Predicting Breast Cancer Survival

Elisha Martis - 20000170, Ferris Nowlan - 20005582, Bronwyn Rowland - 20052438,
Ben Wiebe - 20010837

Abstract—The ability to accurately classify breast cancer patients into risk classes is an integral component of individualised treatment plans. Predicting the likely outcome of a cancer pathology using gene expression data has been a goal in cancer research for several decades, but improvements in machine learning techniques can continually be applied to old data sets to glean new information and improve prediction accuracy. We investigate the use of generative adversarial networks to predict patient survival from a famous breast cancer microarray data set, with the aim of discerning previously unrecognized relationships in the data.

Index Terms—11, Life Sciences, Breast cancer classification, Deep Learning, Application project

1 Introduction

Globally, breast cancer is the most commonly diagnosed cancer in the world, with 2.1 million cases diagnosed each year according to the World Health Organization [1]. However, this statistic can be misleading, because breast cancer is not one monolithic disease. Cancer is caused by mutations that dysregulate cell growth, which can be caused by many different combinations of mutations. Not all mutations that can cause breast cancer are equally dangerous. However, when a physician first diagnoses an unusual growth, it is hard to accurately determine the risk level. Tumours often look similar even when they are driven by totally different molecular mechanisms, so the patient may be prescribed a treatment that does not match the true severity of their disease. Improving our ability to predict of the likely course of breast cancer upon diagnosis is an important goal because it allows the physician to personalise treatment. Improved prediction of prognosis could reduce mortality among breast cancer patients, prevent patients suffering from unnecessary treatments, improve use of hospital resources, and contribute to a better understanding of the genetic factors that cause breast cancer development [2].

To address this concern, a generative adversarial network (GAN) was trained on a large 2002 breast cancer data set from the Netherlands Cancer Institute (NKI) that is considered one of the gold standard data sets in the field [3]. It has been analyzed before, but not with newer deep learning techniques like GAN. Our goal was to increase the accuracy of survival prediction over the methods that had already been tried, which included k-nearest neighbours, hierarchical clustering, Support Vector Machine, Random Forest, Multilayer Perceptrons, and Genetic Programming [4]. Of those, Genetic Programming had the highest accuracy with an average classification accuracy of 67.2% on the binary prediction problem of whether the patient survived for 10.3 years [4]. We attempted an eight category prediction, where we predicted the length of survival in patients in either living and deceased categories by the end of the study. Pre-processing was carried out in R, and the pipeline was created using Python. Using GAN, we achieved a prediction accuracy of 32.1%. Better prediction accuracy could likely be attained by either using the most important features in a random forest, or by reducing the complexity to a binary category prediction problem.

- *Elisha Martis, Ferris Nowlan, Bronwyn Rowland, and Ben Wiebe are with School of Computing at Queen's University*
E-mail: 15emm3@queensu.ca, 15fn4@queensu.ca, 16ber3@queensu.ca, b.wiebe@queensu.ca

2 Related Work

In 2013, a group of researchers analysed a dataset of breast cancer patients from the Iranian Centre for Breast Cancer with the goal of attempting to predict cancer recurrence within two years [5]. The dataset used included numerous features about patient histories and tumour information. Three predictors were tested in the study: decision tree, support vector machine (SVM), and feed-forward neural network. Of these, the SVM predictor achieved the highest accuracy of 95.7%.

A research paper published in 2019 investigated the prediction of breast cancer using supervised machine learning techniques: Logistic Regression, Support Vector Machines (SVM) and K-nearest neighbor (KNN) [6]. The paper analyzed a dataset from the UCI Machine Learning repository. A major finding was that support vector machines was the best classification technique, with prediction accuracy of 92.7% [6]. Another major finding from this research paper was that carrying out machine learning classification prior to diagnosis can enhance the survival rate by identifying the disease in early stages [6]. This allows patients to have a better chance for survival by taking the clinical treatment at the right time. For example, patients classified as having benign cancers may avoid taking needless treatments. The contributions to theory and practice would appear to be helping in prediction of early stage detection, rescuing people's lives, minimizing the cost of medicines, and aid people's health [6].

A paper published in 2020 investigated the importance of various variables relating to breast cancer survival over different time periods of disease progression [7]. GA and LASSO models were used for variable selection and RUS

and SMOTE were used for resampling procedures. To predict breast cancer survival at 1, 5, and 10 years, ANN and LR models were employed. It was found that different variables lost or gained importance over time, and that not all factors commonly considered relevant to cancer treatment equally contribute to survival.

3 Dataset

3.1 Dataset Description

The Netherlands Cancer Institute (NKI) data set contained information on 272 patients with breast cancer. All chosen patients were relatively young, generally between 40 and 50 years old. The dataset included features such as number of years of survival after their first treatment date, the treatment administered, their status at the end of the study, and a microarray dataset of tumour gene expression [8]. Microarray data shows the expression of each gene in tumour cells relative to the expression of the same gene in a normal healthy breast cells. Each patient's microarray dataset included 1500 gene expressions, selected from the 25,000 tested gene arrays because they were significantly different from normal gene expression in at least three tumour samples [9].

Notably, length of survival after first treatment was a complicated feature because it could represent either survival until the patient's passing or, if they were alive at the end of the study, simply survival until their most recent check-up. Clearly the biological meaning of survival time is quite different between these two cases. This sort of bias in survival analysis is known as right-censoring, where not all patient's fates are known by the end of the study.

3.2 Dataset Exploration

To gain an initial understanding of the nature of the dataset and the relationship between features, a heatmap correlation matrix was generated to provide a visual that is easy to interpret. It was found that genes were most often correlated to each other rather than to survival (Fig. 1). This may suggest that redundant variables exist in the dataset, which has the potential to hinder classification. Nevertheless, several gene expressions were found to be moderately correlated with survival including Contig51749_RC, Contig48328_RC, NM_020974, esr1, NM_000125, Contig56390_RC, NM_014585, Contig53307_RC, and NM_005375 (1). Overall, the correlation matrix demonstrates that a nuanced relationship exists between a patient's genes and their survival.

To explore the target feature (breast cancer survival), a plot of the survival density distributions for both the alive and deceased patients was created (Fig. 2). The two distributions were positively-skewed in a similar manner with an overall skewness of about 0.47. The data was not transformed as this degree of skewness is not likely to have a large adverse effect on model performance. Survival in patients who passed away before the end of the study peaked around two and a half years after diagnosis. On the other hand, survival in patients who were alive at the end of the study is shown to progress well into 20 years after diagnosis.

In addition to the correlation matrix and the density distribution, general statistics were performed on the dataset, such as calculating min, max, mean, and quartiles, as well as searching for any missing values. This was to identify any outliers or mistakes in data collection.

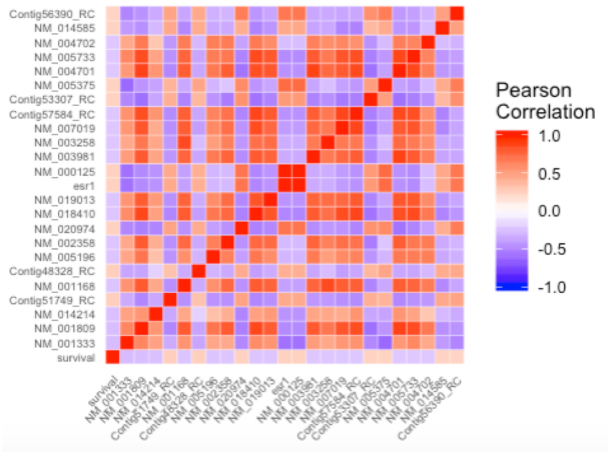

Fig. 1. Correlation matrix of genes most highly correlated with breast cancer survival.

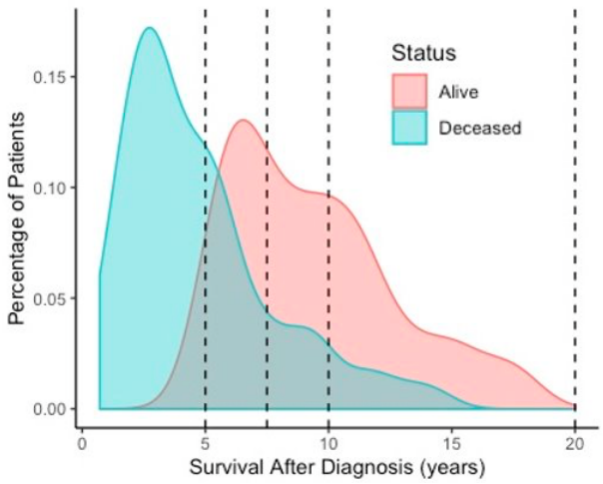

Fig. 2. Distribution of the length of survival for alive and deceased breast cancer patients.

4 Methodology

4.1 Dataset Preprocessing

Pre-processing was performed to prepare the data for modeling. The previously calculated quartiles were used to bin patient survival into 8 buckets, as shown in Table 1. Bins numbers 1 to 4 correspond to the four quartiles of the deceased patients, and bins numbers 5 to 8 correspond to the four quartiles of the alive patients. This was done to account for the right-censoring of the target feature. Overall, patient survival was binned to smooth the data and reduce noise.

As another aspect of preprocessing, features that were not deemed relevant to prediction were removed. For example, "patient ID" number was removed, as well as "time

Bin Number	Health Status At End of Study	Survival (years)
1	Deceased	0-5
2	Deceased	5-7.5
3	Deceased	7.5-10
4	Deceased	10-20
5	Alive	0-5
6	Alive	5-7.5
7	Alive	7.5-10
8	Alive	10-20

TABLE 1
Binned breast cancer survival.

to recurrence" as it was the same as survival except the measurement of time started at the first day of treatment rather than the day of diagnosis. Finally, the features were normalized using the `MinMaxScaler` from the `scikit-learn` library for Python. This was to bring all features into a common range and prevent systematic bias when training the models.

4.2 Feature Importance

Random Forest was used to determine the most predictive gene expressions in the dataset. The `scikit-learn` library for Python was used to build, train, and use this model to classify the data according to the quartile classifcation of breast cancer survival. To optimize the random forest model, random search was carried out using the `RandomizedSearchCV`. To find the best hyper-parameters for the model, forward selection was used, where the model is created with three parameters and one parameter is added at a time. Recall was employed to determine whether the model's performance improved with each newly added parameter. The best hyper-parameters for the random forest model, which produced a recall of 47.7%, were:

- bootstrap: False
- criterion: entropy
- min impurity split: 1e-07
- min samples leaf: 5
- criterion: min samples split: 16

The Random Forest provided a measure of impurity decrease for each incorporated feature, which can be used to identify the important features in the dataset and help us to better understand the drivers behind the model. The feature importance attribute of Random Forests gives an indication as to which features the model considers most predictive. The top 20 of these features for identifying patient survival can be seen in the Fig. 3. The figure shows that NM_003430 is considered by the model to be the most important expressed gene in breast cancer. Feature importance can be useful in feature selection by removing the low importance features from the model.

The confusion matrix shown in Fig. 4 indicates that the accuracy of the Random Forest model is approximately 36%.

	feature	importance
340	NM_003430	0.008458
449	AL117418	0.008094
1100	NM_016359	0.007373
973	Contig23211_RC	0.007019
1369	NM_001168	0.005831
904	NM_005733	0.005479
207	NM_003258	0.005311
1096	Contig55725_RC	0.005240
1442	NM_001333	0.005034
451	AB040926	0.004874
1234	Contig46452_RC	0.004767
1519	Contig58301_RC	0.004661
163	Contig37562_RC	0.004658
842	NM_014398	0.004585
1129	AL137566	0.004509
575	NM_004456	0.004467
1235	U79293	0.004147
1126	AL049337	0.004136
732	Contig35629_RC	0.003986
457	NM_021069	0.003920

Fig. 3. Top 20 significant features for predicting the correct survival bin.

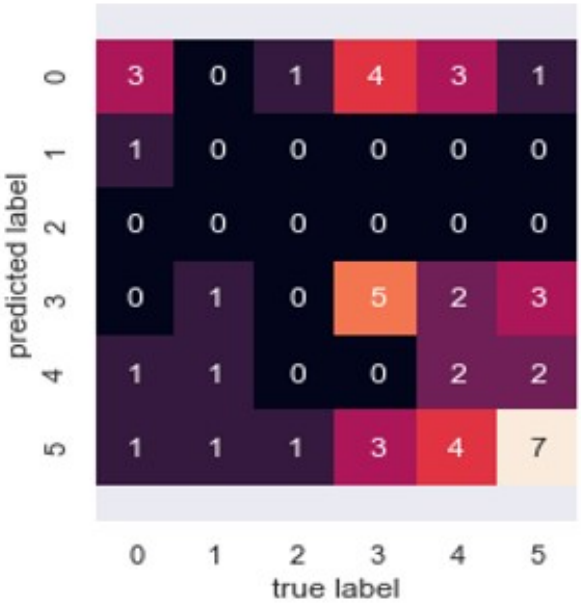

Fig. 4. Random Forest confusion matrix with a heatmap.

5 Experiments and Results

5.1 Convolution Neural Network (CNN)

For a baseline model, a convolutional neural network was created with the intent of gauging the effectiveness of our

deep learning model. To build the baseline model, a sequential model in Keras was used as it stacks sequential layers of the network easily from input to output as seen in Fig. 5.

If every observation in our test dataset was predicted correctly, the model had about 37% accuracy. The model obtained an average of 43% precision and 69% recall as seen in Table. 2. This baseline model was used as a benchmark for comparison with GAN model. The challenge of building a CNN was to automatically assess a breast cancer patient's chances of survival by inspecting gene expressions while maintaining a high accuracy because people's lives are very important.

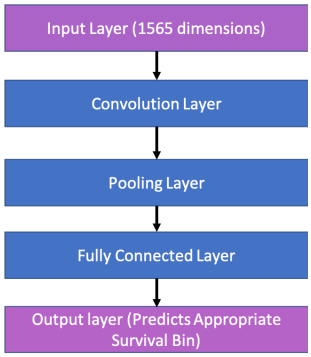

Fig. 5. Structure of the CNN and the layers used in the network.

Evaluation Metric	Value
Accuracy	37.0%
Precision	43.0%
Recall	69.0%
F1-Score	53.0%

TABLE 2
Overall classification metric for baseline CNN.

5.2 Generative Adversarial Network (GAN)

GANs work by having two separate networks: a generator and a discriminator. The generator takes a sample of noise as the input feature and produces an output feature in the desired format. Afterwards, this data is fed into the discriminator, along with a sampling of real data, and the network attempts to determine which datapoints are real and which are fake. Both networks alternate training, essentially forcing the other to become better at its task, until the desired output accuracy is achieved.

The final GAN that was constructed, shown in Fig. 6, was based off of an existing network that was used for images [10]. This network structure achieved good results in that field, and given the high-dimensional nature of this dataset as well, it was presumed that the performance would be similar.

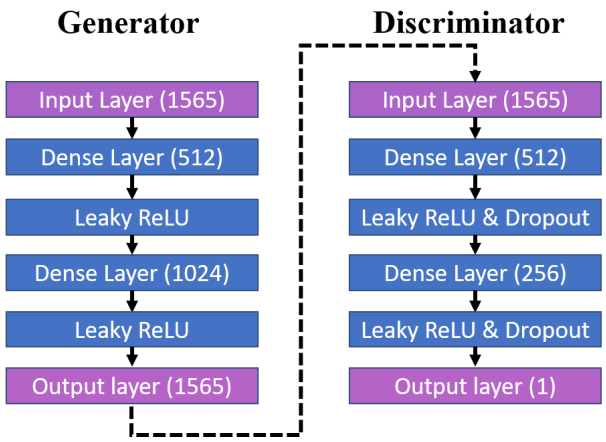

Fig. 6. The GAN structure and the layers used in each network. Layer dimensions are noted in parentheses.

GANs can be used to either generate or discriminate data, depending on which trained network is used. As a result, the trained GAN cannot be used directly to predict cancer patient data. However, since the original dataset is small, the trained generator can be used to augment the existing dataset with new data. This in turn may result in a better accuracy when training the baseline CNN with the augmented data.

To train the GAN, a batch size of 128 generated samples and 128 real samples was used. Training was run for 2048 epochs, with one epoch consisting of training the generator and discriminator separately. Epochs typically took four seconds to execute, giving a total training time of approximately two hour and 20 minutes. This training was performed on a desktop computer with an AMD Ryzen 1600 processor, 16GB DDR4 RAM, and NVIDIA GTX 1060 graphics card. Keras was configured to execute on the graphics card, resulting in the short epoch execution time.

Once training was complete, the generator network was used to generate 4096 samples to augment the training set. The same CNN configuration that was used for the baseline (Fig. 5) was once again used to create a new network for classification. This network was trained using the augmented data set for 256 epochs of batch size 128. Training this network took less than a minute on the same system described above.

In the end, the accuracy of the augmented-trained CNN was 32.1%, which is noticeably worse than the baseline CNN (37%).

5.3 Hyper-parameter Tuning

Three important hyper-parameters controlled the behavior of the neural network and had a significant impact on the performance of the model:

1) Batch size
2) Number of epochs
3) Learning rate optimizer

5.3.1 Batch Size

Batch size was one of the most important hyper-parameters that can be tuned in deep learning. Our goal is to maximize

the performance by minimizing the computational time required. The training time was greatly affected by the batch size as opposed to the value of the learning rate hyper-parameter.

Some research papers suggested that increasing the batch size could be supported by the system's memory. Other research papers suggested modifying the batch size rather than the learning rate.

Using a large batch size guaranteed the convergence to the objective function due to the use of a high learning rate. However, this often leads to poor generalization. On the other hand, a small batch size showed faster convergence to good results since it allows models to start learning even before seeing all the data.

When the batch size is reduced during training, the final loss values are low compared to the large batch sizes that had low early losses. The drawback of smaller batch size is that the model is not guaranteed to converge. Therefore, a small batch size was used initially, and then the batch size was grown steadily throughout training to maintain the accuracy of model.

5.3.2 Number of Epochs

The correct number of epochs was manually determined by trying different values and observing if the validation loss increased. Any increase in validation error provided an indication of over-fitting. The number of epochs was set as high as possible to avoid over-fitting and training was stopped when the validation error began to increase.

5.3.3 Learning Rate Optimizer

Choosing the correct learning rate and optimizer is important to training success in order to avoid gradient explosion. Additionally, it is important because otherwise the network fails to train and takes much longer to converge.

Keras provides various simple stochastic gradient descent algorithms that support adaptive learning rates. As each method adapts to the learning rate, little configuration is often required with learning rate and optimizer. Adam optimizer was the best choice to use as it learns the fastest from the dataset and is more stable than other optimizers.

Adam optimizers work well in practise as they are efficient and converge very fast. Thanks to this, the learning speed of the model is fast. The optimizer resolves the vanishing learning rate and the slow convergence problems in the parameter updates.

5.4 Overall Pipeline

As seen in Fig. 7, our final data pipeline begins with obtaining the NKI dataset, and then performing pre-processing steps (i.e correlation and statistical analysis). After cleaning the data and finding the important features, the next step is the data modeling using a generative adversarial network. Once trained, the GAN is used to generate fake data using random noise as an input and the learned network weights.

The data from the original set is combined with the fake data from the generator which is used to test the discriminator which learns to distinguish real data from the noisy fake data. The output of the discriminator are weights that is used to update the generator network periodically.The combined data is then run through a CNN to predict the patient survival bin.

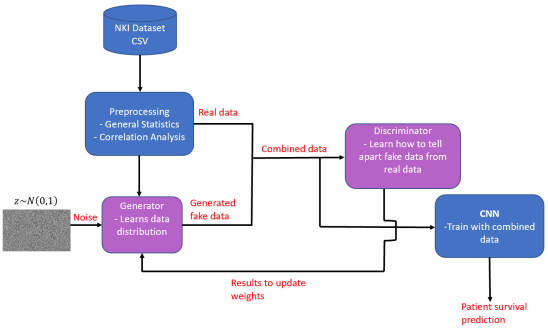

Fig. 7. Generative Adversarial Network Pipeline

6 Group Member Contributions

Elisha Martis

- Writing the "Methodology" and "Experiments and Results" and parts of "Future Work"and "Related Work", sections of the report.
- Helped in pre-processing/cleaning the dataset by adding informative visuals such as the heat map, feature importance, confusion matrix and interpreting the results.
- Helped in setting up the baseline CNN and the GAN using Python.

Ferris Nowlan

- Creating visuals for data exploration (correlation map of survival and top 25 genes, density plot of survival over time).
- Writing "Abstract", "Introduction" and "Dataset Description" sections of the report.

Bronwyn Rowland

- Writing the "Related Work", "Dataset", "Methodology", and "Conclusion and Future Work" sections of the report.
- Performing preliminary research on dataset features.
- Helped in dataset preprocessing by creating visuals and interpreting the results.

Ben Wiebe

- Report formatting (LaTeXand Bibtex) and editing
- Writing GAN, parts of Related Work, and misc. sections
- Preliminary research
- GAN+CNN pipeline creation, iteration, and execution
- GAN+CNN output analysis

7 Replication Package

The code used in this project is available on Github at:
https://github.com/martiselisha/BreastCancerSurvival
The Overleaf project for this report can be found at:
https://www.overleaf.com/read/dsrrczyhprzh

8 Conclusion and Future Work

The GAN-based approach used in this study was only partially successful, with an accuracy (32.1%) that worse than the baseline CNN (37.0%). The dataset proved challenging as it was mainly composed of gene expressions that exhibit nuanced effects on breast cancer and that are not entirely understood yet. This study also highlighted a common data mining challenge of clinical studies, where patient survival is right-censored. Furthermore, lack of information about the dataset features made pre-processing in a way that would be useful for deep learning a challenging task.

Furthermore, the NKI dataset is large and high-dimensional in nature due to the number of gene expressions that were analyzed. As such, next steps could be to search for clusters in a subspace of the dataset features. This could lend itself to feature engineering by combining some features from the original dataset using spectral clustering, for example. Other future work may include using a different neural network architecture as well as a more sophisticated approach to hyper-parameter tuning such as a grid search or bayesian optimization.

References

[1] (2019, February) Breast cancer: early diagnosis and screening. World Health Organization. [Online]. Available: https://www.who.int/cancer/prevention/diagnosis-screening/breast-cancer/en/

[2] R. Kumar, A. Sharma, and R. K. Tiwari. (2012, March) Breast cancer: early diagnosis and screening. [Online]. Available: Applicationofmicroarrayinbreastcancer:Anoverview

[3] van de Vijver MJ, H. YD, van't Veer LJ, H. A. Dai H, V. DW, S. GJ, P. JL, R. C, M. MJ, P. M, A. D, W. A, G. A, D. L, van der Velde T, B. H, R. S, R. ET, F. SH, and B. R, "A gene-expression signature as a predictor of survival in breast cancer." *The New England Journal of Medicine*, no. 347, 2002. [Online]. Available: https://www.nejm.org/doi/full/10.1056/NEJMoa021967

[4] L. Vanneschi, A. Farinaccio, G. Mauri, M. Antoniotti, P. Provero, and M. Giacobini, "A comparison of machine learning techniques for survival prediction in breast cancer," *BioData Mining*, 2011.

[5] L. Ghasem Ahmad, E. AT, A. Pourebrahimi, M. Ebrahimi, and R. AR, "Using three machine learning techniques for predicting breast cancer recurrence," *Journal of Health & Medical Informatics*, January 2013. [Online]. Available: https://www.hilarispublisher.com/open-access/using-three-machine-learning-techniques-for-predicting-breast-cancer-2157-7420.1000124.pdf

[6] S. S. Ch. Shravya, K. Pravalika, "Prediction of breast cancer using supervised machine learning techniques," *International Journal of Innovative Technology and Exploring Engineering (IJITEE)*, April 2019. [Online]. Available: https://www.ijitee.org/wp-content/uploads/papers/v8i6/F3384048619.pdf

[7] S. Simsek, U. Kursuncu, E. Kibis, M. AnisAbdellatif, and A. Dag, "A hybrid data mining approach for identifying the temporal effects of variables associated with breast cancer survival," *Expert Systems with Applications*, January 2020. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0957417419305731?casa_token=tUVk_rn_fc0AAAAA:jeWdqRHCglqkGqMUxm3MZrqN9kq0PHmr-ZwWs4lhaBfd8BVullq8Ove7a_O0I1rSSbaSfOx_-ew

[8] (2017, February) Nki breast cancer data. [Online]. Available: https://data.world/deviramanan2016/nki-breast-cancer-data

[9] L. J. van 't Veer, H. Dai, M. J. van de Vijver, Y. D. He, A. A. M. Hart, M. Mao, H. L. Peterse, K. van der Kooy, M. J. Marton, A. T. Witteveen, G. J. Schreiber, R. M. Kerkhoven, C. Roberts, P. S. Linsley, R. Bernards, and S. H. Friend, "Gene expression profiling predicts clinical outcome of breast cancer," *Nature*, vol. 415, no. 6871, pp. 530–536, 2002. [Online]. Available: https://doi.org/10.1038/415530a

[10] R. Khandelwal. (2019, February) Generative adversarial network(gan) using keras. [Online]. Available: https://medium.com/datadriveninvestor/generative-adversarial-network-gan-using-keras-ce1c05cfdfd3