HIGH SCHOOL EDITION
Journal of Student Research
Volume 11 Issue 3 (2022)

Diagnosing Breast Cancer Using a Novel Dual-Layered Random Forest with Null Handling

Aarav Sharma[1] and Thuy-Anh Nguyen[#]

[1]Archbishop Mitty High School, San Jose, CA, USA
[#]Advisor

ABSTRACT

The purpose of this project was to determine if I could develop an early and accurate model of breast cancer detection that can decrease the mortality rate of women by using novel dual-layered Random Forest with Null Handling. Mammograms have an accuracy of about 86.9% and are susceptible to False negatives, and False positives. In order for my model to be trained and tested, the Wisconsin Data for Breast Cancer was accumulated and duplicated. In the duplicated data, random values were deleted. The first random forest is then trained on x% of the processed data. The next random forest trained on the output of the previous random forest and the processed data. It acted to fine tune results from the previous model. Lastly, the majority of the votes from the individual random forests led to the cancer prediction. I found out that dual-layered random forests with null values in its training data had an accuracy of 94.4%, which is 7% higher than human accuracy. This model also overcame overfitting. All our dual-layered models or models trained with appended null data worked better than human detection and could be built and tested in under 7 seconds with an easy-to-use interface, allowing for results in the same visit to the hospital. The best model had a first layer of 200 trees, a second layer of 800 trees, and accuracy over 94% compared to humans with 86.9%. This model is fast, accurate, and can save people's lives.

Engineering Goals and Purpose

Breast cancer is the most common type of cancer in women globally, occurring in about one-in-eight women. Doctors use mammograms to look for early signs of breast cancer. Mammograms have an accuracy of ~86.9%. A false negative could lead to extreme cases of harmful therapy and may cause severe anxiety. However, mammograms have also increased breast cancer survival rate by over 20%. If people had a better way of analyzing the data, more people would be saved. Mammography is the most widely used breast cancer screening tool, but diagnosing cancer from these images is a challenge. One-in-five cases of breast cancer are missed by radiologists. According to Dr. Mozziyar Etemadi, a research professor from Northwestern University, there are 2 main challenges in diagnosing breast cancer with mammograms: False Negatives, in which the scan appears normal even though cancer is present and False Positive, in which the scan looks abnormal even though no cancer is present. If better data analysis existed, one could find even more cancer patients and save them (especially if one could find them early on). The test could also be extended then to less-at-risk groups as people are not as worried now of misdiagnosis as much, potentially catching more cases. Finally, it can take over 4 years and more money to train someone to examine mammograms, meaning a good algorithm could open up many more resources for nurses to actually treat diseases instead of diagnosing them. Misdiagnosis of cancer might lead to costly or dangerous further testing or treatment, or might mean the public stops taking tests seriously. However, saying someone is healthy when they are ill might cause death. Furthermore, training someone to give a good diagnosis takes a lot of time, resources, and money. Data can also be messy with missing (null) or bad values, making diagnosis harder. The main purpose of this project is to develop an easy to implement algorithm that is accurate and can handle missing (null values) or bad data. The hypothesis for this project is that the

new random forest algorithm as designed will outperform older algorithms and people in its ability to accurately and quickly diagnose cancer on a variety of cancer datasets. The first demonstration presented will be on a breast cancer dataset. The model will use the Random Forest algorithm. Random Forests could help mammographers reduce the number of false alarms without increasing the risk of missing cancer when it's really there.

Introduction

Breast cancer is the most common type of cancer particularly diagnosed in women. It is one of the leading causes of death worldwide. Breast cancer is a cancer that forms in the cells of one of the breasts. Women's breasts are constructed by lobules, ducts, nipples, and fatty tissues. Milk is created in lobules and carried towards the nipple by ducts. Normally tumors grow inside lobules as well as ducts and later form cancer inside the breast. Cancer is always a life-threatening disease.

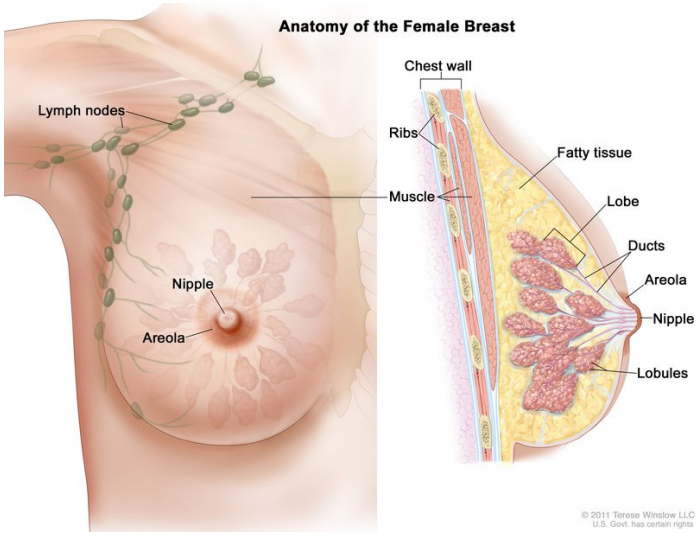

Figure 1: Representation of Female Breast. Adapted from https://visualsonline.cancer.gov/details.cfm?imageid=7127

Breast cancer tumors can be categorized into two broad scenarios: Benign (Noncancerous), and Malignant (Cancerous). The symptoms of breast cancer include a lump in the breast, bloody discharge from the nipple and changes in the shape or texture of the nipple or breast. In breast cancer, there are two types: non-invasive and invasive. Non-invasive breast cancer starts in the milk vessel and does not spread in the other organs even if it grows. But invasive breast cancer is very antagonistic and spreads to other organs and destroys them. Hence, it is necessary to detect the affected cell before it spreads to other nearby organs. Early detection will prevent the death rate of breast cancer patients. As reported by the WHO, there are about 1.38 million new cases and 458,000 deaths from breast cancer each year. Breast cancer is by far the most common type of cancer both in the developed and developing countries. Late diagnosis of this cancer lowers the survivability and, in response, increases the mortality of the patient. There are several ways to detect breast cancer: clinical breast test, mammogram, ultrasound test, magnetic resonance imaging (MRI), blood test, breast biopsy, and molecular breast imaging (MBI). There are several factors which can cause breast cancer in women, i.e., heredity, pregnancy, fat diet, alcohol, and radiation. Mammography is the most widely used breast cancer screening tool, but diagnosing cancer from these images is a challenge. One in five cases of breast cancer is missed by radiologists. One in five women can have missed cancer in a mammogram. According to Dr. Mozziyar Etemadi, research professor from Northwestern university, there are 2 main challenges diagnosing breast cancer with mammograms: False negatives, in which the scan appears normal even though cancer is present and False Positive, in which the scan looks abnormal even though no cancer is present. Mammograms have an

accuracy of ~86.9% (86.9% you have cancer, if it says you have it). The accuracy is worse for younger people (~ 84% for 30-40 years old). A false diagnosis can lead in extreme cases to harmful therapy and may cause severe anxiety for 3 years, but at the very least expensive bills. If better data analysis existed, one could find even more cancer patients and save them (especially if one could find them early on). The test could also be extended then to less-at-risk groups as people are not as worried now of misdiagnosis as much, potentially catching more cases.

Random Forest (RF) is one of the most advanced ensemble learning algorithms and is a very flexible classifier. Random Forest could help mammographers reduce the number of false alarms without increasing the risk of missing cancer when it's really there. Random Forest forms a family of classification methods that depend on a combination of several decision trees and runs efficiently in large databases. Random forest reduces false positives and false negatives. Random Forest performance is better than the other techniques to predict cancer at an early stage. Random Forest reduces the risk of overfitting. For large data, it produces highly accurate predictions. Random Forests can maintain high accuracy with null values. In a nutshell, the involvement of machine learning for breast image classification allows doctors and physicians to take a second opinion, along with satisfaction and confidence of the patient. Machine Learning based diagnostic systems can help the patient to receive timely feedback about the disease which can improve the patient-management scenario. Then RF divides data based on some algorithm into groups that contain mostly yes or no answers, until the ends of a tree are mostly or all yes or no answers. Then the forest makes a new tree with another random sample from the training set and repeats the process. By repeating many times, we can get an accurate model that does not overfit.

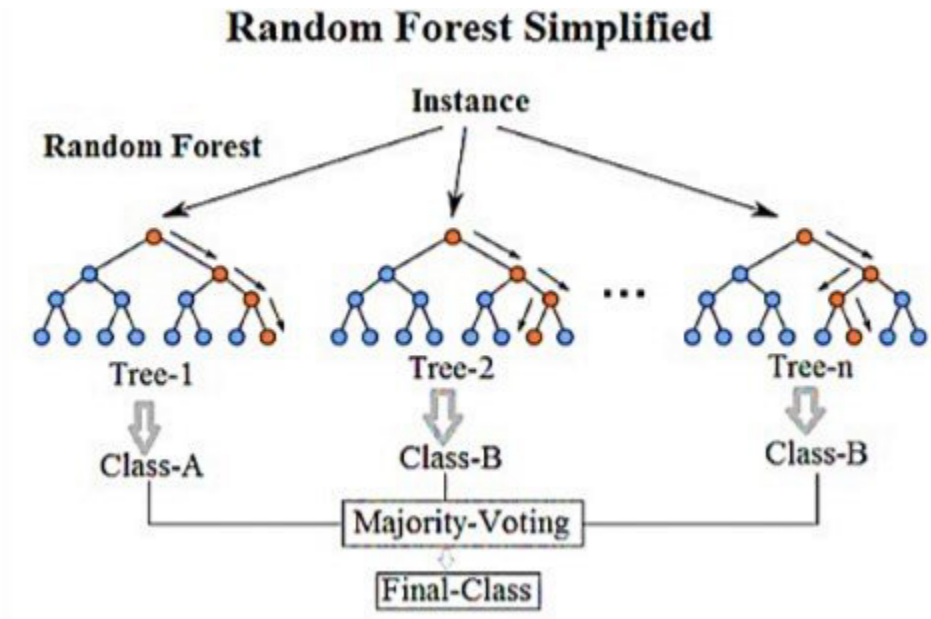

Figure 2. RFs work by taking training data and then taking some of the training data randomly and putting it into a tree (Tree-1). Then it divides data based on some splitting algorithm (either Gini impurity or Entropy) into groups that contain mostly yes or no for cancer. The tree repeats this until the answers at the very end of the tree are mostly yes(blue) or no(red) answers. Then, the forest makes a new tree with another random sample from the training set and repeats the process. By repeating many times, one can get an accurate model that does not overfit, and one can answer questions by putting data into the model and seeing if the majority of trees in the forest say yes or no for the diagnosis. Adapted from: https://en.wikipedia.org/wiki/Random_forest

Overfitting makes a model very reliant on sensitive, fine-tuned parameters. When this happens, the model probably will not work on new data or the testing set. Equations that are simpler probably have more truth to them or

equations relying on more predictors equally. The random selection in a RF prevents overfitting by eliminating sample biases and in that many small trees in a RF also helps eliminating overfitting by:

- Each small tree finds a simple pattern compared to a complex pattern found in a big tree. A simple pattern is less likely to be overfitted and probably has some truth to it (some meaning that it is usually correct, not always).
- By using many small trees, if a majority of them say one answer, one can be confident that it is probably right. This is true even if some are wrong or have found a false pattern in data, as long as the majority of trees are right.

This makes RF ideal to prevent overfitting and give good, generalizable results.

However, traditional RF could not handle null values. This was because choosing a number for a null value would give it a value, and then it will think you made a measurement. A new column for each predictor/independent variable that said if a value was there or not(null) was added because null value could also be zero in actual data. The algorithm also added "fake data" to train it to be capable of handling missing (null) data by adding a data where we copied previous data with no missing values and deleted random values (seen in fig. 3 and 4). This should also prevent overfitting as now the model has to find in some instances a simple model with values missing and find patterns with more variables. This method of training on null values has previously been used in other machine learning algorithms (MLA) like neural networks.

Each tree in the first RF layer only sees a small random sample, which may have randomly more of some variables not deleted, so it may not see important correlations. To overcome this, the model's training dataset gets yes or no predictions from the first RF appended on, and then that new "data" is fed into the second RF. The second layer RF resamples data and will first use patterns found previously (encoded in yes or no predictions appended as seen in fig. 4) but will correct it when the previous model made a wrong guess. This will incorporate correlations that were null before (this will be helped because each tree in the second RF trains on a different random data sample than in the first RF, so it will most likely not find an identical pattern). Fig. 3 in the next section sums up this new algorithm's design.

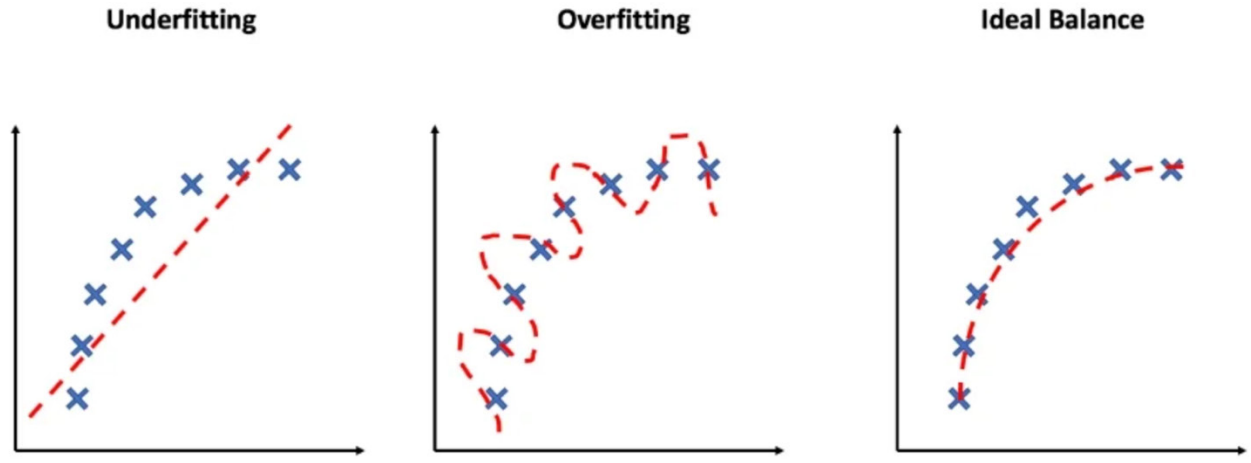

Figure 3: Overfitting is when your model finds too complex of an equation to fit data or relies on too few predictors. This makes the model very reliant on fine-tuned parameters. When this happens, the model probably will not work on new data or the testing set. Equations that are simpler probably have more truth to them than equations relying on more predictors but less on each individual predictor. Adapted from: https://subscription.packtpub.com/book/data/9781838556334/7/ch07lvl1sec82/underfitting-and-overfitting

RF can split data into a node (also called a leaf node) in a tree using many different splitting algorithms. Two of the most widely used splitting algorithms are Gini impurity and Entropy/information gain. In an Entropy and Gini Impurity, both measures how well they can split data into yes or no diagnostics. Gini impurity is biased towards getting both answers (yes or no) right the same amount of time and wrong the same amount of time or in other-words false negative/true negative= false positive/true positive. Entropy splitting algorithm, which is in scikit learning (a python module used to make random forests) is similar to gini impurity and is biased to either make more true positives or negatives. This is shown in figure 4.

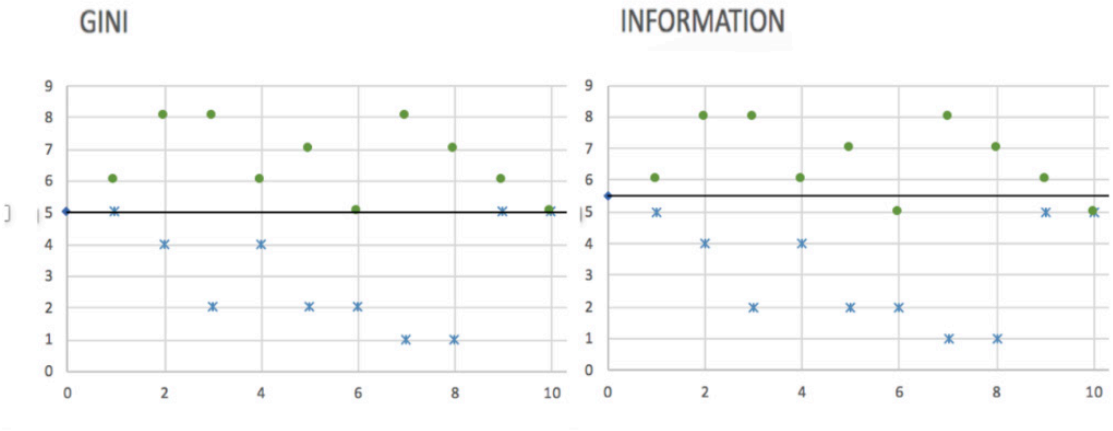

Figure 4: The goal in this example is to split the data on one side of the horizontal line, which is in green and on the other side, in blue. You can't do this perfectly, but if we use the Gini impurity, both sides will have the same amount of wrong guesses percentage wise, while with Entropy/Information Gain, one side will have right guesses and the other side will have more wrong guesses.

Design criteria, testing, and evaluation

Criteria: The model must run relatively fast, can run on a laptop, and train on a laptop with easy to use and install highly compatible software. The model must be able to handle null values. If this criteria is met, the experimenter will test it in the following manner:

1. Train model on computer and record run time for 10% training and 90% testing data. Optimize what criteria the model uses to split data (Gini impurity or Entropy), number of trees in each layer, and how many values are to be deleted based on the first accuracy on training set with null values and without, then other measures (the time it takes, how much null values it can handle, etc.).

A. Optimization is done by trying a wide range of random values for a number of trees in the first and second layer separately, selecting which splitting algorithm to use(either Gini impurity or Entropy) in each layer, and deciding how many null values we put in each layer. Then, we will record the accuracy on testing data with and without null values, and the speed at which it ran. The Independent Variable (IV) will include which splitting algorithm to use (either Gini impurity or Entropy) in each layer, the number of trees in each layer, and the number of values we will make null. The Dependent Variable (DV) will be the measures of performance (accuracy on testing data with and without null values), and the speed at which it ran. For each run,

all IV and DV measurements were recorded. At least 13,000 data points were collected before the next step, Step B.

B. Plot heat/density maps, which are a graphical representation of data where values are depicted by color, with two IV (axes) and one DV (DV gives color). This is used to try to find conservatively the best DV range. Step 1A is repeated with a smaller range of values that one has now found until we find what one thinks is the best DV values.

2. Repeat Step 1 with 5% training and 95% testing, then 30% training and 70% testing, and finally 40% training and 60% testing.

3. Once the model is optimized, run the model many times to see how accurate the model is and how much it fluctuates. Collect 13,000 runs/data-points for accuracy.

4. Repeat Step 3, but with the model containing no missing/null values for the training dataset.

5. Compare results of all algorithms and the best human using z-tests(which help to identify how random something is) and report means. If the model works, it will have a statistically significantly higher accuracy.

Preliminary and final design:

The following diagrams outline preliminary designs and final, along with the rational.

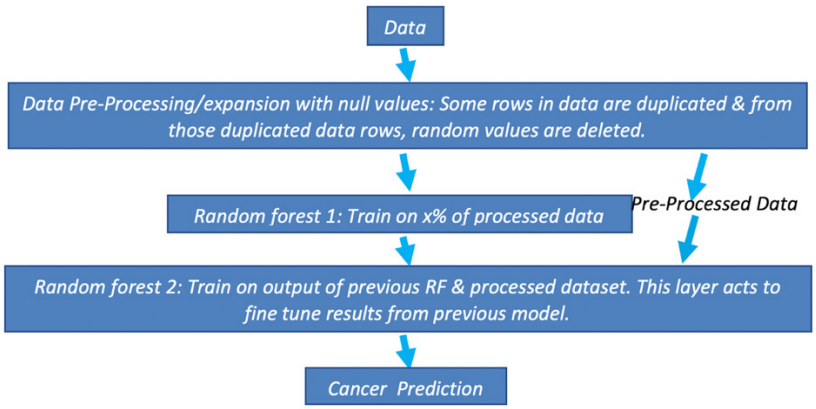

Figure 5: The model takes the data and splits it into a testing and training set. The algorithm does not train on the testing set. The testing set is used instead to see how well the model works.

As shown in Figure 5, the algorithm takes the data and splits it into a testing and training group. The algorithm does not train on the testing group. Instead, the testing group is used to see how well the model works. Additionally, the algorithm takes the data to predict cancer (X values) and duplicates it so there is an identical set. Then, the algorithm deletes random values in set 2, so that it allows the model to learn with more data to train from, learn to handle null values, and does not overfit. Fig. 5. shows how data is pre-processed in more depth. Overfitting is when a model fits too few predictor variables or fits data too tightly, causing the model to not be able to generalize well as seen in

fig. 3. By adding null values to the training set, the model cannot rely on any one variable to predict and needs to figure out how to deal with all sorts of data to prevent overfitting (one of most common problems in MLA).

Table 1: Duplication of data with Null values (Representation)

Original Data:

Dataset	Prediction Variables				Target
	Variable 1	Variable 2	Variable 3	Variable 4	
1	3	7	5	9	Y

Data Duplication:

Dataset	Prediction Variables				Target
	Variable 1	Variable 2	Variable 3	Variable 4	
1	3	7	5	9	Y
Copy	3	7	5	9	Y

Delete random values in duplicate and add a column saying if deleted (1) or not deleted (0).

Dataset	Prediction Variables								Target
	Variable 1	Deleted or Not	Variable 2	Deleted or Not	Variable 3	Deleted or Not	Variable	Deleted or Not	
1	3	0	7	0	5	0	9	1	Y
Null values	0	1	7	0	0	1	9	1	Y

Note: An extra column is added, saying yes (1) if deleted or not deleted (0). This is used to identify null values.

Prototype:

Materials:

- Python 3.0 (modules: scipy, random, numpy, pandas, and time module).
- Scikit Learn Python Module
- Mathematica
- Laptop
- Dataset from: https://www.kaggle.com/uciml/breast-cancer-wisconsin-data, with 357 cases of no cancer and 212 cancer cases (UCI)

The prototype was built using the designs shown in Figure 5 and 6 for the breast cancer dataset, with testing variables for optimization being amounts of trees in each layer, null values, and splitting algorithms as described in Section 4, which includes Design Criteria, Testing, and Evaluation. For the prototypes, these values were randomly chosen in a set range. The exact parameters of the data-set from the website description are as follows in quotes (UCI) on the next page.

“Features are computed from a digitized image of a fine needle aspirate (FNA) of a breast mass. They describe characteristics of the cell nuclei present in the image in the 3-dimensional space that is described in: [K. P. Bennett and O. L. Mangasarian: "Robust Linear Programming Discrimination of Two Linearly Inseparable Sets", Optimization

Methods and Software 1, 1992, 23-34]. This database is also available through the UW CS ftp server: ftp ftp.cs.wisc.edu
cd math-prog/cpo-dataset/machine-learn/WDBC/
Also can be found on UCI Machine Learning Repository:
https://archive.ics.uci.edu/ml/datasets/Breast+Cancer+Wisconsin+%28Diagnostic%29

Attribute Information:

1) ID number
2) Diagnosis (M = malignant, B = benign)
3) 32 Predictors

Ten real-valued features are computed for each cell nucleus:

a) radius (mean of distances from center to points on the perimeter)
b) texture (standard deviation of gray-scale values)
c) perimeter
d) area
e) smoothness (local variation in radius lengths)
f) compactness (perimeter^2 / area - 1.0)
g) concavity (severity of concave portions of the contour)
h) concave points (number of concave portions of the contour)
i) symmetry
j) fractal dimension ("coastline approximation" - 1)

The mean standard error (SE) and "worst" or largest (mean of the three largest values) of these features were computed for each image, resulting in 29 features. For instance, field 3 is Mean Radius, field 13 is Radius SE, field 23 is Worst Radius. All feature values are recorded with four significant digits.
Missing attribute values: none
Class distribution: 357 benign, 212 malignant
Data was originally from Breast Cancer Wisconsin (Diagnostic) Dataset”

Initial testing and evaluation

The design was tested with random values in order to determine what splitting algorithm to use in each forest (Gini impurity or Entropy), number of trees in each forest, amount of null values used, and whether the algorithm was a control or experiment. Gen. 1, was programmed only to test if the algorithm worked with null values included in the training set. Gen. 2, had null and no null values (control) included in the training set. The number after an underscore indicates what model it was, with later models being cleaner (better comments and cleaner code, deleted lines that were used to fix bugs (mostly due to data type incompatibility or array problems). Programs can be found in the supplemental section or lab notebook. The model was tested at 5%, 10%, 30%, and 40% of data being training data. No difference in what values were the best were seen. It was seen that a broad range of decision tree values were optimal, but 400 for each layer were chosen to make both forests equal size. Null value amount variable was set to 2 (this means on data appended, half of values were deleted) was chosen as it was in the range of best null values and would allow theoretically for it to train on some of most missing data. The best criteria to split nodes along was found to be Gini impurity in the first RF, and in the second RF it was found to be Entropy.

Each run to optimize appended at least 13,000 points (parameters of model tested, all measures of accuracy, and accuracies compensated for null values) to a .csv file, before being plotted. Accuracy compensated for null values means that accuracy scores were increased if more data was missing, since theoretically it would be harder to achieve the same fit with less data. Five optimizations were made. An example of accuracy compensated for null values would

be that if two runs scored 1.0 (is 100% pure) for accuracy, but run A had ½ of its values missing and run B had none of its values missing, then Run A would have a greater score compared to run B with a score of 1.0.

Density or heat map shows values of a phenomena as color in two dimensions. Different colors represent different numbers and can be used to figure out if a section on the map has low or high values for a measurement. If a region of a map has a high value, it is colored differently than one with low values. If one region in a heat map with x and y locations, representing the number of trees in each layer, for example, has a high value for accuracy that one wants, one can look at where that region is to find what values would give a high value for accuracy. On the next page is an example of data used to optimize a run (only graphs that were useful for optimization are shown).

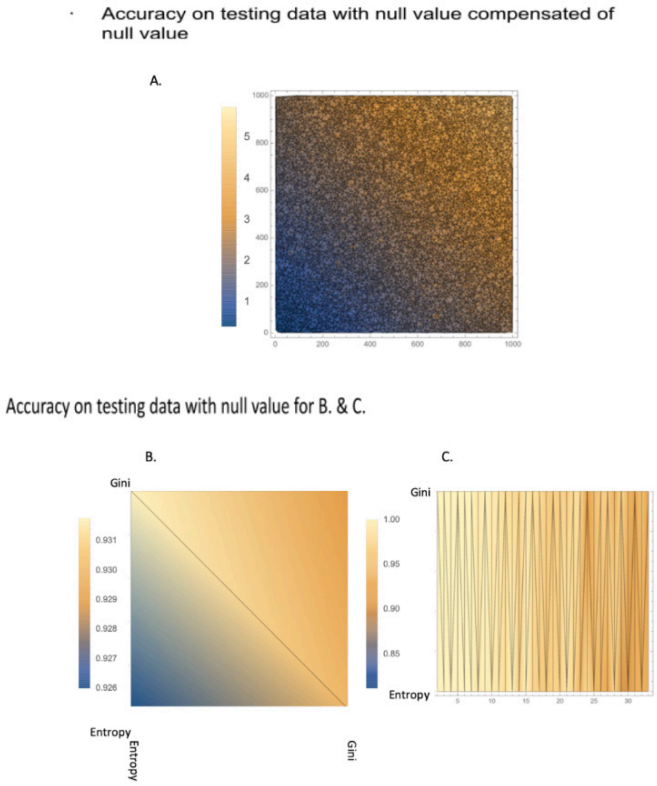

Figure 6: Heat Maps are shown above. Figure A displays the accuracy of testing data with null value compensated of a null value. As you can see, the lighter the color, the greater the accuracy. Figure B and Figure C are showing the Gini impurity and entropy layer successfully working together. The same pattern is present in Figure B. When the Gini impurity is on the y-axis and the Entropy is on the x-axis, the accuracy is the highest. Figure C shows layer 1 splitting criteria versus the number of null values. The colors correspond to accuracy on testing data with null values. As a result, higher accuracy is better.

Redesign, testing, and evaluation

After hyperparameters were chosen, the following models were built and tested:

Table 2: Decided initial tests. Gini impurity was used for 1-layer forest as it worked best for that model.

	Global	Forest 1		Forest 2	
Model	Null Values	Trees	Splitting Criteria	Trees	Splitting Criteria

2-layer RF with null values in training	2	400	Gini impurity	400	Entropy
2-layer RF	NA	400	Gini impurity	400	Entropy
1-layer RF with null values in training	2	800	Gini impurity	NA	NA
1-layer RF	NA	800	Gini impurity	NA	NA

However, after running the single layered RF, our new model seemed to be levelling off with the second layer, not helping that much or at all after 1000 trees were used in total (600-1000 trees in dual forest gave the same answers as one forest). Thus, it was decided then to make the first layer smaller at 300 trees, which helped in performance. 300 trees were chosen as no difference was seen in this layer if 300 or more trees were used. This was probably not caught in the first round of optimization, as it was too "low resolution" (not enough data points close together to see the difference). Decreasing the first layer to 200 showed additional benefits and was done later. Fig. 8 on the below shows the new model.

Table 3: Decided final test parameters.

	Global	Forest 1		Forest 2	
Model	Null Values	Trees	Splitting Criteria	Trees	Splitting Criteria
2-layer RF with null values training	2	200	Gini impurity	800	Entropy
2-layer RF	NA	200	Gini impurity	800	Entropy
1-layer RF with null values in training	2	1000	Gini impurity	NA	NA
1-layer RF	NA	1000	Gini impurity	NA	NA

Results

The following figures (fig. 9 to 14) shows that adding null values significantly improved the new model, and the new model could handle better prediction without null values added than a single RF. Figures 9 and 12 show that double layer vs. single layer, bigger training samples, and using null values in the training consistently improve outcome for accuracy on testing set with the null values included. For accuracy on testing set with no null included, the only significant increase in accuracy due to sample size was significant. This indicates the model was not overfitting on this data. This was seen in figure 10 and 13. The last two figures, 11 and 14 showed that accuracy was significantly less (around 84 to 89%) on a training dataset with null values for a single forest with no null values appended to training. This indicates that the normal, single layer RF over-fitted, unlike any of the other models, on training data with no null values appended. All the other measures on the training data for all the other models were 100% accurate. Compared to a human accuracy of ~86.9%, all were significantly better. All of this was seen for a wide range of training versus testing sample sizes. For a testing dataset with null values, all the models that trained with null values outperformed humans who have an accuracy of around 86.9% by at least 2% more. The best model was the dual layered model with null values and had an accuracy of around 94.4%, over 7% higher than humans. This should be even higher in reality, because people are not trained and evaluated on datasets with null values. For accuracy on the testing set with no null values, everything outperformed humans with the lowest value being around 92.2% and the highest being around 95.1%. All testing set accuracy was above 87%.

Table 3: Testing with null values. S means single layer, D means Dual layer. Bold, red, italic, and highlighted means trained with nulls, and number after it is the percent used in training. NA is not relevant. The mean is listed in the left column, if it is significantly bigger than the column heading, it is +; if it is smaller, it is —; and if it is not significantly different, it is 0. The cutoff is p-value of 0.001.

	S5	D5	S5	D5	S10	D10	S10	D10	S30	D30	S30	D30	S40	D40	S40	D40

S5, 0.8293	NA y	0	_	—	0	—	—	—	—	—	—	—	—	—	—	—
D5, 0.8282	NA	N A	—	—	—	—	—	—	—	—	—	—	—	—	—	—
S5 ,0.893 9	NA	N A	N A	—	+	+	—	—	+	+	—	—	+	+	—	—
D5, 0.9211	NA	N A	N A	N A	+	+	+	—	+	+	—	—	+	+	—	—
S10, 0.8418	NA	N A	N A	N A	NA	0	—	—	—	—	—	—	—	—	—	—
D10, 0.8415	NA	N A	N A	N A	NA	NA	—	—	—	—	—	—	—	—	—	—
S10, 0.9165	NA	N A	N A	N A	NA	NA	NA	—	+	+	—	—	+	+	—	—
D10, 0.9319	NA	N A	N A	N A	NA	NA	NA	NA	+	+	—	—	+	+	—	—
S30, 0.8464	NA	N A	N A	N A	NA	NA	NA	NA	NA	—	—	—	0	—	—	—
D30, 0.8509	NA	N A	N A	N A	NA	NA	NA	NA	NA	NA	—	—	+	0	—	—
S30, 0.9357	NA	N A	N A	N A	NA	NA	NA	NA	NA	NA	NA	0	+	+	—	—
D30, 0.9346	NA	N A	N A	N A	NA	NA	NA	NA	NA	NA	NA	NA	+	+	—	—
S40, 0.8457	NA	N A	N A	N A	NA	NA	NA	NA	NA	NA	NA	NA	NA	0	—	—
D40, 0.8520	NA	N A	N A	N A	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	—	—
S40, 0.9393	NA	N A	N A	N A	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	—
D40, 0.9436	NA	N A	N A	N A	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA

Table 4: Testing with null values. S means single layer, D means Dual layer. Bold, red, italic, and highlighted means trained with nulls, and number after it is the percent used in training. NA is not relevant. The mean is listed in the left column, if it is significantly bigger than the column heading, it is +; if it is smaller, it is —; and if it is not significantly different, it is 0. The cutoff is p-value of 0.001.

	S5	D5	S5	D5	S10	D10	S10	D10	S30	D30	S30	D30	S40	D40	S40	D40
S5, 0.9231	NA	0	0	0	—	—	—	—	—	—	—	—	—	—	—	—

D5, 0.9228	NA	NA	0	0	—	—	—	—	—	—	—	—	—	—	—	—
S5, 0.9224	NA	NA	NA	0	0	0	—	—	—	—	—	—	—	—	—	—
D5, 0.9220	NA	NA	NA	NA	0	0	—	—	—	—	—	—	—	—	—	—
S10, 0.9359	NA	NA	NA	NA	NA	0	0	0	—	—	—	—	—	—	—	—
D10, 0.9355	NA	NA	NA	NA	NA	NA	0	0	—	—	—	—	—	—	—	—
S10, 0.9322	NA	NA	NA	NA	NA	NA	NA	0	0	0	—	—	—	—	—	—
D10, 0.9319	NA	NA	NA	NA	NA	NA	NA	NA	0	0	—	—	—	—	—	—
S30, 0.9506	NA	NA	NA	NA	NA	NA	NA	NA	NA	0	0	0	—	—	—	—
D30, 0.9507	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	0	0	—	—	—	—
S30, 0.9458	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	0	0	0	—	—
D30, 0.9456	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	0	0	—	—
S40, 0.9541	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	0	0	—
D40, 0.9544	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	0	0
S40, 0.9496	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	0
D40, 0.9493	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA

Title: Accuracy on training data with null values for single layer forest with no special null training. All other methods are 100% accurate.

Table 5: Training with null values. All other methods achieved a perfect accuracy and were significantly bigger. Therefore, they are not shown.

	S5	S10	S30	S40
S5, 0.8847	NA	0	+	+
S10, 0.8874	NA	NA	+	+
S30, 0.8785	NA	NA	NA	+
S40, 0.8745	NA	NA	NA	NA

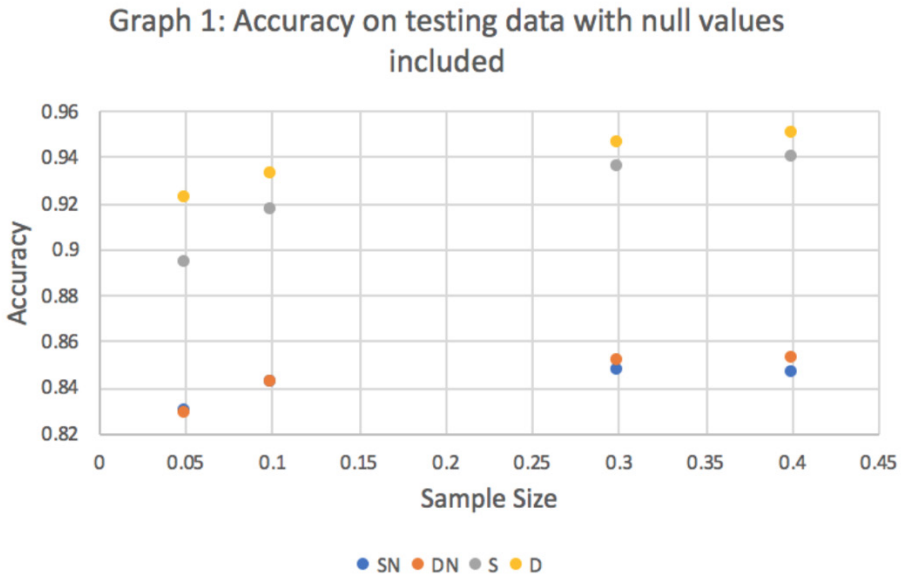

Figure 7: This graph illustrates the accuracy of testing data with null values included. SN represents a Single Layer of either Gini Impurity or Entropy (aka Information Gain) with null values, and DN symbolizes Double Layers with Null Values in its training data. The other two, S and D, stand for Single and Double Layers, respectively, without null values in its training data. As seen in the graph above, Double and Single Layers without null values worked pretty well, while Single Layered and Double Layered with Null Values in its training data had a respectable maximum accuracy of about 86%.

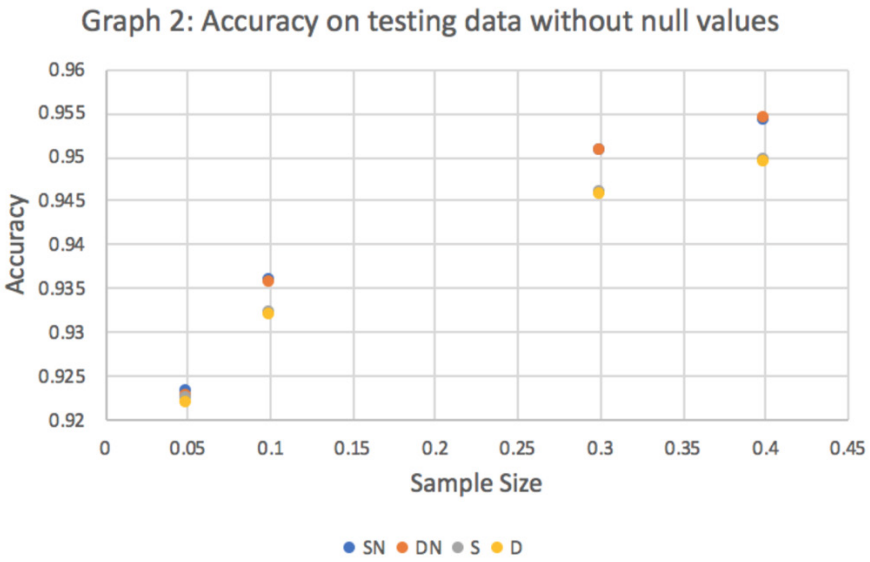

Figure 8: Accuracy for testing data with no null values. D stands for double, S stands for single layer, and N means no null value used in training. The change in accuracy was significant due to the sample size. This indicates it was not overfitting on this data.

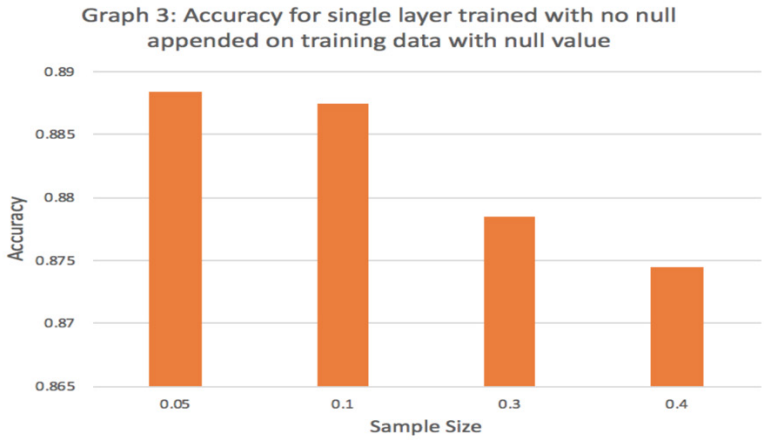

Figure 9: Training with null values. Only the single layers which are trained without null values are shown above for accuracy on training data with null values, as the rest of them have perfect accuracy on training data with null values. This indicates that the single layer that trained with no null values overfitted. 0.05 compared to 0.1 sample size accuracy was not significantly different. The same was with a sample size of 0.3 compared with a sample size of 0.4.

Once this was done, a user interface (ModelUI.py as prototype and then cleaner version called ModelUIclean.py) was designed with the final being user interface program, labelled as FinalUserInterFaceProgram.py and previous ones being older generations to fix bugs or were not as neat of code (or well commented). To prepare the user interface, one trains the model using CodeGenerator.py (which generates the code), which then saves forests to a file labelled FinalRandomForest1.py (this saves the random forests). Downloading those files and running this file BreastCancerProject.py in python, which is compatible with windows, mac, or Linux, allows for one to use the interface. The interface asks for each value to be typed in and will write an answer as shown in fig. 15. The whole model ran in under 6 seconds (including training) on my MacBook computer.

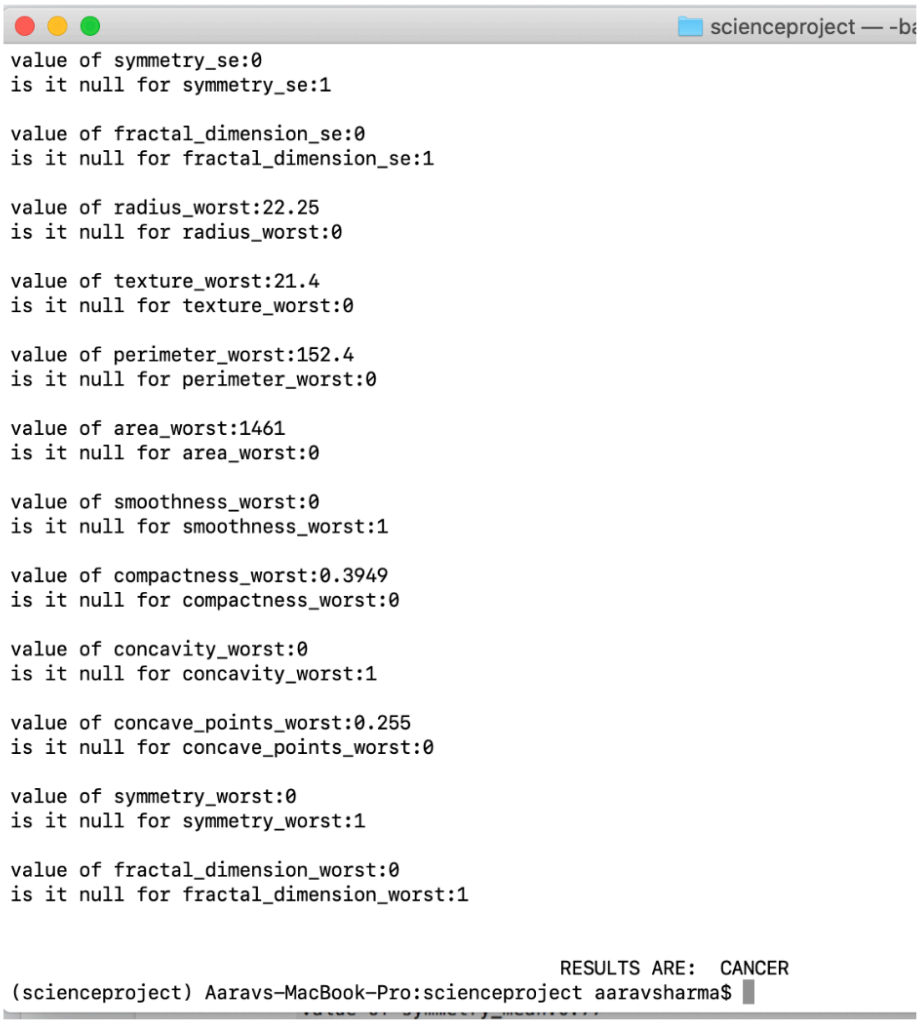

Figure 10: Example of how user input and output appear.

Discussion

In this article, we gave an overview of supervised learning and Random Forest. We applied these concepts and techniques to a data set of patients and we developed a model to diagnose breast cancer. We find that our model is 94% accurate, which is certainly much better than mammograms, which would be 87% accurate, but not satisfactory enough for the model. To improve the accuracy of the model we propose to explore more complex types of models, such as neural networks and use computer vision. These research directions will be pursued in the future and reported in a future article.

Conclusion

It is hard to diagnose if someone has many forms of cancers. Saying someone has cancer when they do not, might lead to costly or dangerous further testing or treatment, or might mean people stop taking tests seriously. However, saying someone is healthy when they are not, could mean death. Additionally, training someone to give a good diagnosis takes a lot of time, resources, and money. Data can also be messy with null values, making diagnosis harder.

To fix this, the experimenter attempts to develop an easy to train and use algorithm that is accurate and can handle null or bad data. The hypothesis was that the new algorithm as outlined will outperform in its ability to accurately diagnose cancer and its speed compared to trained people and existing algorithms on a variety of breast cancer datasets. The project demonstrated this as our model worked better for a wide range of training sizes, and for all training sizes separately and some together, using our null value-added training method and/or dual layers improved performance compared to controls without one or without either. All our dual layered models or models trained with appended null data worked better than human detection and could be built and tested in under 7 seconds with an easy to use interface, allowing for results in the same visit to the hospital. The best model had a first layer of 200 trees, second layer of 800 trees, and accuracies over 94% compared to humans with 86.9%.

The final design could be improved with more hyperparameter fine-tuning, more data, a better user interface, and perhaps instead of making a prediction of yes or no, to send in between layers using an algorithm that counts in the first layer how many yes or no votes are present.

Limitations

While the results in this paper are promising, they do have some limitations. These limitations are likely to be due to the fact that we have used Random Forest instead of the more complex technique of neural networks.

Acknowledgements

I would like to thanks my mentor, Mrs.Thuy-Anh Nguyen, Science Research Mentor, Archbishop Mitty High School.

Works Cited

Breast Cancer Surveillance Consortium. “sensitivity, specificity, and false negative rate for 1,682,504 screening mammography examinations from 2007 - 2013.” BCSC, National Cancer Institute, 31 Dec. 2014, www.bcsc-research.org/statistics/screening-performance-benchmarks/screening-sens-spec-false-negative .
Howley, Elaine K, and Anna M Miller. “False Positives, False Negatives in Breast Cancer.” U.S. News & World Report, U.S. News & World Report, 18 Apr. 2019, health.usnews.com/health-care/patient-advice/articles/2017-04-13/false-positives-false-negatives-in-breast-cancer.
Horev, Rani. “BERT Explained: State of the Art Language Model for NLP.” Medium, Towards Data Science, 17 Nov. 2018, towardsdatascience.com/bert-explained-state-of-the-art-language-model-for-nlp-f8b21a9b6270.
Koehrsen, Will. “Random Forest Simple Explanation.” Medium, Medium, 27 Dec. 2017, medium.com/@williamkoehrsen/random-forest-simple-explanation-377895a60d2d.
mc.ai. “Overfitting and Underfitting (BUG IN ML MODELS).” Mc.ai, Deep Learning on Medium, 7 Jan. 2020, mc.ai/overfitting-and-underfitting-bug-in-ml-models.
Morris, Elizabeth et al. “Implications of Overdiagnosis: Impact on Screening Mammography Practices.” Population health management vol. 18 Suppl 1, Suppl 1 (2015): S3-11. doi: 10.1089/pop.2015.29023.mor
Silipo, Rosaria. “From a Single Decision Tree to a Random Forest.” Medium, Towards Data Science, 8 Oct. 2019, towardsdatascience.com/from-a-single-decision-tree-to-a-random-forest-b9523be65147.
Study.com. STUDY.COM, Study.com, 1 Apr. 2020, study.com/mammography_training.html.
UCI. “Breast Cancer Wisconsin (Diagnostic) Data Set.” Kaggle, UCI Machine Learning, 25 Sept. 2016, www.kaggle.com/uciml/breast-cancer-wisconsin-data.
World Health Organization. “Breast Cancer.” World Health Organization, World Health Organization, 12 Sept. 2018, www.who.int/cancer/prevention/diagnosis-screening/breast-cancer/en/.

Nahid, Abdullah-Al, and Yinan Kong. “Involvement of Machine Learning for Breast Cancer Image Classification: A Survey.” Computational and Mathematical Methods in Medicine, Hindawi, 31 Dec. 2017, www.hindawi.com/journals/cmmm/2017/3781951/ .
January 01, 2020 | By Marla Paul. Artificial Intelligence Improves Breast Cancer Detection on Mammograms in Early Research, news.northwestern.edu/stories/2020/01/ai-breast-cancer/.
“AI Could Help Radiologists Interpret Mammograms More Accurately.” Stanford School of Engineering, 9 Sept. 2019, engineering.stanford.edu/magazine/article/ai-could-help-radiologists-interpret-mammograms-more-accurately.
Dietsche, Erin, et al. “Doc.ai Is Creating Robo-Doctors That Can Converse with Patients (Updated).” MedCity News, 17 Dec. 2018, medcitynews.com/2017/08/doc-ai/.
Demaitre, Eugene, et al. “COVID-19 Pandemic Prompts More Robot Usage Worldwide.” The Robot Report, 11 June 2020, www.therobotreport.com/covid-19-pandemic-prompts-more-robot-usage-worldwide/ .
“AI to the Rescue: Robot Nurses Deliver Medicines to COVID-19 Patients in Tamil Nadu.” News18, News18, www.news18.com/news/buzz/ai-to-the-rescue-robot-nurses-deliver-medicines-to-covid-19-patients-in-tamil-nadu-2563569.html .
Dickson, Ben. “How AI Can Determine Which Coronavirus Patients Require Hospitalization.” Neural | The Next Web, 3 Apr. 2020, thenextweb.com/neural/2020/04/02/ai-can-help-manage-hospital-resources-during-the-coronavirus-crisis-syndication/ .
“How AI Is Transforming the Future of Healthcare.” Corporate, www.internationalsos.com/client-magazines/in-this-issue-3/how-ai-is-transforming-the-future-of-healthcare .
Patrick GALEY, AFP. “AI Is Now Officially Better at Diagnosing Breast Cancer Than Human Experts.” ScienceAlert, www.sciencealert.com/ai-is-now-officially-better-at-diagnosing-breast-cancer-than-human-experts .