Machine Learning Models and Neural Networks for Hepatocellular carcinoma Microarray Diagnosis

Raymond Tian
Franklin Research Internship Program
July 2021

***Abstract*—This paper examines the effectiveness of using machine learning algorithms including a neural network on a hepatocellular carcinoma (HCC) microarray dataset. The dataset contained 358 samples and over 22,000 features (genes) and was pre-processed for ML applications. This dataset had been extensively used to compare the accuracy of various ML classification models. This paper proposes a new method; the utilization of ML models and neural networks could contribute to more affordable HCC diagnosis and could also be applied to other types of cancer. The discovery of a direct relationship between the best-performing genes and HCC demonstrated that machine learning could be a powerful tool for HCC diagnosis. The exceptional results produced by the neural network proved that neural networks could be the future for HCC diagnosis.**

***Index Terms*—machine learning, deep learning, hepatocellular carcinoma, diagnosis, microarray**

I. Introduction

Hepatocellular carcinoma (HCC) is the primary form of liver cancer responsible for the second-most cancer deaths with its high fatality rate [17]. Currently, HCC diagnosis commonly relies on a CT scan or an MRI [26]. However, CT scans' capability is limited for HCC smaller than 1 cm, and can hardly achieve sensitivity of 94% on HCC larger than 1cm. [6]. Properly trained machine learning (ML) models and neural networks (NN) consistently perform much better with sensitivity rates of up to 100%. ML analysis can be used to curate specific genes on microarrays that allow for more affordable and accurate HCC diagnosis.

ML in biology continues to become widely adopted as ML algorithms heighten the capabilities of technology on the scientific world with its diverse applications. ML has been utilized throughout numerous subfields of biology, such as neuroscience and genetics; for example, using ML to study neurological behavior for potential mind reading [24] or efficiently drawing conclusions from data that would take humans years of manual work [25]. ML is a branch of artificial intelligence that "learns" from data to accurately form predictions and discover patterns [18].

There are two types of ML algorithms—supervised learning and unsupervised learning. Supervised learning is when a model takes both the inputs and their respective output as training data, and uses that data to perform predictions on future inputs [8]. Conversely, unsupervised learning is where an algorithm separates the data into multiple classes or clusters by discovering patterns within the data that is inputted [12]. Supervised learning learns from inputted data to make predictions, while unsupervised learning determines patterns within inputted data.

Classification is an objective that supervised learning can accomplish where data points are classified into multiple different classes, with the amount of classes varying from two to infinity. Examples of classification include classifying a handwritten letter as a letter from [a to z], or even something as simple as determining if a value is positive or negative. Through classifying a dataset, new data can be plotted and its class can be accurately determined [4].

Proper HCC diagnosis requires a patient to be diagnosed as HCC positive or HCC negative—each represents a class (or output) of a sample. Since there are only two classes for HCC diagnosis, binary classification models using ML can be utilized for this problem.

ML models have been used on HCC microarray datasets in previous papers [29] with feature selection techniques ,such as maximum relevance with incremental feature selection to determine genes that are excellent for HCC diagnosis [30]. Not only does this paper include ML models, but an NN was used as well to determine which method proves most promising for HCC diagnosis.

The classification algorithms used in this paper are Support Vector Machines, Random Forest, K-Nearest Neighbors, and a feedforward neural network. This paper will include details on each model with their performance and its prevalence in affordable HCC diagnosis on a curated HCC dataset [10].

Similarly to how ML is a subset of artificial intelligence, deep learning (DL) is a subset of ML. While ML algorithms require human intervention to make proper adjustments, deep learning models can learn on their own to optimize the accuracy of their predictions. DL models rely on various NN that can learn independently through analyzing data [13]. A feedforward neural network (FFNN) was used on the dataset and its results were compared to the ML algorithms previously stated.

ML is revolutionizing the world with its benefits that range from efficiency and power to solving problems without human intervention. As the number of datasets for numerous diseases, such as HCC continue to grow, ML can take advantage of this data by applying ML through training models/NNs to more efficiently and accurately diagnose individuals for specific diseases relative to existing diagnostic procedures.

II. Methods

A. Preprocess

This paper used a hepatocellular carcinoma microarray dataset from a microarray cancer database [2] that was specifically curated for ML applications. The curation process for all of the databases' datasets include background correction, normalization, and unwanted probes extraction [2]. The dataset used for this paper contained 358 samples and 22,277 columns of gene expression data which would also be referred to as "features."

Applying ML algorithms for classification required a training set and testing set; the former was used for training the model, and the latter was used to test that model. The module scikit-learn [21] was used the split the data into a training set and testing set. Scikit-learn has numerous methods to split a dataset, such as train-test split and k-fold cross validation. Train-test split splits the dataset into a training set and testing set, while k-fold cross validiation splits a shuffled dataset into "k" groups, with one group used as the testing set while the other groups act as the training set, and this process is enumerated across all groups. The evaluation score can be averaged across each testing group and provides a more accurate evaluation compared to only performing train-test split [5]. After the data was split, 286 of the 358 samples made up the training set, with the remaining 72 samples as the testing set. The samples in each set varied each time a new split occurred.

Scikit-learn was also used to train and test the ML models with various algorithms along with Pytorch [20] to train the NN.

B. Feature Selection

Apart from just comparing scores between ML models and a NN on the original dataset, feature selection was also experimented with in an effort to minimize the number of genes that affected the class of each testing sample without a plummet in accuracy.

The main purpose of feature selection on this dataset was to conclude if a small microarray containing 100 genes was accurate enough to be a viable method of HCC diagnosis. This allows for further development of microarrays that can provide a cheaper alternative to CT scans and MRI for HCC screening. This method can be applied to other microarray datasets as well.

This process of feature selection was achieved by compiling the accuracy of each gene by testing each individual gene with an SVM (Rather than training a model that included all 22,277 genes, 22,277 models that contain one gene each were trained). The genes were sorted by their performance which was determined by the accuracy of their respective model. Afterwards, research was done on the best-performing genes (genes that have the best accuracy) and the worst-performing genes on BioGPS [28] to see if there was a clear correlation between a performance of a gene and their role in the presence of HCC.

The accuracy, ROC/AUC, precision, recall and F1-score were retrieved from using the ML models and NN on a smaller dataset that consisted of the top 10 best-performing genes, and were then compared the results with the models that contained the original 22,277 genes.

C. Support Vector Machines

Support vector machines (SVM) is an exceptional ML algorithm for classifying data. Given a n-dimensional space of points, SVM looks for an "optimal hyperplane" (decision boundary) that best splits the points into two classes [7]. Just like how a one-dimensional point divides a two-dimensional line into two parts, a hyperplane of the n-1 dimension divides a n-dimensional space into two parts [27].

An optimal decision boundary is present when the margin between a set of data points (support vectors) is greatest. The equation of an SVM hyperplane is

$$w^T x + b = 0$$

where w represents the normal vector to the hyperplane, and x represents a set of points where the distance between the hyperplane and x is maximized [7]. Thus the decision boundary is

$$g(x) = sign(w^T x + b)$$

The input's class is determined by the sign of the output [7].

SVM was one of the ML alrogithms experimented with and utilized due to its versatility along with the hypothesis that certain genes heavily influence the presence of HCC. SVM also works well with high dimensional spaces [14], which is optimal for the initial feature count of over 22,000. The final SVM was trained using the linear kernel because a tuned SVM model with the RBF kernel had equivalent results.

D. Random Forest

The random forest (RF) algorithm was the second ML algorithm used for HCC classification. As the "F" in RF stands for "forest," one can assume that this algorithm's structure is based on a forest, or multiple trees. In this case, the trees used by RF are decision trees (DT) [1], [22], another ML algorithm that can be used for classification.

The decision tree algorithm classifies data through split conditions; each split condition can be viewed as a branch splitting into two branches. Depending on the split condition and a data point's value, the data point will continue through a specified branch and will repeat the process of multiple split conditions until its class is determined. Optimally, the numerous split conditions contained with the DT separates the data points based on class. DT compares every possible split condition and chooses the split condition that offers the most information, which can be calculated through entropy. Even though DTs are powerful, they are also very prone to overfitting, especially when the sample size is relatively small [22].

RF greatly reduces the possibility of overfitting through utilizing multiple DTs. Its building steps consist of bootstrap and aggregating, or bagging, through building trees, each using random sampling with replacement and then using a subset of features on each tree. This process of bagging allows for less sensitivity to generalization as each tree is likely to be different from one another. Rather than having a single DT to predict a class from a data point, prediction with RF requires the data point to be passed through the entire forest of DTs, with the class being finally determined through majority rule [3].

RF was the second ML model used mainly to compare with SVM, and because it is a better algorithm than DT for the limited sample size of 358 as it is less likely to overfit. After experimenting with the number of trees the DT should consist of, 200 trees was the golden number that produced the best results.

E. K-Nearest Neighbors

K-nearest neighbors (KNN) was the third and final ML model used to compare with SVM and RF. KNN is another supervised learning classification algorithm that determines an input's class based on which class is more prominent among the input's "k"-nearest neighbors [15].

KNN permits the user to specify how many neighbors the model should contain. When a model receives an input, it takes the input's coordinates and using euclidean distance, finds the neighbors of the input, and returns the class that encapsulates a majority of the neighbors, and returns that class as the class of the input [15].

After experimenting with KNN, having only one neighbor yielded the most consistent results.

F. Neural Network

An (NN) was the last model used for training the HCC microarray dataset. Using Pytorch [20], an FFNN was constructed that consisted of three to four hidden layer of various sizes depending on the initial number of features, or input neurons. Refer to Figure 1 for the FFNN's structure with ten input neurons, made using NN-SVG [16].

An FFNN is the simplest NN architecture. It consists of an input layer, multiple hidden layers, and an output layer. Each layer consists of one or more neurons. The amount of neurons in the input layer is equivalent to the number of features in the dataset, while the amount of neurons in the output layer can be the number of classes, or in the case of binary classification (HCC diagnosis is a binary classification problem), one single neuron that was rounded to either 1 or 0, which represented the two possible classes of the samples. The neurons of each layer feeds into subsequent layers, with each connection having a unique weight and each neuron with the hidden layer having a bias. Every epoch, the NN tunes these weights and biases to minimize loss [11]. The loss function that produced the best results for the NN was BCEWithLogitsLoss, a combination of a sigmoid layer and binary cross-entropy loss [20]. Although the FFNN model was slightly overfit, a portion of it was due to the small sample size of the dataset. The following is a table that describes the layer sizes, number epochs, learning rate, and batch size of the FFNN models depending on the feature size (input layer size).

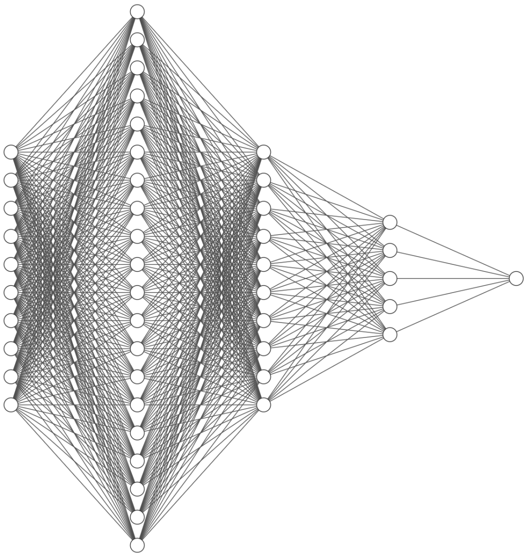

Fig. 1. Feedforward neural network structure with 10 input neurons

Feature size	22,277	10
Input layer size	22,277	10
Hidden layer 1 size	1000	10
Hidden layer 2 size	500	20
Hidden layer 3 size	100	10
Hidden layer 4 size	50	-
Output layer size	1	1
Epochs	700	10,000
Batch size	32	32
Learning rate	1.00E-04	1.00E-03

TABLE I
COMPARISION OF NN MODEL PARAMETERS

All the code mentioned can be found on my github: https://github.com/raymondytian/microarray-ml/.

III. RESULTS

The accuracy metrics that were used to compare the ML models and NN include ROC/AUC, confusion matrices, precision, recall, and F1-score. These accuracy metrics were also used to compare accuracy when training with 10 features versus 22,277 features.

The first accuracy metric to consider is the receiver operating characteristic and the area under its curve (ROC/AUC). The ROC curve consists of two axes—the x-axis represents the number of negatives incorrectly classified over total negatives (false positive rate); the y-axis (recall) represents the number of positives correctly classified over total positives (true positive rate/recall) [9]. Optimally, a model will have a false positive rate of 0 and a true positive rate of 1 (at least for cancer diagnosis). The following is a figure comparing the ROC/AUC score of the trained models (SVM, RF, KNN, NN) using all 22,277 genes as features.

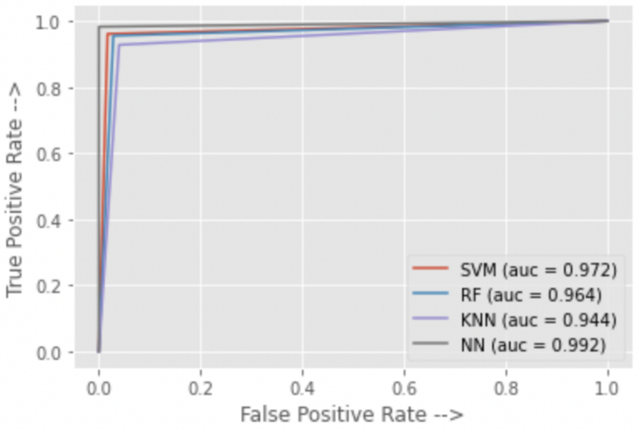

Fig. 2. ROC/AUC model comparison using 22,277 genes

When comparing with ROC/AUC, the NN outperformed the other ML algorithms by a large margin of at least 2.5%.

Analysis was also done with the 10 best performing genes and the 10 worst performing genes determined through feature selection.

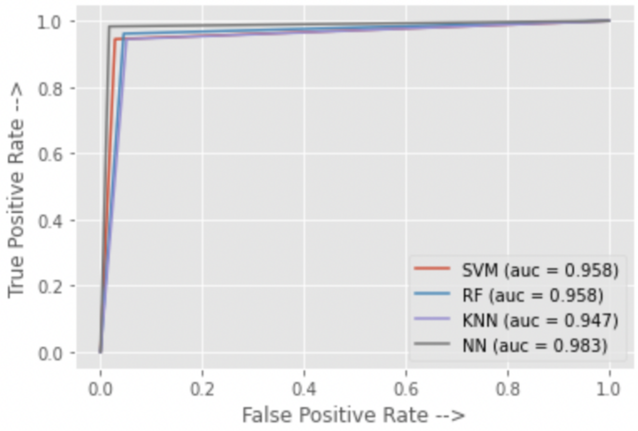

Fig. 3. ROC/AUC model comparison using best 10 genes

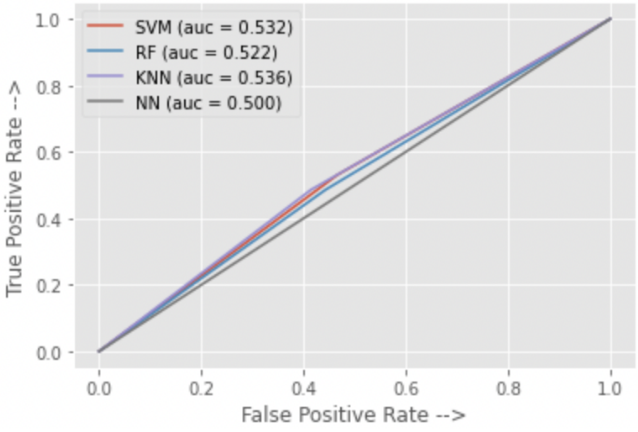

Fig. 4. ROC/AUC model comparison using worst 10 genes

From the above figures, it can be concluded that there is no significant drop-off in ROC/AUC score when using the best 10 genes compared to all 22,277 genes, but there is a massive drop off when using the worst 10 genes. Table 2 shows the best performing genes do indeed have a direct relationship with HCC. The results were obtained through using BioGPS [28].

Gene	Accuracy	Relation with HCC?
STAB2	0.986	yes
DAP3	0.986	yes
CCNBI	0.972	yes
KPNA2	0.972	yes
SMAD6	0.972	yes
...	...	...
ETV2	0.319	no
OVOL1	0.319	no
DCX	0.319	no
CDH5	0.319	no
SS18	0.319	no

TABLE II
CORRELATION BETWEEN GENE ACCURACY AND RELATION WITH HCC

The following figures are confusion matrices from each model trained with all 22,277 genes. The testing values are different because k-fold cross validation was only done on the ML algorithms. However, the accuracy measures can be comparable through precision, recall, and F1-score [23] (figures will be included).

The confusion matrix shows the number of test samples that were classified correctly or incorrectly for each class. 1 and 0 denote HCC positive and HCC negative respectively. Precision represents the number of positives correctly classified over number of samples classified positive. Recall represents the number of positives correctly classified over total positives. The F1-score encapsulates both the precision and recall by finding their harmonic mean.

Examine Fig. 5-8 for the confusion matrix of each model and Table III for their respective precision, recall, and F1-score.

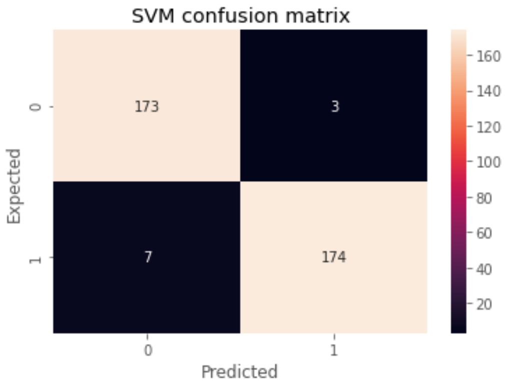

Fig. 5. Confusion matrix of SVM model

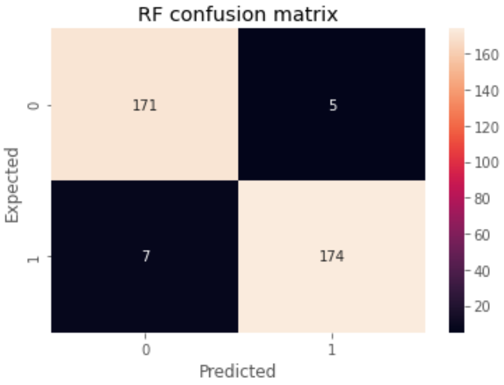

Fig. 6. Confusion matrix of RF model

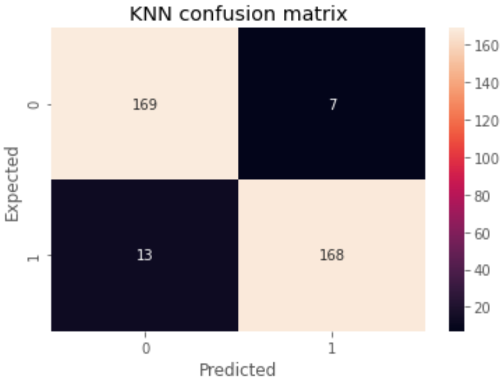

Fig. 7. Confusion matrix of KNN model

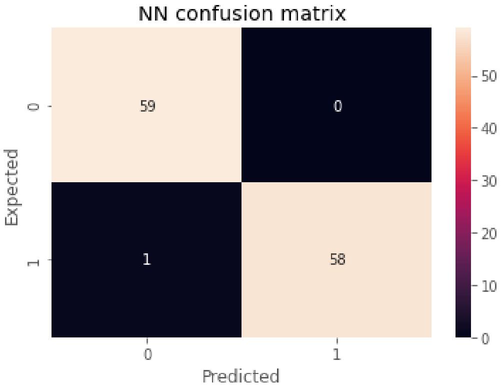

Fig. 8. Confusion matrix of NN model

Model	Precision	Recall	F1-score
SVM	0.98	0.96	0.97
RF	0.97	0.96	0.97
KNN	0.96	0.93	0.94
NN	1.00	0.99	0.99

TABLE III
COMPARISON OF PRECISION, RECALL, AND F1-SCORE BETWEEN MODELS

Through these results, it was discovered that the NN model greatly exceeded in performance compared to SVM, RF, and KNN (in order). With an increase in sample size, the NN model can perform even better and possible achieve perfect results.

IV. Discussion

Given the results, the models ranked in order of performance are as follows: NN, SVM, RF, and KNN. Each model has its own set of advantages that plays a role in its performance.

Beforehand, the limitations of the dataset should be considered. The dataset used contained only 358 samples, a relatively low number of samples for ML or NN application. On top of that, 20% of the dataset's samples was used for testing the model produced by the remaining 80% of the orignal dataset, or 286 samples. If a substantial increase in sample size is used for further training of ML and NN models, the models will certainly perform even better.

Advantages of SVM include being an exceptional algorithm for high dimensional spaces and for datasets with a feature size that exceed its sample size [14], both qualities of the HCC microarray dataset. However, like all other ML algorithms and NN, a low sample size makes SVM and other ML models prone to overfitting. RF improves this overfitting issue that would be more prominent if a decision tree model was used [3], but since the model is still rule-based, overfitting is not completely resolved. The RF model's performance is also due to its method of building trees with random sets of features, a perfect attribute for datasets with large feature sizes. KNN is the simplest of the ML models used in this paper, which is the main factor for its quick training speed [15] and a lack of performance compared to the other models. The NN likely outperformed the ML models due to the fundamental difference between ML and DL. ML includes set algorithms that are executed by a machine while DL are based around NNs that ”learn,” which allows it to maximize performance each iteration [19].

The remarkable results produced by the ML models (especially the FFNN) along with results from other papers [29], [30] prove that ML should be considered as the next step for HCC diagnosis. While the accuracy of the ML models matched with previous research, the nearly flawless performance of the FFNN serves promising as its simple to train and will only become better when more data is used for training and testing.

Further research could be done including use of various NN models atop of the current models in use, including a recurrent neural network, convolutional neural network, etc. These neural network architectures are primarily useful for imaging but their power certainly seem promising for a binary classification problem like HCC diagnosis. Different methods of feature selection, such as the ones used in previous papers [30] could have also been more optimal for determining the best genes to train models with for HCC diagnosis. Microarray datasets of various diseases can be trained with the same process to conclude if ML can act as a superior alternative to existing diagnostic procedures for a multitude of diseases.

V. Figures and Tables

Figures

Tables

References

[1] Jehad Ali, Rehanullah Khan, Nasir Ahmad, and Imran Maqsood. Random forests and decision trees. *International Journal of Computer Science Issues (IJCSI)*, 9(5):272, 2012.

[2] Tanya Barrett, Stephen E Wilhite, Pierre Ledoux, Carlos Evangelista, Irene F Kim, Maxim Tomashevsky, Kimberly A Marshall, Katherine H Phillippy, Patti M Sherman, Michelle Holko, et al. Ncbi geo: archive for functional genomics data sets—update. *Nucleic acids research*, 41(D1):D991–D995, 2012.

[3] Leo Breiman. Random forests. *Machine learning*, 45(1):5–32, 2001.

[4] Jason Brownlee. Supervised and unsupervised machine learning algorithms. *Machine Learning Mastery*, 16(03), 2016.

[5] Jason Brownlee. A gentle introduction to k-fold cross-validation, Aug 2020.

[6] Ye Ra Choi, Jin Wook Chung, Mi Hye Yu, Myungsu Lee, and Jung Hoon Kim. Diagnostic accuracy of contrast-enhanced dynamic ct for small hypervascular hepatocellular carcinoma and assessment of dynamic enhancement patterns: Results of two-year follow-up using cone-beam ct hepatic arteriography. *PloS one*, 13(9):e0203940, 2018.

[7] Corinna Cortes and Vladimir Vapnik. Support-vector networks. *Machine learning*, 20(3):273–297, 1995.

[8] Ciro Donalek. Supervised and unsupervised learning. In *Astronomy Colloquia. USA*, volume 27, 2011.

[9] Tom Fawcett. An introduction to roc analysis. *Pattern recognition letters*, 27(8):861–874, 2006.

[10] Bruno César Feltes, Eduardo Bassani Chandelier, Bruno Iochins Grisci, and Márcio Dorn. Cumida: An extensively curated microarray database for benchmarking and testing of machine learning approaches in cancer research. *Journal of Computational Biology*, 26(4):376–386, 2019. PMID: 30789283.

[11] Terrence L Fine. *Feedforward neural network methodology*. Springer Science & Business Media, 2006.

[12] Zoubin Ghahramani. Unsupervised learning. In *Summer School on Machine Learning*, pages 72–112. Springer, 2003.

[13] Brett Grossfeld and Associate content marketing manager. Deep learning vs. machine learning: What's the difference?

[14] D Kumar. Top 4 advantages and disadvantages of support vector machine or svm, 2019.

[15] Jorma Laaksonen and Erkki Oja. Classification with learning k-nearest neighbors. In *Proceedings of International Conference on Neural Networks (ICNN'96)*, volume 3, pages 1480–1483. IEEE, 1996.

[16] Alexander LeNail. Nn-svg.

[17] Sahil Mittal and Hashem B El-Serag. Epidemiology of hcc: consider the population. *Journal of clinical gastroenterology*, 47:S2, 2013.

[18] Mehryar Mohri, Afshin Rostamizadeh, and Ameet Talwalkar. *Foundations of machine learning*. MIT press, 2018.

[19] Kevin Parrish. Deep learning vs. machine learning: what's the difference between the two. *Online: https://www. digitaltrends. com/cool-tech/deep-learning-vs-machine-learning-explained/2/[Last accessed: 08.05. 2018]*, 2018.

[20] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. Pytorch: An imperative style, high-performance deep learning library. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems 32*, pages 8024–8035. Curran Associates, Inc., 2019.

[21] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

[22] J. Ross Quinlan. Induction of decision trees. *Machine learning*, 1(1):81–106, 1986.

[23] Takaya Saito and Marc Rehmsmeier. Basic evaluation measures from the confusion matrix. *Access link: https://classeval. wordpress. com/introduction/basic-evaluation-measures*, 2017.

[24] Neil Savage. How ai and neuroscience drive each other forwards. *Nature*, 571(7766):S15–S15, 2019.

[25] Adi L Tarca, Vincent J Carey, Xue-wen Chen, Roberto Romero, and Sorin Drăghici. Machine learning and its applications to biology. *PLOS Computational Biology*, 3(6):e116, 2007.

[26] Melanie B Thomas, Deborah Jaffe, Michael M Choti, Jacques Belghiti, Steven Curley, Yuman Fong, Gregory Gores, Robert Kerlan, Phillipe Merle, Bert O'Neil, et al. Hepatocellular carcinoma: consensus recommendations of the national cancer institute clinical trials planning meeting. *Journal of clinical oncology*, 28(25):3994, 2010.

[27] Eric W Weisstein. Hyperplane.

[28] Chunlei Wu, Camilo Orozco, Jason Boyer, Marc Leglise, James Goodale, Serge Batalov, Christopher L Hodge, James Haase, Jeff Janes, Jon W Huss, et al. Biogps: an extensible and customizable portal for querying and organizing gene annotation resources. *Genome biology*, 10(11):1–8, 2009.

[29] Qing-Hai Ye, Lun-Xiu Qin, Marshonna Forgues, Ping He, Jin Woo Kim, Amy C Peng, Richard Simon, Yan Li, Ana I Robles, Yidong Chen, et al. Predicting hepatitis b virus–positive metastatic hepatocellular carcinomas using gene expression profiling and supervised machine learning. *Nature medicine*, 9(4):416–423, 2003.

[30] Zi-Mei Zhang, Jiu-Xin Tan, Fang Wang, Fu-Ying Dao, Zhao-Yue Zhang, and Hao Lin. Early diagnosis of hepatocellular carcinoma using machine learning method. *Frontiers in bioengineering and biotechnology*, 8:254, 2020.