Prediction of Benign or Malignant Tumor Cells Based on Cell Measurements: A Neural Network Approach

ELI BRIGNAC, XIAOFAN LI, and JONATHAN MA, University of Delaware

Accurate diagnosis of tumor cells as benign or malignant is crucial for the effective treatment of cancer patients. In this study, we propose a machine learning approach to predict whether a tumor cell is benign or malignant based on its measurements. We use a dataset containing 569 tumor cell samples, with 357 samples classified as benign and 212 samples classified as malignant.
We use linear regression, and multi-layered perceptrons to build predictive models. Our results show that our proposed model achieved an accuracy of over 98% in predicting the tumor cell's classification as benign or malignant based on its measurements alone. Our model's high accuracy demonstrates the potential of neural networks in aiding medical practitioners in accurately diagnosing tumors.

Our approach offers several advantages over traditional diagnostic methods, which are often subjective and can rely on experience and training of individual practitioners. The neural network approach we propose can provide consistent and objective results, reducing the chances of misdiagnosis and improving patient outcomes.

In conclusion, our study demonstrates the potential of neural networks in improving the accuracy of tumor cell classification. Our proposed approach can provide a reliable and objective tool for medical practitioners to diagnose and treat cancer patients accurately and effectively.

Additional Key Words and Phrases: datasets, neural networks, bioinformatics

ACM Reference Format:
Eli Brignac, Xiaofan Li, and Jonathan Ma. 2023. Prediction of Benign or Malignant Tumor Cells Based on Cell Measurements: A Neural Network Approach. 1, 1 (May 2023), 7 pages.

1 INTRODUCTION

Accurate classification of tumor cells as benign or malignant is crucial for effective cancer treatment. However, traditional diagnostic methods can be costly, invasive, and subjective, leading to potential misdiagnosis [2]. In this study, we propose a machine learning approach using linear regression and multi-layered perceptrons to predict tumor cell classification based on 30 different metrics regarding its shape and size.

Our dataset comprises 569 tumor cell samples, with 357 samples classified as benign and 212 samples classified as malignant. We use linear regression and multi-layered perceptrons to build predictive models, leveraging their ability to identify complex nonlinear patterns in the data.

Our model achieved an accuracy of over 98%, demonstrating the potential of machine learning in aiding medical practitioners in accurately diagnosing tumors. Our approach offers several advantages over traditional diagnostic methods, providing consistent and objective results.

Furthermore, our model's ability to identify complex nonlinear patterns in the data sets it apart from traditional diagnostic methods that rely on a single or a few metrics. By incorporating 30 different metrics, we can obtain a comprehensive understanding of tumor cells, enabling accurate diagnosis.

Authors' address: Eli Brignac, ebrignac@udel.edu; Xiaofan Li, xiaofan@udel.edu; Jonathan Ma, johnma@udel.edu, University of Delaware, 210 South College Ave, Newark, Delaware, 19713.

In conclusion, our study demonstrates the potential of linear regression and multi-layered perceptrons in accurately predicting tumor cell classification. Our proposed approach can provide a reliable and objective tool for medical practitioners to diagnose and treat cancer patients accurately and efficiently, reducing the need for invasive diagnostic procedures and ultimately leading to improved patient outcomes.

2 RELATED WORK

Min et al. extensively cover contemporary research efforts involving bioinformatics, and deep learning methods [2] . We drew heavily upon this paper to find a problem space which we believed multi-layered perceptrons would fit.

A notable outcome from this paper was that we learned emergent architectures in machine learning bioinformatics are primarily deep learning based.

Indeed, a machine learning architecture we had also researched was support vector machines. Yang covered, in 2004, the usage of support vector machines in bioinformatics, particularly, in the fields of DNA and Protein encoding, and gene expression [3]. However, the majority of papers we had read regarding SVMs were dated.

We found a paper written by Islam et al. detailing practical machine learning applications in breast cancer malignancy classification, and contrasting various machine learning methods, including using support vector machines, K-nearest neighbors, artificial neural networks, random forests, and logistic regression. Their comparative study inspired us to study neural networks in our project, as they found ANNs to be the most effective when predicting malignancy of breast cancer cells [1].

3 DATASET

The dataset we used is publicly available at the following link: Cancer Data Dataset.

The dataset used in this study comprises a rich set of features extracted from diagnostic images and clinical assessments. These features include measurements related to the tumor's size, shape, texture, and smoothness. Additionally, compactness, concavity, and symmetry attributes are also incorporated. Each tumor sample is represented as a vector of numerical values, coupled with the labeling of each cell's diagnosis enabling the use of supervised learning methods. In total, there are 569 entries of which 357 are Benign and 212 of them are malignant.

Below, we present a detailed overview of the dataset's key characteristics, along with a table that categorizes each geometric measurement into 10 categories each with 3 attributes (Table 1.):

- **id**: The id of the cancer cell
- **diagnosis**: The diagnosis of the cancer cell. 'M' for malignant and 'B' for benign
- **radius_mean**: The average of the distances from the center to points on the perimeter of the tumor.
- **radius_mean**: The average of the distances from the center to points on the perimeter of the tumor.
- **radius_se**: The standard error of the mean of the distances from the center to points on the perimeter of the tumor.
- **radius_worst**: The "worst" or largest mean value of the distances from the center to points on the perimeter of the tumor among all nuclei.
- **texture_mean**: The average gray-scale intensity values of the pixels in the tumor boundary.
- **texture_se**: The standard error of the gray-scale intensity values of the pixels in the tumor boundary.
- **texture_worst**: The "worst" or largest mean value of the gray-scale intensity values of the pixels in the tumor boundary among all nuclei.

- **perimeter_mean**: The average size of the tumor boundary.
- **perimeter_se**: The standard error of the size of the tumor boundary.
- **perimeter_worst**: The "worst" or largest mean value of the size of the tumor boundary among all nuclei.
- **area_mean**: The average area of the tumor.
- **area_se**: The standard error of the area of the tumor.
- **area_worst**: The "worst" or largest mean value of the area of the tumor among all nuclei.
- **smoothness_mean**: The average smoothness of the tumor boundary, which is a measure of local variations in radius lengths.
- **smoothness_se**: The standard error of the smoothness of the tumor boundary.
- **smoothness_worst**: The "worst" or largest mean value of the smoothness of the tumor boundary among all nuclei.
- **compactness_mean**: The average compactness of the tumor, which is a measure of the squared perimeter divided by the area.
- **compactness_se**: The standard error of the compactness of the tumor.
- **compactness_worst**: The "worst" or largest mean value of the compactness of the tumor among all nuclei.
- **concavity_mean**: The average severity of concave portions of the tumor boundary.
- **concavity_se**: The standard error of the severity of concave portions of the tumor boundary.
- **concavity_worst**: The "worst" or largest mean value of the severity of concave portions of the tumor boundary among all nuclei.
- **concave points_mean**: The average number of concave portions of the tumor boundary.
- **concave points_se**: The standard error of the number of concave portions of the tumor boundary.
- **concave points_worst**: The "worst" or largest mean value of the number of concave portions of the tumor boundary among all nuclei.
- **symmetry_mean**: The average symmetry of the tumor boundary.
- **symmetry_se**: The standard error of the symmetry of the tumor boundary.
- **symmetry_worst**: The "worst" or largest mean value of the symmetry of the tumor boundary among all nuclei.
- **fractal_dimension_mean**: The average fractal dimension of the tumor boundary, which captures the complexity of the boundary shape.
- **fractal_dimension_se**: The standard error of the fractal dimension of the tumor boundary.
- **fractal_dimension_worst**: The "worst" or largest mean value of the fractal dimension of the tumor boundary among all nuclei.

Table 1. Attributes and their Categories

Category	Attribute 1 (Mean)	Attribute 2 (SE)	Attribute 3 (Worst)
Radius	radius_mean	radius_se	radius_worst
Texture	texture_mean	texture_se	texture_worst
Perimeter	perimeter_mean	perimeter_se	perimeter_worst
Area	area_mean	area_se	area_worst
Smoothness	smoothness_mean	smoothness_se	smoothness_worst
Compactness	compactness_mean	compactness_se	compactness_worst
Concavity	concavity_mean	concavity_se	concavity_worst
Concave Points	concave points_mean	concave points_se	concave points_worst
Symmetry	symmetry_mean	symmetry_se	symmetry_worst
Fractal Dimension	fractal_dimension_mean	fractal_dimension_se	fractal_dimension_worst

Note: The attributes 'diagnosis' and 'id' are purposefully excluded from this table

3.1 Data Preprocessing

Firstly we remove the `id` attribute from our dataset as it is only a key value. Secondly, we modify the `diagnosis` attribute, mapping each 'M' value to 1 and each 'B' value to 0. Once this was complete, we used the `sklearn.preprocessing.StandardScalar` to perform z-score normalization on the dataset. This normalization prevents certain features from dominating the analysis or modeling process based solely on their larger values. It ensures that each feature contributes proportionally to the analysis and prevents biases that may arise due to different scales. Normalization also aids the optimization algorithm in converging with more haste to the optimal solution.

4 METHODOLOGY

We used `sklearn.model_selection.train_test_split` to split our dataset the dataset into two sets: 70% of our dataset was used for training, and the remaining 30% was used for testing. We used a `sklearn.neural_network.MLPClassifier` to generate our model, and used a randomizer to generate our initial weights and biases.

During our training and hyperparameter tuning step, we kept the batch size, learning rate, random state, and activation function fixed, which minimized the number of model permutations we had to work with.

We used the following heuristic to determine the optimal layering architecture of our MLP.

- Randomly generate a tuple of length $2 \leq n \leq 6$, with $10 \leq n_i \leq 100$. This tuple represents our neural network model, with the length of the tuple representing the number of layers in our model, and each sequential tuple element representing the number of nodes in each layer.
- Train and test a model with our generated shape.
- If the model performs better than a previous model, take this model to be our optimal shape. Otherwise, discard this model.

We performed 100 iterations of our heuristic to yield our optimal neural network architecture.

Upon choosing a model, we then study the learning curve generated during training to find signs of overfitting. If we found our model to be overfitting, we would tweak our batch size and alpha value, and repeat our training process.

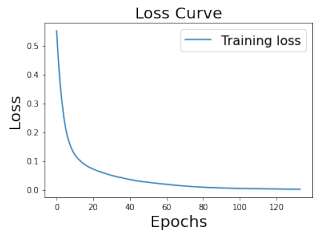
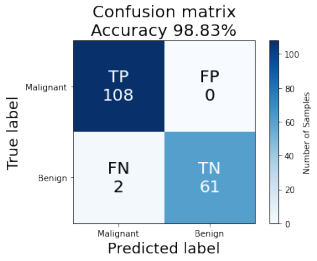
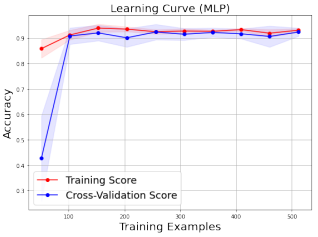
5 EXPERIMENTATION AND RESULTS

The loss curve serves as a visual representation of a machine learning model's performance throughout the training process. In the current scenario, the depicted curve demonstrates a continuous

Table 2. Model 1

Hyperparameter	Value
hidden_layer_sizes	(16, 32)
max_iter	1000
activation	'relu'
solver	'adam'
batch_size	32
random_state	42

Table 3. Graphs Associated with Model 1

Loss Curve	Confusion Matrix	Learning Curve
		

decrease in the model's loss (or error) over time. This trend indicates that the model's accuracy is progressively improving, leading to enhanced predictive capabilities with each epoch. Initially, the curve starts at approximately 0.7, signifying an initial accuracy of around 30% at the beginning of training. Subsequently, it converges to around 0.005, indicating a substantial enhancement in accuracy.

The learning curve plot above reveals that our model initially exhibits overfitting, as indicated by a significant gap between the training score and cross-validation score. However, as the training data size increases, this gap diminishes, implying a reduction in overfitting as more data is incorporated. Towards the end of the training process, the disparity between the training score and cross-validation score becomes negligible, accompanied by high accuracy, indicating that our model appropriately fits the dataset.

Having constructed a satisfactory model, our next objective was to evaluate its predictive capability using a minimal number of attribute categories. Instead of adopting an exhaustive approach involving the creation and optimization of ten models, we pursued an alternative methodology. Initially, we conducted linear regression on each attribute in relation to the diagnosis to determine the attributes with the highest correlation. The top five attributes, exhibiting the strongest correlation with predicting the diagnosis based on the highest R^2 values, were identified as `perimeter_mean`, `concave points_mean`, `radius_worst`, `perimeter_worst`, and `concave points_worst`. Subsequently, we grouped these attributes into their respective categories.

Among the attribute categories, `perimeter` and `concave` exhibited the highest frequency, appearing twice. We proceeded to build a model exclusively utilizing these perimeter and concave attributes to assess its impact on accuracy.

Upon considering only the perimeter and concave attributes, our accuracy experienced a slight decline from 98.83% to 97.66%. Notably, the learning curve plot did not exhibit signs of overfitting, suggesting a well-fit model. Despite the slight decline, the accuracy of 97.66% still exceeds performs

extraordinarily well, indicating a noteworthy achievement. To further explore this approach, we now aim to evaluate the performance by exclusively utilizing the perimeter *or* concave statistics.

Table 4. Graphs Associated with Training Model 1 Using Exclusively Perimeter and Concave Statistics

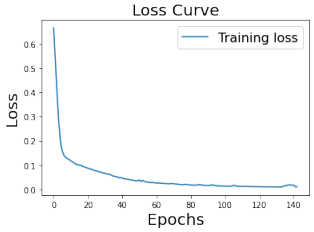
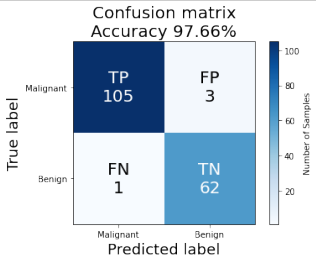
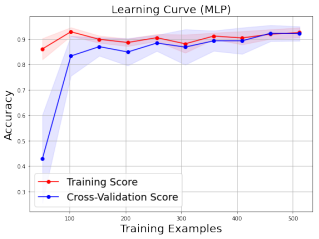
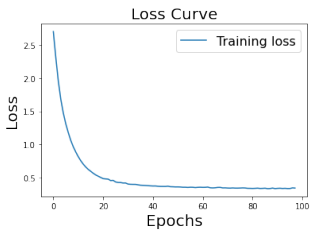
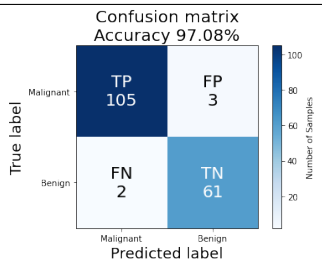
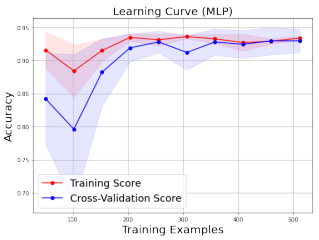
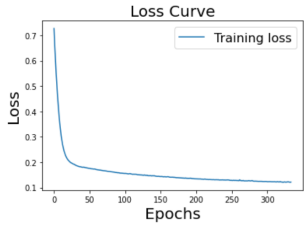
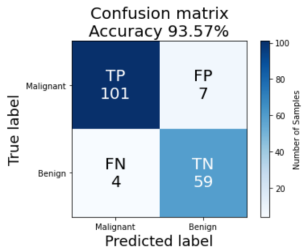
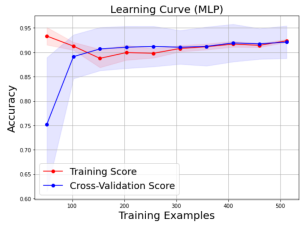
Loss Curve	Confusion Matrix	Learning Curve
		

Table 5. Graphs Associated with Training Model 1 Using Exclusively Perimeter Statistics

Loss Curve	Confusion Matrix	Learning Curve
		

By utilizing only the perimeter-related inputs, our model achieved an accuracy of 97.08%. Although the learning curve plot for this particular scenario does not exhibit the same level of smoothness as the other two cases, there are no indications of overfitting. This accuracy of 97.08% is comparable to that of the modified SVM model, yet it only relies on three inputs from a single input category. This outcome is highly promising as it demonstrates the possibility of attaining similar accuracy levels in our model using a significantly reduced feature set. Consequently, medical professionals may only need to obtain measurements from one or two categories rather than ten categories, potentially leading to cost reductions in the diagnostic process.

Table 6. Graphs Associated with Training Model 1 Using Exclusively Concave Statistics

Loss Curve	Confusion Matrix	Learning Curve
		

When considering only the concave-related inputs, a notable decrease in performance was observed. The model achieved an accuracy of only 93% and displayed no indications of overfitting. Although this accuracy level is still relatively high, it falls short compared to the perimeter model. Consequently, the concave model can be disregarded as it does not surpass the performance of the aforementioned approach.

6 CONCLUSION

Our developed multi-layer perceptron (MLP) model exhibits outstanding performance in accurately classifying tumor cells as benign or malignant. The model achieves an impressive accuracy of 98.83% without displaying any signs of overfitting. This high level of accuracy underscores the effectiveness and robustness of our MLP approach. By considering key geometric measurements, particularly those related to perimeter attributes, our model demonstrates the ability to make precise classifications, highlighting the importance of these specific metrics in tumor cell classification. The reliability and accuracy of our model position it as a valuable tool for medical professionals in diagnosing and distinguishing between benign and malignant tumor cells.

REFERENCES

[1] M.M Islam et al. 2020. Breast Cancer Prediction: A Comparative Study Using Machine Learning Techniques. *SN COMPUT. SCI.* 1, 290 (Sept. 2020), 290. https://doi.org/10.1007/s42979-020-00305-w

[2] Sungroh Yoon Seonwoo Min, Byunghan Lee. 2017. Deep Learning in Bioinformatics. *Briefings in Bioinformatics* 18, 5 (Sept. 2017), 851–869. https://doi.org/10.1093/bib/bbw068

[3] Zheng Rong Yang. 2004. Biological Applications of Support Vector Machines. *Briefings in Bioinformatics* 5, 4 (Dec. 2004), 328–338. https://doi.org/10.1093/bib/5.4.328

Received 14 May 2023