

4.1.2 Base Random-Forest on Resampled Data

Building on our base model, we applied a combination of SMOTE and random under-sampling on our training data and trained the model with this resampled data. The model then attained a summary score of 0.92066, showing significant improvement from the base model.

4.1.3 Base Random-Forest on Normalized Data

The Random-Forest model is then trained again using the resampled normalized data, whose visualization illustrated a greater difference in profiles of the various stages. The resulting summary score was 0.92066 when tested on the validation data.

4.1.4 Tuned Random-Forest on Resampled Data

To further improve on this model, we tuned its hyper-parameters using Grid Search. We implemented our scoring system of the summary score to find the best hyper-parameters for the model. After tuning, the summary score increased to 0.93987.

As we built on our initial base model and improved it, we saw a notable increase in performance. We summarized the model's performance after each refinement in the table as follows:

Model	Summary Score
Base Random-Forest (Raw)	0.33302
Base Random-Forest (Resampled)	0.92066
Base Random-Forest (Normalized)	0.92066
Tuned Random-Forest (Normalized)	0.93987

Table 1: Performance of Random-Forest models after each improvement

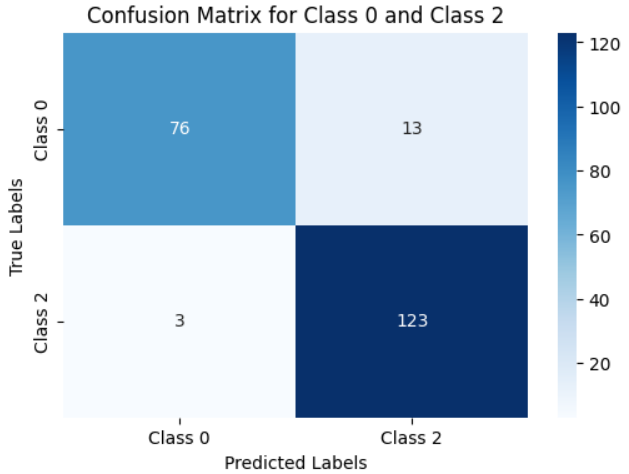

Figure 4: Confusion matrix of tuned Random-Forest (Normalized) for class 0 and class 2 only

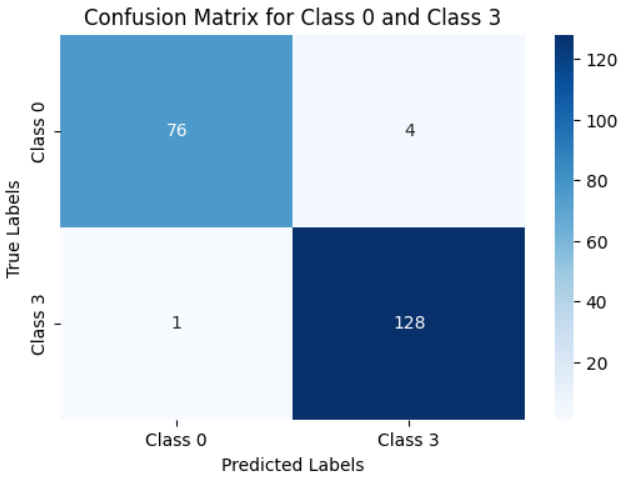

Figure 5: Confusion matrix of tuned Random-Forest (Normalized) for class 0 and class 3 only

From Table 1 above, the model trained on resampled normalized data with its hyper-parameters tuned performed the best with a score of 0.93987 after all the improvements made. Therefore, this model was selected to compare with the other 3 models, namely KNN, ANN and Decision Tree.

4.2 Comparison of Models

For each of our models, we followed the process outlined above to determine the best dataset used to train each model. We then evaluated the models based on their summary scores and summarized each model's best performance in the table as follows:

Model	Summary Score
KNN (Normalized)	0.88082
ANN (Standardized)	0.74886
Decision Tree (Standardized)	0.86691
Random Forest (Normalized)	0.93987

Table 2: Comparison of models tested on validation data

The Random-Forest model trained using normalized data has the highest summary score. Thus, Random-Forest was chosen as our best model to test on the test data.

4.2.1 Test Model on Raw Test Data

Firstly, we tested the Random-Forest model on the raw test data to provide a basis of comparison with the normalized test data, which is our theoretical best processed data. When tested on the raw test data, the Random-Forest model obtained a summary score of 0.64483.

4.2.2 Test Model on Normalized Test Data

Afterwards, we tested the model on the normalized test data and the Random-Forest model obtained a summary score of 0.6896, a 6.94% increase compared to raw. Since the summary score on test data is lower than the validation, we tried to further improve our model by addressing the noise that is present in the dataset, especially at the regions with sharp edges. We looked into smoothing to account for the noise that is present in the dataset.

4.2.3 Test Model using LOESS Data

Due to the noise observed in the curve in Figures 1 and 2, we considered smoothing the curve using Locally Estimated Scatterplot Smoothing (LOESS) also known as local regression then applying it onto the data. LOESS applies locally weighted polynomial regression on subsets of points such that the further away from the point, the less weight it has.

	Raw	Normalized	Loess
Summary Score	0.6448	0.6896	0.6826
$TPR_{2,0}$	0.4963	0.5294	0.5938
$TNR_{2,0}$	0.7308	0.7586	0.6923
$TPR_{3,0}$	0.9234	0.9147	0.9390
$TNR_{3,0}$	0.5758	0.6667	0.6000

Table 3: Summary of evaluation metrics

When we tested this transformation, the model obtained a summary score of 0.6826, suggesting that smoothing did not improve the model's performance. Upon closer inspection of Table 3 above, the model performs better on normalized dataset in terms of summary score, $TNR_{2,0}$, and $TNR_{3,0}$ but worse than Loess for $TPR_{2,0}$ and $TPR_{3,0}$. Thus, the usage of Random-Forest on the type of data depends on which metric we want to maximize.

The poor performance on test data could be attributed to our feature selection process, as the important features were selected solely based on the 1st sample of every class, which may not be representative of the entire dataset. Additionally, the data used to train our model contains DNA fragment frequencies for the same person at different time periods, which may have resulted in overfitting of the model. Thus, heavily affecting the performance of our model on the test data.

5. Conclusion and Future Work

In conclusion, Random-Forest trained using normalized resampled data, with tuned hyper-parameters and without LOESS smoother gives the best result in maximizing the summary score. More samples from each class can be used for visualization before feature extraction, generalizing the pattern of the DNA fragment frequency to possibly improve the model's performance on the test data.

References

Abiodun, O. I., Jantan, A., Omolara, A. E., Dada, K. V., Mohamed, N. A., and Arshad, H. 2018. *State-of-the-art in Artificial Neural Network applications: A survey.* Heliyon, 4(11), e00938. https://doi.org/10.1016/j.heliyon.2018.e00938

Alharbi, F., and Vakanski, A. 2023. *Machine learning methods for cancer classification using Gene Expression Data: A Review*. Bioengineering (Basel, Switzerland). https://www.ncbi.nlm.nih.gov/pmc/articles/PMC9952758/

Ali, Jehad and Khan, Rehanullah & Ahmad, Nasir & Maqsood, Imran. 2012. *Random Forests and Decision Trees.* International Journal of Computer Science Issues(IJCSI). 9.

Breiman, L. *Random Forests. Machine Learning* 45, 5–32 2001. https://doi.org/10.1023/A:1010933404324

Chawla, N. V., Bowyer, K. W., Hall, L. O., and Kegelmeyer, W. P. 2002. *SMOTE: Synthetic Minority Over-sampling Technique*. Journal of Artificial Intelligence Research, 16, 321–357. https://doi.org/10.1613/jair.953

Cristiano, S., Leal, A., Phallen, J., Fiksel, J., Adleff, V., Bruhm, D. C., Jensen, S. Ø., Medina, J. E., Hruban, C., White, J. R., Palsgrove, D. N., Niknafs, N., Anagnostou, V., Forde, P., Naidoo, J., Marrone, K., Brahmer, J., Woodward, B. D., Husain, H., … Velculescu, V. E. 2019. *Genome-wide cell-free DNA fragmentation in patients with cancer.* Nature, 570(7761), 385–389. https://doi.org/10.1038/s41586-019-1272-6

Crosby, D., Bhatia, S., Brindle, K. M., Coussens, L. M., Dive, C., Emberton, M., Esener, S., Fitzgerald, R. C., Gambhir, S. S., Kuhn, P., Rebbeck, T. R., and Balasubramanian, S. 2022. *Early detection of cancer.* Science, 375(6586). https://doi.org/10.1126/science.aay9040

Hvilshøj, F. 2022. Introduction to balanced and imbalanced datasets in Machine Learning. Balanced and Imbalanced Datasets in Machine Learning [Full Introduction]. https://encord.com/blog/an-introduction-to-balanced-and-imbalanced-datasets-in-machine-learning. Accessed: 2024-03-27

Kenjee. 2023. *Titanic project example*. Kaggle. https://www.kaggle.com/code/kenjee/titanic-project-example

Liu, L., Chen, X., and Wong, K.-C. 2021. *Early cancer detection from genome-wide cell-free DNA fragmentation via shuffled frog leaping algorithm and support vector machine.* Bioinformatics, 37(19), 3099–3105. https://doi.org/10.1093/bioinformatics/btab236

Roser, M., Ritchie, H. 2024. Cancer. Our World in Data. https://ourworldindata.org/cancer. Accessed: 2024-03-24

Shahid, N., Rappon, T., and Berta, W. 2019. *Applications of Artificial Neural Networks in health care organizational decision-making: A scoping review.* PLOS ONE, 14(2), e0212356. https://doi.org/10.1371/journal.pone.0212356

Sklearn.ensemble.randomforestclassifier. scikit. n.d.. https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.RandomForestClassifier.html. Accessed: 2024-04-04

Wu, J., and Hicks, C. 2021. *Breast cancer type classification using machine learning*. Journal of personalized medicine. https://www.ncbi.nlm.nih.gov/pmc/articles/PMC7909418/. Accessed: 2024-04-04

Appendix

Visualization using Normalized Data

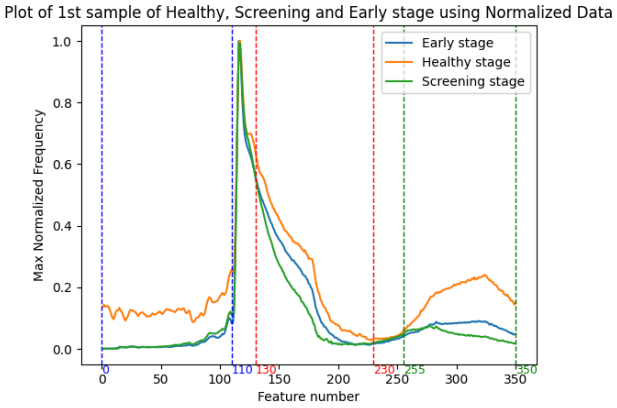

Figure 6: Plot of maximum normalized frequency against feature number using normalized data for healthy, screening stage and early stage cancer.

Hyperparameters of KNN (Normalized Data)

- algorithm = 'auto'
- n_neighbors = '3'
- p = 1
- weights = 'distance'

Hyperparameters of ANN (Standardized Data)

Layer (type)	Output Shape	Param #
dense_84 (Dense)	(None, 230)	80730
dense_85 (Dense)	(None, 160)	36960
dense_86 (Dense)	(None, 100)	16100
dense_87 (Dense)	(None, 50)	5050
dense_88 (Dense)	(None, 5)	255

Total params: 139095 (543.34 KB)
Trainable params: 139095 (543.34 KB)
Non-trainable params: 0 (0.00 Byte)

Table 4: Summary of ANN model trained with standardized data

- First 4 layers: activation = 'relu'
- Output layer: activation = 'softmax'
- loss = 'sparse_categorical_crossentropy'
- optimizer = 'sgd'
- metrics = ['accuracy']
- epochs = 65

Hyperparameters of Decision Tree (Standardized Data)

- criterion: 'gini'
- max_depth: 9
- max_leaf_nodes: 25
- min_samples_split: 2

Hyperparameters of Random Forest (Normalised Data)

- n_estimators=150
- criterion= 'entropy'
- bootstrap= False
- max_depth= 20
- max_features= 'sqrt'
- min_samples_leaf= 2
- min_samples_split= 4

Visualization of Test Data

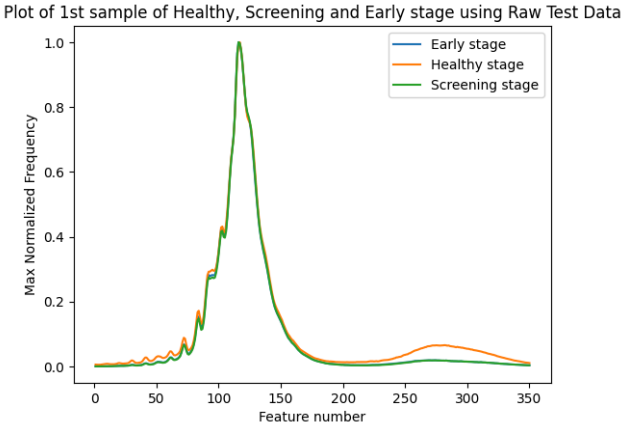

Figure 7: Plot of maximum normalized frequency against feature number using raw test data for healthy, screening stage and early stage cancer.

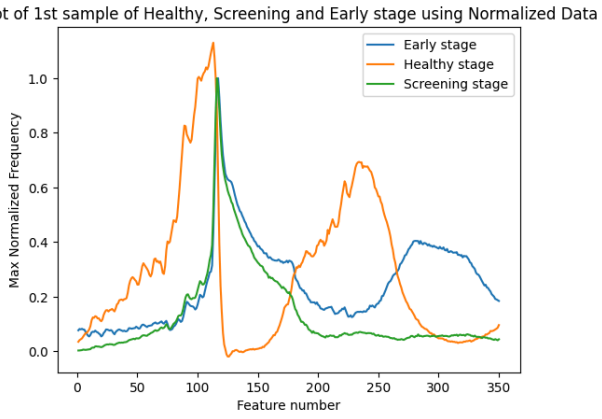

Figure 8: Plot of maximum normalized frequency against feature number using normalized test data for healthy, screening stage and early stage cancer.