Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 29 June 2022 doi:10.20944/preprints202206.0394.v1

Improving Clinical Prediction of Later Occurrence of Breast Cancer Metastasis Using Deep Learning and Machine Learning with Grid Search

Xia Jiang, Chuhan Xu
Department of Biomedical Informatics
University of Pittsburgh
Pittsburgh, PA

Corresponding Author: Xia Jiang

Mailing Address: Department of Biomedical Informatics
University of Pittsburgh School of Medicine
5607 Baum Blvd, Pittsburgh, PA 15217

Email: xij6@pitt.edu
Phone: 412-648-9310
Chuhan Xu: chx@pitt.edu

ABSTRACT

Background

It is important to be able to predict, for each individual patient, the likelihood of later metastatic occurrence, because the prediction can guide treatment plans tailored to a specific patient to prevent metastasis and to help avoid under- or over-treatment. Deep Neural Network (DNN) learning, commonly referred to as deep learning, has become popular due to its success in image detection and prediction, but questions such as whether deep learning outperforms other machine learning methods when using non-image clinical data remain unanswered. Grid search has been introduced to deep learning hyperparameter tunning for the purpose of improving its prediction performance, but the effect of grid search on other machine learning methods are under-studied. In this research, we take the empirical approach to study the performance of deep learning and other machine learning methods when using non-image clinical data to predict the occurrence of breast cancer metastasis (BCM) 5, 10, or 15-years after the initial treatment. We developed DNN models as well as models using 9 other machine learning methods including Naive Bayes (NB), Logistic Regression (LR), Support Vector Machine (SVM), LASSO, Decision Tree (DT), k-Nearest Neighbors (KNN), Random Forrest (RF), AdaBoost (ADB), and XGBoost (XGB). We used grid search to tune hyperparameters for all methods. We then compared the deep learning models to the models trained using the 9 other machine learning methods.

© 2022 by the author(s). Distributed under a Creative Commons CC BY license.

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 29 June 2022 doi:10.20944/preprints202206.0394.v1

Results

Based on the mean test AUC results, DNN ranks 6th, 4th, and 3rd when predicting 5-year, 10-year, and 15-year BCM respectively, out of 10 machine learning methods. The top performing methods in predicting 5-year BCM are XGB(1st), RF(2nd), and KNN(3rd). For predicting 10-year BCM the top performers are XGB (1st), RF(2nd), and NB(3rd) . Finally, for 15-year BCM the top performers are SVM (1st), LR and LASSO (tied for 2nd), and DNN (3rd). The ensemble methods RF and XGB outperform other methods when data are less balanced, while SVM, LR, LASSO, and DNN outperform other methods when data are more balanced. Our statistical testing results show that at a significance level of 0.05 DNN overall performs no worse than other machine learning methods when predicting 5-year, 10-year, and 15-year BCM.

Conclusions

Our results show that deep learning with grid search overall performs at least as well as other machine learning methods when using non-image clinical data. It is interesting to note that some of the other machine learning methods such as XGB, RF, and SVM are very strong competitors of DNN when incorporating grid search. It is also worth noting that the computation time required to do grid search with DNN is way more than that required to do grid search with the other 9 machine learning methods.

Keywords: deep learning, DNN, machine learning; breast cancer, metastasis; metastatic breast cancer; distant recurrence of breast cancer metastasis; prediction, clinical, EHR

BACKGROUND

In 2020, female breast cancer has surpassed lung cancer as the most commonly diagnosed cancer worldwide, with an estimated 2.3 million new cases in 2020 [1]. Breast cancer remains one of main cancer-related causes of death in women globally [2] and was responsible for 685,000 deaths worldwide in 2020 [1]. Breast cancer is the second leading cause of cancer death among US women after lung cancer, estimated to account for 43,600 deaths in 2021 [3, 5]. It is the number one cause of cancer-related deaths for US women aged 20 to 59 [6].

Women rarely die of breast cancer confined to the breast or draining lymph nodes; rather, they die mainly due to metastasis, a condition in which cancer spreads to other vital organs, such as the lung and brain. *Metastatic breast cancer* (*MBC*) is the cause of over 90% of breast cancer related deaths [7] and remains a largely incurable disease. Although most newly diagnosed breast cancer cases are not metastatic, all patients are at risk of developing metastatic cancer in the future, even if they are free of cancer for years after the initial treatment. The ability to effectively predict, for each individual patient, the likelihood of later metastatic occurrence is important, because the prediction can guide treatment plans tailored to a specific patient to prevent metastasis and to help avoid under- or over-treatment.

Clinicians face uncertainty in determining the ideal treatment course for individual patients with breast cancer. For example, image-guided core needle biopsy of the breast is a common procedure that can return non-definitive results in 5% to 15% of women. In these cases it is difficult to determine the subtype of the breast cancer. Variation in breast cancer subtypes has been known to be associated with a patient's drug response, progression of the tumor, and survival of the patient

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 29 June 2022 doi:10.20944/preprints202206.0394.v1

[8, 9]. There can also be significant uncertainty about the treatment and prognosis for breast cancer. For example, HER2-amplified breast cancer is a subtype with poor prognosis if untreated, but the targeted therapeutic trastuzumab (Herceptin) has vastly improved the survival rate of such patients. Although Herceptin is used in the therapy of all patients with HER2-amplified tumors, only some respond. Furthermore, it is expensive and can cause cardiac toxicity [10,11]. Therefore, it is important to limit its usage to patients who are likely to benefit from it. Furthermore, histology alone does not predict long term outcome well, as most breast cancers are considered localized to the breast at the time of diagnosis, with most of these patients 'cured' upon excision. Still, up to one third of these patients will suffer distant recurrences, often after many years [12]. As treatments are toxic, clinical decisions need to account for prognostic predictors of outcome.

Various learning methods have been developed and applied in biomedical prediction [13-19]. For instance, machine learning and language processing have been used to identify breast cancer local recurrence [13]. A logistic regression model was developed for cancer classification and prediction [14]. Various machine learning methods were used for predicting ubiquitination sites by training models from physicochemical properties of protein sequences data [15]. Bayesian network learning was used to model miRNA-mRNA interactions that cause phenotypic abnormality in breast cancer patients [16]. The risk prediction of prostate cancer recurrence was investigated through regularized rank estimation in partly linear AFT (Accelerated Failure Time) models using high-dimensional gene and clinical data [17]. An automatically derived class predictor was presented to determine the class of new leukemia cases based on gene expression monitoring by DNA micro-arrays [18]. An effective hybrid approach for selecting marker genes was developed for phenotype classification using micro-array gene expression data [19].

A Neural Network (NN) is one of the machine learning methods that can be used to conduct prediction and classification. A NN consists of layers of artificial neurons, also called nodes, mimicking structurally in a sense the impulse propagation mechanism in the human nervous system [20, 21] , so it is also called an Artificial Neural Network (ANN). ANNs can be used for unsupervised learning on unlabeled data or supervised learning on labeled data. Deep learning [22-24] is the use of Neural Networks composed of more than one hidden layer, which are also referred to as Deep Neural Networks (DNNs).

Artificial Neural Networks (ANNs), including DNNs, are widely used in science and information technology due to their notable properties including parallelism, distributed storage, and adaptive self-learning capability [25-30]. They have also been used in health care including cancer diagnosis and prediction. For example, an ANN was developed to help diagnose breast cancer based on the age of the patient, mass shape, mass border, and mass density; it achieved high predictive accuracy [30]. A noise-injected neural network was designed for breast cancer diagnosis and prognosis using expression data [29]. A hybrid neural network and genetic algorithm method was applied to breast cancer detection [27]. In another study, an ANN was used to reduce the number of gene signatures for the classification of breast cancer patients and the prediction of clinical outcomes, including the capability to accurately predict breast cancer metastases [26]. The DNN has obtained significant success in commercialized applications, such as voice and pattern recognition, computer vision, and image processing [28, 31-36]. However, its power has not been fully explored or demonstrated in clinical applications, such as the prediction of breast cancer metastasis (BCM). This is because the sheer magnitude of the number of variables involved in these problems presents formidable computational and modeling challenges [37].

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 29 June 2022 doi:10.20944/preprints202206.0394.v1

Precision medicine promises to help us improve patient outcomes by tailoring healthcare to the individual patient [38]. The electronic health record (EHR), a widely available data resource, has been underutilized for the purpose of tailoring therapies and providing prognostic information. An EHR database contains abundant data about patients' clinical features, disease status, interventions, and clinical outcomes, affording us the opportunity to provide highly-personalized medicine beyond only looking at the genomic level. It is believed that, "coupled with new analytics tools, they open the door to mining information for the most effective outcomes across large populations" [10]. Such data are invaluable to tailoring diagnosis and prognoses to individual with diseases such as breast cancer.

LSDS (Lynn Sage Dataset) was a de-identified and publicly available clinical dataset about breast cancer that was created and published via previous studies [39, 40]. It was curated using clinical data from the Lynn Sage Database (LSDB) hosted at Lynn Sage Comprehensive Breast Center at Northwestern Memorial Hospital and the EHR data hosted at The Northwestern Medicine Enterprise Data Warehouse (NMEDW) Northwestern University Feinberg School of Medicine and Northwestern Memorial HealthCare. The LSDS consists of records on 6726 breast cancer patients, which span 03/02/1990 to 07/28/2015. The dataset contains 61 patient features, including breast cancer metastasis and its follow-up. Three LSM (LSDS for Metastasis) datasets were retrieved from LSDS, which focus on 5, 10, and 15-year BCM status respectively [39,40]. A detailed description of the three LSM datasets are presented in the Methods section.

In this research, we took the empirical approach to study the performance of deep learning when predicting BCM using clinical data. We applied DNN to learn MBC prediction models from LSM datasets. These models can be used to predict 5, 10, and 15-year BCM. The performance of a DNN model is affected by the number of hidden layers and number of nodes per hidden layer, which are called hyperparameters. In addition, there are other hyperparameters that can be used to adjust the prediction performance of deep learning. For example, the number of epochs is a hyperparameter we consider. It is the number of times in which a deep learning model is trained by each of the training set samples exactly one. The learning might not converge when the number of epochs is too low, and model overfitting tends to get severe when it is too high. Tuning hyperparameters is the process of identifying the set of parameter values that are expected to produce the best prediction model from all sets of hyperparameter values examined. Grid search is designed to conduct hyperparameter tunning in a systematic way by going through a possible set of hyperparameter values automatically during learning. In this study, we optimized DNN model performance by conducting hyperparameter tunning via grid search.

To evaluate the performance of DNN, we compared our DNN models with the ones that we trained using 9 other well-known machine learning methods. We applied hyperparameter tunning and grid search to optimize model performance for each of the 9 comparison methods. We conjectured that deep learning with grid search would perform no worse than the comparison methods when predicting the binary status of BCM. We posit this conjecture because deep learning is a very powerful tool for prediction and has been successful in other applications such as image recognition [36, 41-49]. In this study we use feed-forward DNN models to predict 5, 10, 15-year BCM by learning from non-image clinical EHR data. Through literature searching we found some deep learning related studies that use image data to predict BCM [42-46] . But we haven't found a study that resembles ours.

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 29 June 2022 doi:10.20944/preprints202206.0394.v1

METHODS

Datasets

In this study, we used three LSM datasets about breast cancer metastasis: LSM-5Year, LSM-10Year, and LSM-15Year. Metastatic case counts of each of the three datasets are shown in Table 1. Each of the three datasets contains 32 variables: 31 predictors and the target variable "metastasis." Using LSM-5Year as an example, as described in [40], the value "yes" was assigned to "metastasis" if the patient metastasized within 5 years of initial diagnosis, the value "no" to "metastasis" if it was known that the patient did not metastasize within 5 years. The 31 predictors are defined in Table 2. Our objective was to learn and optimize prediction models from LSM datasets using DNN and 9 other machine learning methods, and then to compare the performance of these models.

Table 1. Case counts of the LSM datasets

	Total # of cases	# Positive cases	# Negative cases
LSM-5year	4189	437	3752
LSM-10year	1827	572	1255
LSM-15year	751	608	143

Table 2 The variables of the LSM datasets

Variables included	Description	Values
race	race of patient	white, black, Asian, American Indian or Alaskan native, native Hawaiian or other Pacific islander
ethnicity	ethnicity of patient	not Hispanic, Hispanic
smoking	smoking history of patient	ex smoker, non smoker, cigarettes, chewing tobacco, cigar
alcohol usage	alcohol usage of patient	moderate, no use, use but nos (non otherwise specified), former user, heavy user
family history	family history of cancer	cancer, no cancer, breast cancer, other cancer, cancer but nos
age_at_diagnosis	age at diagnosis of the disease	0-49, 50-69, >69
menopausal_status	inferred menopausal status	pre, post
side	side of tumor	left, right
TNEG	triple negative status in terms of patient being ER, PR, and HER2 negative	yes, no

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 29 June 2022 doi:10.20944/preprints202206.0394.v1

ER	estrogen receptor expression	neg, pos, low pos
ER_percent	percent of cell stain pos for ER receptors	0-20, 20-90, 90-100
PR	progesterone receptor expression	neg, pos, low pos
PR_percent	percent of cell stain pos for PR receptors	0-20, 20-90, 90-100
P53	whether P53 is mutated	neg, pos, low pos
HER2	HER2 expression	neg, pos
t_tnm_stage	prime tumor stage in TNM system	0, 1,2,3,4, IS, 1mic, X
n_tnm_stage	# of nearby cancerous lymph nodes	0,1,2,3,4,X
stage	composite of size and # positive nodes	0,1,2,3
lymph_nodes_removed	number of lymph nodes removed	0-11, 12-22, > 22
lymph_nodes_positive	number of positive lymph nodes	0, 1-8, >8
lymph_node_status	patient had any positive lymph nodes	neg, pos
histology	tumor histology	lobular, duct
size	size of tumor in mm	0-32, 32-70, >70
grade	grade of disease	1, 2, 3
invasive	whether tumor is invasive	yes, no
histology2	tumor histology subtypes	IDC, DCIS, ILC, NC
invasive_tumor_location	where invasive tumor is located	mixed duct and lobular, duct, lobular, none
DCIS_level	type of ductal carcinoma in situ	solid, apocrine, cribriform, dcis, comedo, papillary, micropapillary
re_excision	removal of an additional margin of tissue	yes, no
surgical_margins	whether residual tumor	res. tumor, no res. tumor, no primary site surgery
MRIs_60_surgery	MRIs within 60 days of surgery	yes, no

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 29 June 2022 doi:10.20944/preprints202206.0394.v1

Feedforward Neural Networks

Our DNNs are fully-connected feedforward neural networks composed of more than one hidden layer. Figure 1 shows the general structure of a feed-forward deep neural network that contains n hidden layers and an output layer that has two nodes. The inputs to the neural network are the observed values of the predictor variables in the dataset, while the outputs are the values of the target variable. In this research, we have 31 predictor variables so m, the number of nodes in our input layer, is equal to 31. X_0 represents the node for the bias passing from the input layer to the first hidden layer. The activation function $f(x)$ of a node determines the value to be passed to the next node based on the value of the current node x. We used a rectifier linear unit (ReLU) [37, 51], in which $f(x) = \max(0, x)$ as the activation function in the input layer. Since our datasets only contain positive values, by using ReLU as the activation function, all input values to our neural network model are directly passed to the hidden layers. In figure 1, the first hidden layer has p hidden nodes, the second hidden layer has q hidden nodes, and the nth hidden layer has r hidden nodes, indicating each hidden layer is allowed to have a different number of hidden nodes. We used ReLU as the activation function in each of the hidden layer(s) to avoid the vanishing gradient problem [37, 51]. $w_{ij}^{[1]}(i = 0, 2, \dots, \mathrm{m};\ j = 1, 2, \cdots, \mathrm{p})$ represents the connecting weights between the input layer and the first hidden layer, $w_{kl}^{[2]}(k = 0, 1, 2, \cdots, p; l = 1, 2, \cdots, q)$ represents the connecting weights between the first hidden layer and the second hidden layer, and $w_{st}^{[n+1]}(s = 0, 2, \cdots, r; t = 1, 2)$ represents the connecting weights between the last hidden layer and the output layer. n is the number of hidden layers. $b^{[1]}{}_j(j = 1, 2, \cdots, p)$ represents the biases of the nodes in the first hidden layer, $b^{[2]}{}_l(l = 1, 2, \cdots, q)$ represents the biases of the nodes in the second hidden layer, and $b^{[n+1]}{}_o(o = 1, 2)$ represents the biases of the nodes in the output layer. We have two nodes in the output layer, one for each target value. Recall that "metastasis" is our binary target variable, which has two values: "yes" or "no". We used the binary cross-entropy loss function, and sigmoid activation function in the output layer [37, 51]. In this study, the initial values of weights and bias are provided by the he_normal [54] weight initializer. He_normal draws samples from a truncated normal distribution centered on 0 with $\mathrm{stddev} = \mathrm{sqrt}(2 / \mathrm{num\_in})$ where num_in is the number of nodes in a layer. Tensorflow is an open-source library widely used for developing deep learning models. Keras is a high-level neural network API built on top of Tensorflow [52, 53]. Our DNN model learner was coded in Python and implemented using the Keras and Tensorflow packages.

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 29 June 2022 doi:10.20944/preprints202206.0394.v1

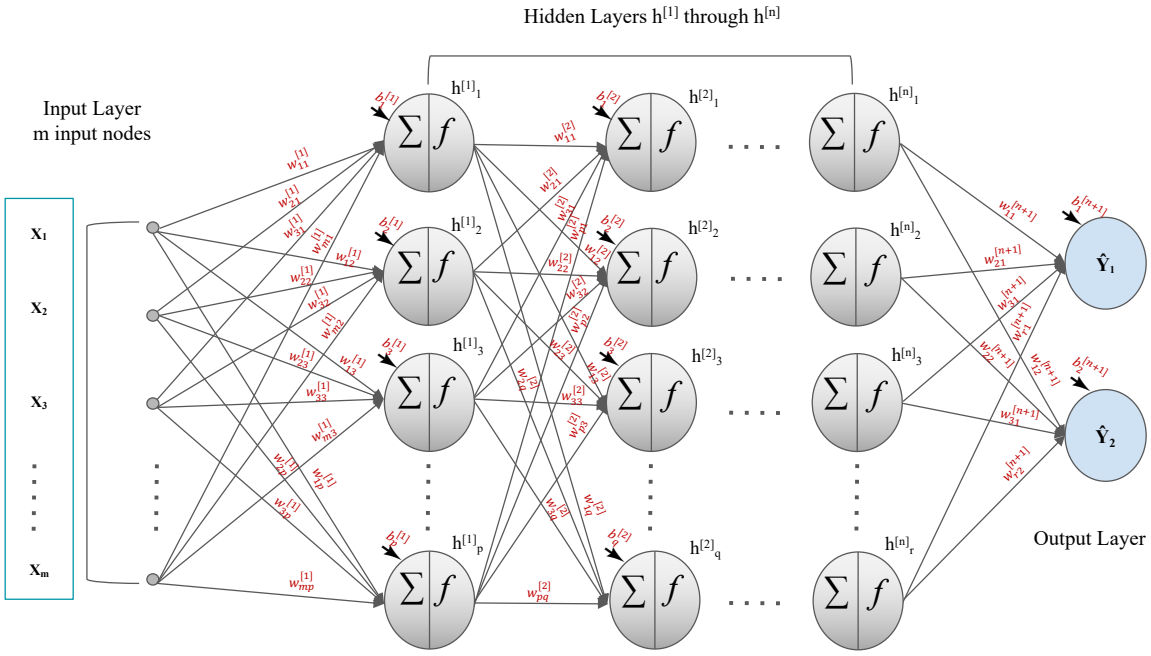

Figure 1. A feed-forward ANN that contains n hidden layers.

Hyperparameter Tuning with Grid Search

Deep learning is a powerful machine learning method due to its large number of hyperparameters that can be optimized [41]. See Table 3 for the hyperparameters and their values that we tested when training our DNN models. The number of hidden layers and number of hidden nodes are structural hyperparameters that greatly affect model performance, each of which can assume numerous different values. SGD (Stochastic Gradient Descent) and Adagrad (Adaptive Gradient Descent) are two commonly used optimizers. SGD adjusts its learning rate via momentum and decay, the two other hyperparameters that can be tuned during training. Adagrad adapts the learning rate to the parameters, conducting smaller-step updates for parameters linked to frequently appearing features, and larger-step updates for parameters linked to less frequent features. The learning rate is a hyperparameter that governs how big of a step it takes each time to update the internal model parameters (weights and biases) in response to the estimated error during the model training process. It is used by both the SGD and Adagrad. The momentum, a moving average of the gradients, is integrated in SGD to help accelerate the convergence of training. The decay is an iteration-based decay factor that can be used to decrease learning rate in each epoch during the optimization process. It is a hyperparameter incorporated in both SGD and Adagrad to help optimize model performance. The batch_size is also a hyperparameter in Keras, which controls the number of the training samples that are "fed" to the neural network before internal model parameters are updated. Other hyperparameters including epochs, dropout rate, L1, and L2 will be discussed in the "Overfitting" subsection below.

Hyperparameter Tunning is the process of identifying the set of hyperparameter values that is expected to produce the best prediction model from all sets of hyperparameter values being examined. Hyperparameter tuning gives us the power to optimize model performance but tuning

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 29 June 2022 doi:10.20944/preprints202206.0394.v1

a large number of hyperparameters presents a major challenge in terms of computation time [37, 55, 56]. Grid search is designed to conduct hyperparameter tunning in a systematic way by going through each of the sets of hyperparameter values automatically during the model training process. In our study, we tried to improve model performance by conducting grid search implemented in python using the scikit-learn package [52, 53]. In a grid search, each of the hyperparameters is given a series of values, the program will then iterate through every hyperparameter value combination possible to train models. We call a hyperparameter value combination a hyperparameter setting. We conducted grid search many times, each time focusing on giving a set of values to each of the hyperparameters. The range of values for a hyperparameter are predetermined in various ways such as preliminary experiments, literal searching, and computation resource and time limitation that we have. For example, we decided to focus on checking up to 4 hidden layer models, because we found that further extending number of hidden layers takes up too much computing power but with overall worse results based on some preliminary experiments we conducted. So the deepest model we trained contains 6 layers counting the input and output layer. Another example, model performance normally becomes worse once number of epochs exceeds 800 based on our preliminary experiments, so we set the maximum number of epochs to be 800. In each grid search, we randomly chose a set of values from the range of values for each of the hyperparameters (Table 3) based on the maximum number of hyperparameter settings that we can handle in reasonable time.

Overfitting

Overfitting is a phenomenon in which the model performs well on training data but generalizes poorly to unseen data [57–59]. Overfitting occurs when the model is complex and has a large number of parameters, such as in a DNN model, but insufficient data to accurately capture the underlying relationships between the variables. Overfitting is a common problem in machine learning, and it is overwhelmingly discussed in deep learning due to its significant effect on the performance of DNN models. A google search using "overfitting in deep learning" identified 280,000 articles published between 2015 and 2020. This is not only because we are dealing with a large set of hyperparameters in deep learning, but also because the number of internal parameters increases dramatically as the number of hidden layers and the number of hidden nodes per layer increase.

It is not possible to completely eliminate overfitting, but we took multiple approaches to minimize the effect of overfitting. First, we tuned "dropout rate" and "epochs" to reduce the effect of overfitting [41]. The "dropout" is a hyperparameter with which neurons are randomly dropped out during training to reduce time cost and minimize model overfitting. The number of epochs is a hyperparameter that helps balance model convergence and overfitting. It defines the number of times that the entire training data are used by the learning algorithm during training. One epoch means every sample in the training set has been used exactly once to update the internal model parameters. Secondly, we tunned regularization hyperparameters L1 and L2 to reduce overfitting. L1, a factor associated with LASSO regularization, can be used to remove the effect of the "noisy" input nodes and make the network less complex. L1 is also called a sparsity regularization factor. L2 is a regularization factor based on weight-based decay, which penalizes large weights to adjust the weight updating step during model training. We also introduced another parameter named as "L1OrL2", with which we can choose to tune L1 alone, L2 alone, or L1 and L2 simultaneously in a grid search. Finally, we used percent_auc_diff to quantify and keep track of the overfitting of a model. The percent_auc_diff is an output parameter in our grid search procedure, which represents

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 29 June 2022 doi:10.20944/preprints202206.0394.v1

the percent difference between mean train AUC and mean test AUC. When we selected the best DNN models, we not only considered the mean test AUC values, but also made sure the percent_auc_diff, was less than 5%.

Table 3: Description of the DNN hyperparameters and Their Values Tested

Hyperparameter	Description	Values
# of Hidden Layers	The depth of a DNN	1,2,3,4
# of Hidden Nodes	Number of neurons in a hidden layer	10,20, …,70 75 80 90 … 120, 200, 300 … 1100
Optimizer	Optimizes internal model parameters towards minimizing the loss	SGD, Adagrad
Learning rate	Used by both SGD and Adagrad	0.001 to 0.3, step size: 0.001
Momentum	Smooths out the curve of gradients by moving average. Used by SGD.	0, 0.4, 0.5, 0.9
Iteration-based Decay	Iteration-based decay; updating learning rate by a decreasing factor in each epoch	0 0.0001, 0.0002, …., 0.001, 0.002, … 0.01
Dropout rate	Manage overfitting and training time by randomly selects nodes to ignore	0, 0.4, 0.5
Epochs	Number of times model is trained by each of the training set samples exactly one	20, 30, 50, 80, 100, 200, …, 800
Batch_size	Unit number of samples fed to the optimizer before updating weights	1, 10, 20, …, 100
L1	Sparsity regularization;	0, 0.0005, 0.0008, 0.001, 0.002, 0.005, 0.008, 0.01, 0.02, 0.05, 0, 0.1, 0.2, 0.5
L2	Weight decay regularization; it penalizes large weights to adjust the weight updating step	0, 0.0005, 0.0008, 0.001, 0.002, 0.005, 0.008, 0.01, 0.02, 0.05, 0, 0.1, 0.2, 0.5
L1ORL2	Using L1 and L2 combinations to regularize overfitting;	L1 only, L2 only, L1 and L2

Performance Metrics and 5-fold Cross Validation

We designed an output format for grid search and recorded 64 different output values for each of the models trained in a grid search. Among the output values are information about the computer system used, computation time, and measures for model performance. For a given binary diagnostic test, a receiver operator characteristic (ROC) curve plots the true positive rate against the false positive rate for all possible cutoff values. The area under a ROC curve (AUC) measures the discrimination performance of a model. We conducted a 5-fold cross validation to

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 29 June 2022 doi:10.20944/preprints202206.0394.v1

train and evaluate each model in a grid search. The entire dataset was split evenly into 5 portions. The division was mostly done randomly except that each portion had approximately 20% of the positive cases and 20% of the negative cases to ensure that it was a representative fraction of the dataset. Training and testing were repeated five times. Each time, a unique portion was used as the validation set to test the model learned from the training set, which combined the remaining four portions. Training and testing AUCs were reported. The average training and testing AUC across all five times were also derived and reported. The best-performing set of hyperparameter values was chosen based on the highest mean test AUC. The best model would be the one refitted from the entire dataset using the best-performing set of hyperparameters values. We used this procedure for all methods involved in this study.

Comparing to 9 Other Machine Learning Methods

We compared the performance of the best-performing DNN model to that of a representative set of machine-learning methods, each obtained via grid search. The representative set of methods include Naïve Bayes (NB), Logistic Regression (LR), Decision Tree (DT), Support Vector Machine (SVM), the least absolute shrinkage and selection operator (LASSO), k-Nearest Neighbors (KNN), eXtreme Gradient Boosting (XGB), Adaptive Boosting (Adaboost), and Random Forest (RF). We used the scikit-learn [52, 53] package in Python to implement these machine learning classifiers. Like neural networks, these methods have hyperparameters that can be tuned to improve prediction performance. We conducted grid search for each method using each of the three LSM datasets. Like we did in our DNN grid-searches, we conducted 5-fold cross validation for each set of hyperparameter values and measured the performance by the AUC. Below, we provide a summary of the hyperparameters and their values that we tested for each of these methods

NB [60–63] represents a special type of Bayesian network model. Bayesian networks (BNs) are used for uncertain reasoning and machine learning in many domains, including biomedical informatics. *A* BN *consists of a* directed acyclic graph *(*DAG*)* $G = (V, E)$, whose nodeset V contains random variables and whose edges E represent relationships among the random variables. A BN also includes a conditional probability distribution of each node $X \in V$ given each combination of values of its parent nodes. Each node V in a BN is conditionally independent of all its nondescendents given its parents in the BN. NB is a simplified BN which normally only contains one parent node and a set of children nodes. In a basic NB model, there is an edge from the parent to each of the children. When a NB model is used to conduct classification, it is called a NB classifier. We used BernoulliBN classifier in this study because we have binary classes. Alpha is the Laplace smoothing parameter that deals with the problems of zero probability and regularize complexity, the larger the alpha, the stronger the smoothing and the lower the complexity of the model. We tested 500 alpha values, which are all positive integers from 1 to 500.

LR [64, 65] is a supervised learning classification method, which is normally suitable for binary classification problems. It is named after the logistic function, a core function of LR for nonlinear transformation on the output value. C is the inverse of regularization strength ($C=1/\lambda$). Smaller values result in stronger regularization. We tested 300 evenly spaced values on a logarithmic scale between 10^{-4} and 10^{4}. Regularization can be used to train models that generalize better on unseen data, by preventing the algorithm from overfitting the training dataset. We used either L1 or L2 methods to regularize the LR model.

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 29 June 2022 doi:10.20944/preprints202206.0394.v1

Decision Trees [65–67] is one of the most widely used machine learning methods. It contains a tree-like structure in which each internal node represents a test on a feature and each leaf node represents a class value. It can be used for both classification and regression tasks. This parameter max_depth indicates how deep the tree can be. The deeper the tree, the more splits it has, which allows it to capture more information about the data. We fit a decision tree with depths ranging from 3 to 32. The parameter min_samples_split governs the minimum number of samples required to split an internal node. The values we tested in our grid search are 0.1, 0.2, 0.3, 0.4, 0.6, 0.8, and 1. The parameter max_features indicates the max features when building a decision tree; we tested all values: none, 'log2' and 'sqrt'. The parameter max_leaf_nodes controls the maximum number of leaf nodes of each decision tree, we tested 7, 10, 15, and none. max_depth and max_leaf_nodes are important hyperparameters to control overfitting. Criterion is a function for measuring the quality of a split, and we tested both values 'gini' and 'entropy'.

SVM [68–74] is a machine learning method that identifies a hyperplane with margins defined by support vectors. Support vectors are a set of data points that are closer to the hyperplane and can influence both the position and direction of the hyperplane, which can be used to classify (separate) input samples. SVM can be used for both regression and classification tasks, and it is widely applied in the later. The parameter C trades off misclassification of training examples against simplicity of the decision surface. Smaller values result in a smoother decision surface, while larger values give the model more freedom to select more samples as support vectors. We tested values in the range 2^{-5}, 2^{-3}… 2^{15}. The parameter γ defines how far the influence of a single training example reaches (inverse of the radius of influence of samples selected by the model as support vectors). Low values mean "far" and high values mean "close". We tested values in the range 2^{-15}, 2^{-13}… 2^{3}.

LASSO [75] is a regression-based method classifier that is capable of conducting variable selection and regularization in order to enhance prediction performance and control overfitting. The parameter alpha is the sum of absolute value of coefficients which provides a trade-off between balancing residual sum of squares and magnitude of coefficients. Alpha can take various values that are greater than 0. We tested 400 evenly spaced alpha values on a logarithmic scale between 10^{-5} and 10^{5}.

KNN [76–78] is a supervised machine learning method that can be used for both classification and regression tasks. KNN predicts the class value of an incoming sample by its k nearest neighboring data points. KNN assumes that cases with similar covariate values are near to each other. The parameter k_neighbors is the number of training samples closest in distance to a query point in order to predict the label of the query. We tested all integers between 1 and 300. The parameter weights is the weighting criteria used to assign a value to a query point. We tested both the two available values uniform and distance. The value uniform assigns uniform weights to each neighbor. The value distance assigns weights to neighbors proportional to the inverse of the distance from the query point, so closer neighbors would weigh more. Metric is a parameter for choosing the method for calculating distance. We tested all available values, which are eluclidean, manhattan, and chebyshev.

RF [67, 79–82] is a typical model of bagging in ensemble learning, the trainer will randomly select a certain amount of sample data and create a corresponding decision tree. Many of these decision trees form a random forest. An advantage of RF is that the independent character of each decision tree tends to reduce overfitting. The parameter n_estimators is the number of

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 29 June 2022 doi:10.20944/preprints202206.0394.v1

decision trees in the random forest. We tested values 10, 50, 60, 70, ..., 200, and 500; Other parameters come from DT, and we tested the same values as we did with DT for them.

ADB [81–84] is a typical model of boosting in ensemble learning. Unlike the RF model, where each decision tree is independent, Adaboost is a classifier with cascade structure which means the next learner is based on the result of the previous weak learner. During the learning process, if the current sample is classified incorrectly, the degree of difficulty of the sample will increase to make the next learner focus on the difficult part on which previous model performed poorly. The parameter n_estimators is the number of weak learners. A model tends to overfit for large values of n_estimators. The values of n_estimators we tested include 10, 20, …, and 100. Learning_rate is used to shrink the contribution of each classifier. We tested all values from 0.002 to 0.01 with an increment of 0.001.

XGB [85–91] is another common approach for boosting in ensemble learning. Unlike ADB, it uses gradient boosting. The XGB classifier is based on the difference between true and predicted values to improve model performance. The parameter gamma is a pseudo-regularization hyperparameter in gradient boosting, and it affects pruning to control the overfitting problem. Gamma values we tested are 0, 0.01, 0.1, 0.3, 0.5, and 0.9. The parameter min_child_weight is minimum sum of weights of all observations of a child node. The larger the value, the more conservative the algorithm will be. The values tested were 1, 2, 4, and 6. Alpha and lambda are both regularization hyperparameters which can help control overfitting. The values we tested for each of them are 1e-5, 1e-2, 0.1, 1, and 100. The parameter max_depth is the maximum depth of the individual regression estimators. The values of max_depth we tested were 3, 4, 5, …, 30, 31. The learning_rate values we tested were 0, 0.01, 0.1, 0.3, and 0.5.

Statistical Testing

We conducted the Wilcoxon rank sum tests to determine the statistical significance of the AUC results. We conjectured that deep learning with grid search would perform no worse than other methods when predicting the binary status of 5, 10, and 15-year BCM. We paired the DNN with each of the 9 other machine learning methods, and conducted both the right-tailed (greater) and left-tailed (less) Wilcoxon tests for each pair of the methods and repeated these tests for each of the three datasets separately. The null hypothesis for all the Wilcoxon tests is that the two methods perform indifferently. The alternative hypothesis of the right-tailed Wilcoxon tests is DNN does better (greater) than the comparison method, and this is to test whether DNN performs better than other method. The alternative hypothesis of the left-tailed Wilcoxon tests is DNN does worse (less) than other method, and this is to test whether DNN performs worse than the comparison method. We conducted the Wilcoxon rank sum test in R using the wilcox.test() function included in the R package.

RESULTS

Table 4 shows the mean AUCs from 5-fold cross validation of the best-performing model for each method and each dataset, selected based the grid search results. Table 5 contains the results of the right-tailed Wilcoxon rank sum tests in which the alternative hypothesis is that the first method performs better (greater) than the second method in a pair of methods, while Table 6 shows the results of the tests in which the alternative hypothesis is the first method performs worse (less)

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 29 June 2022 doi:10.20944/preprints202206.0394.v1

than the second method. As shown in the first row of Table 5, X represents the first method and Y represents the second method. For example, in the cell of row 1 and column 2, DNN is the first method and BN is the second method. We included in Tables 5 and 6 W, the p-value, and the 95% confidence interval (CI) for each of the Wilcoxon tests we conducted. W is the test statistic used in the Wilcoxon rank sum test.

Table 7 contains the hyperparameter values of the best-performing DNN models learned from grid search using each of the three datasets. For example, the best model trained using the LSM-15year dataset contains 3 hidden layers, and each of them contains 300 hidden nodes; When we selected the best models, we not only considered the mean test AUC values, but also considered the percent_auc_diff as defined previously. To identify the best-performing DNN model, we first ordered the result table according to the mean test AUC values going from the highest to lowest. Then we looked at the percent_auc_diff values from the top of the ordered results and selected as the best model the first model whose percent_auc_diff value was less than 5%. Table 8 shows the average experiment time per model (in seconds), the number of all models trained via grid search, and total experiment time (in days) for each method and dataset.

We compared side by side the ROC curves of the best-performing models of DNN and the 9 comparison methods. Figures 2, 3, and 4 show these comparisons in the prediction of 5, 10, and 15-year BCM, each respectively. Figure 5 contains 4 panels of boxplots for comparing mean test AUC values of all methods side by side, one for each dataset separately and one for all datasets combined. We notice that for each of the methods, including deep learning, the prediction performance improves in general as the number of years it takes to metastasize increases. We also notice that LR, LASSO, SVM, and DNN perform extremely well when predicting the 15-year BCM. We demonstrate this using a bar graph as shown in Figure 6.

Table 4: The mean test AUCs and mean train AUCs of the best-performing models

Mean Test AUC/Mean Train AUC	LSM-5year	LSM-10year	LSM-15year
DNN	0.769/0.806	0.793/0.830	0.842/0.873
Naïve bayes	0.751/0.753	0.797/0.798	0.763/0.826
Logistic Regression	0.771 /0.773	0.777/0.809	0.844/0.884
Decision Tree	0.762/0.780	0.783/0.827	0.783/0.838
SVM	0.739/0.811	0.771/0.808	0.845/0.867
LASSO	0.772/0.774	0.778/0.806	0.844/0.887
K nearest neighbor	0.789/0.816	0.793/0.819	0.799/0.832
Random Forest	0.789/0.801	0.804/0.840	0.802/0.849
Adaboost	0.759/0.754	0.792/0.800	0.796/0.829
Xgboost	0.793/0.813	0.806/0.845	0.800/0.854

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 29 June 2022 doi:10.20944/preprints202206.0394.v1

Table 5: Significance test results: one-tailed (greater) Wilcoxon rank sum tests

X, Y		DNN, NB	DNN, LR	DNN, DT	DNN, LASSO	DNN, SVM	DNN, KNN	DNN, RF	ADB, DNN	DNN, XGB
5-year Metastasis	W	19	11	16	11	18	6	7	11	6
	p-value	0.111	0.655	0.274	0.655	0.155	0.925	0.889	0.655	0.925
	95 % CI	[-0.0174 Inf]	[-0.0269 Inf]	[-0.0112 Inf]	[-0.0254 Inf]	[-0.0057 Inf]	[-0.0453 Inf]	[-0.0463 Inf]	[-0.0388 Inf]	[-0.0520 Inf]
10-year Metastasis	W	10	19	19	22	22	12	8	14	5
	p-value	0.726	0.111	0.111	0.028	0.028	0.579	0.843	0.421	0.952
	95 % CI	[-0.0197 Inf]	[-0.0005 Inf]	[-0.0030 Inf]	[0.0012 Inf]	[0.0047 Inf]	[-0.0198 Inf]	[-0.0312 Inf]	[-0.0168 Inf]	[-0.0300 Inf]
15-year Metastasis	W	24	9	22	9	9	20	19	6	20
	p-value	0.011	0.799	0.030	0.799	0.799	0.071	0.104	0.929	0.071
	95 % CI	[0.0291 Inf]	[-0.0555 Inf]	[0.0013 Inf]	[-0.0505 Inf]	[-0.0529 Inf]	[-0.0012 Inf]	[-0.0026 Inf]	[-0.1120 Inf]	[-0.0017 Inf]

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 29 June 2022 doi:10.20944/preprints202206.0394.v1

Table 6: Significance test results: one-tailed (less) Wilcoxon rank sum tests

X, Y		DNN, NB	DNN, LR	DNN, DT	DNN, LASSO	DNN, SVM	DNN, KNN	DNN, RF	ADB, DNN	DNN, XGB
5-year BCM	W	19	11	16	11	18	6	7	11	6
	p-value	0.925	0.421	0.790	0.421	0.889	0.111	0.155	0.421	0.111
	95 % CI	[-Inf 0.0583]	[-Inf 0.0252]	[-Inf 0.0380]	[-Inf 0.0241]	[-Inf 0.0735]	[-Inf 0.0145]	[-Inf 0.0032]	[-Inf 0.0223]	[-Inf 0.0041]
10-year BCM	W	10	19	19	22	22	12	8	14	5
	p-value	0.345	0.925	0.925	0.984	0.984	0.500	0.210	0.655	0.075
	95 % CI	[-Inf 0.0096]	[-Inf 0.0317]	[-Inf 0.0274]	[-Inf 0.0303]	[-Inf 0.0375]	[-Inf 0.0182]	[-Inf 0.0068]	[-Inf 0.0192]	[-Inf 0.0041]
15-year BCM	W	24	9	22	9	9	20	19	6	20
	p-value	0.994	0.265	0.982	0.265	0.265	0.953	0.929	0.104	0.953
	95 % CI	[-Inf 0.1494]	[-Inf 0.0826]	[-Inf 0.1431]	[-Inf 0.0745]	[-Inf 0.0719]	[-Inf 0.0927]	[-Inf 0.0920]	[-Inf 0.0149]	[-Inf 0.0979]

Table 7: The hyperparameter values of the best-performing DNN models learned from 5-year, 10-year, and 15-year datasets, respectively.

Hyperparameter Values of the Best-performing Model	LSM-5 Year	LSM-10 Year	LSM-15 Year
Number of hidden layers.	2	1	3

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 29 June 2022 doi:10.20944/preprints202206.0394.v1

Number of hidden nodes	{75, 75}	{75}	{300, 300, 300}
Kernel initializer	he_normal	he_normal	he_normal
Optimizer	SGD	SGD	SGD
Learning rate	0.005	0.01	0.005
Momentum Beta	0.9	0.9	0.9
Iteration-based decay	0.01	0.01	0.01
Dropout rate	0.5	0.5	0.5
Epochs	100	100	100
L1	0	0	0
L2	0.008	0.008	0.008
L1 and L2 combined	No	No	No

Table 8: Experiment time per model per dataset, number of models trained, and total experiment time

Method	LSM-5(sec)	LSM-10(sec)	LSM-15(sec)	# of Models Trained	Total Time (days)
DNN	117.430	45.021	20.212	24111	50.974
NB	0.060	0.046	0.026	18109	0.028
LR	0.563	0.353	0.253	22399	0.303
DT	0.048	0.037	0.032	107351	0.145
LASSO	0.860	0.372	0.189	1024	0.017
SVM	12.197	2.876	0.362	1799	0.321
KNN	1.636	0.436	0.132	42341	1.080
RF	0.774	0.603	0.549	27000	0.602
ADB	0.655	0.508	0.403	13	0.000
XGB	4.710	4.566	3.850	46980	7.137

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 29 June 2022 doi:10.20944/preprints202206.0394.v1

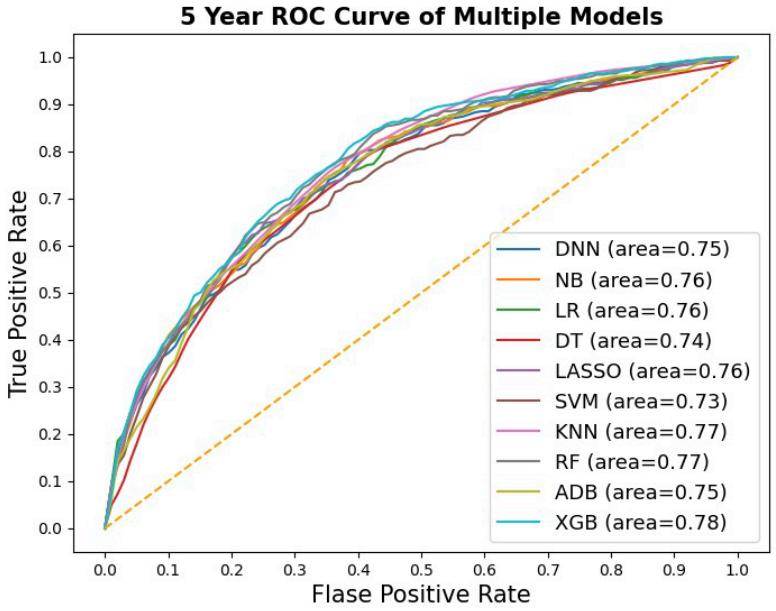

Figure 2: ROC curves of the best-performing models for all methods each respectively for predicting 5-year metastasis.

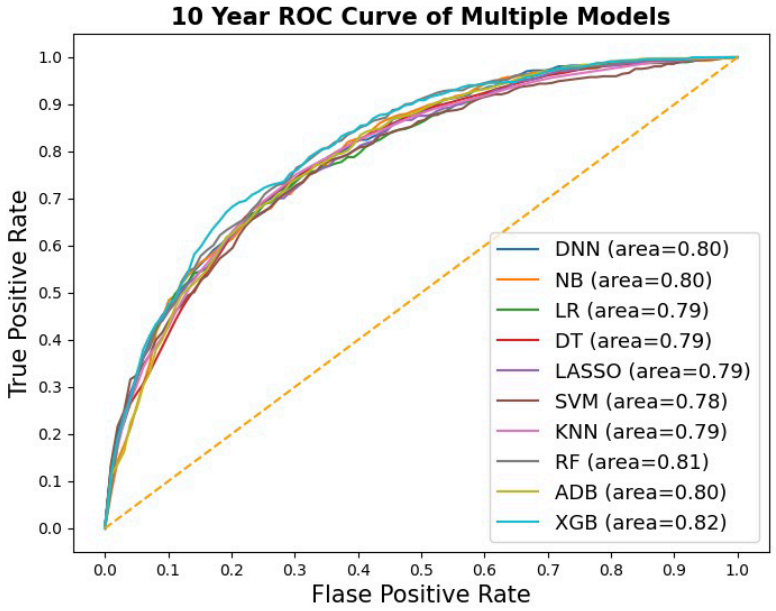

Figure 3: ROC curves of the best-performing models for all methods each respectively for predicting 10-year metastasis.

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 29 June 2022 doi:10.20944/preprints202206.0394.v1

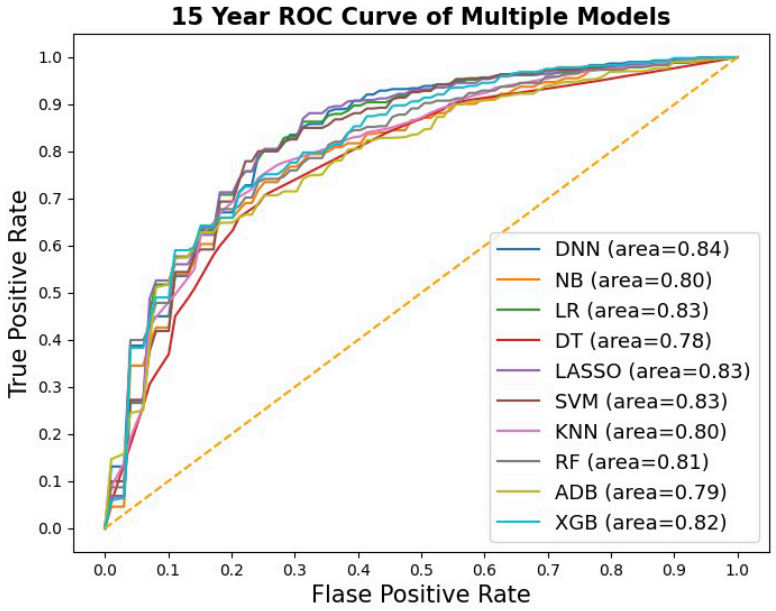

Figure 4. ROC curves of the best-performing models for all methods each respectively for predicting 15-year metastasis.

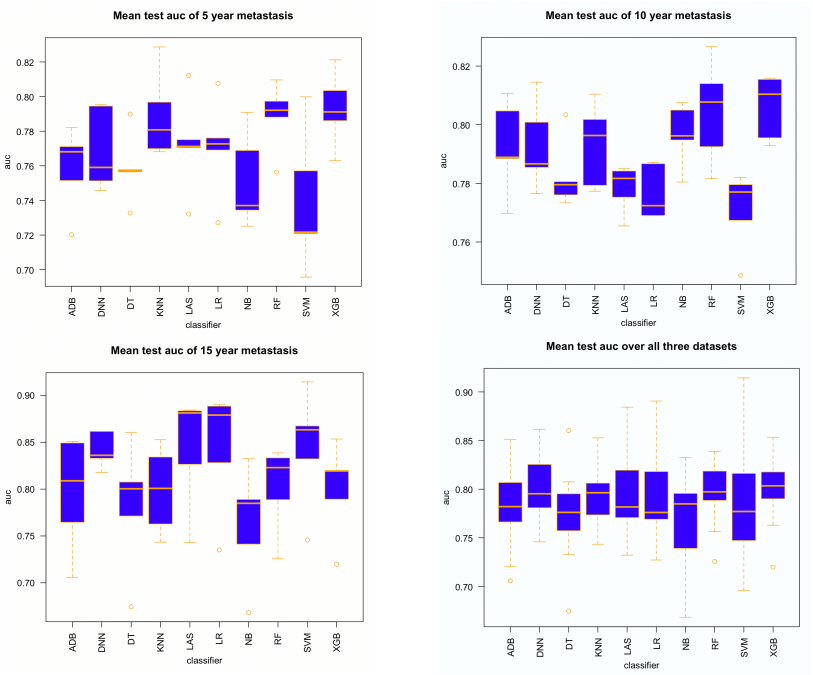

Figure 5: Boxplots to compare the mean test AUCs of all methods

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 29 June 2022 doi:10.20944/preprints202206.0394.v1

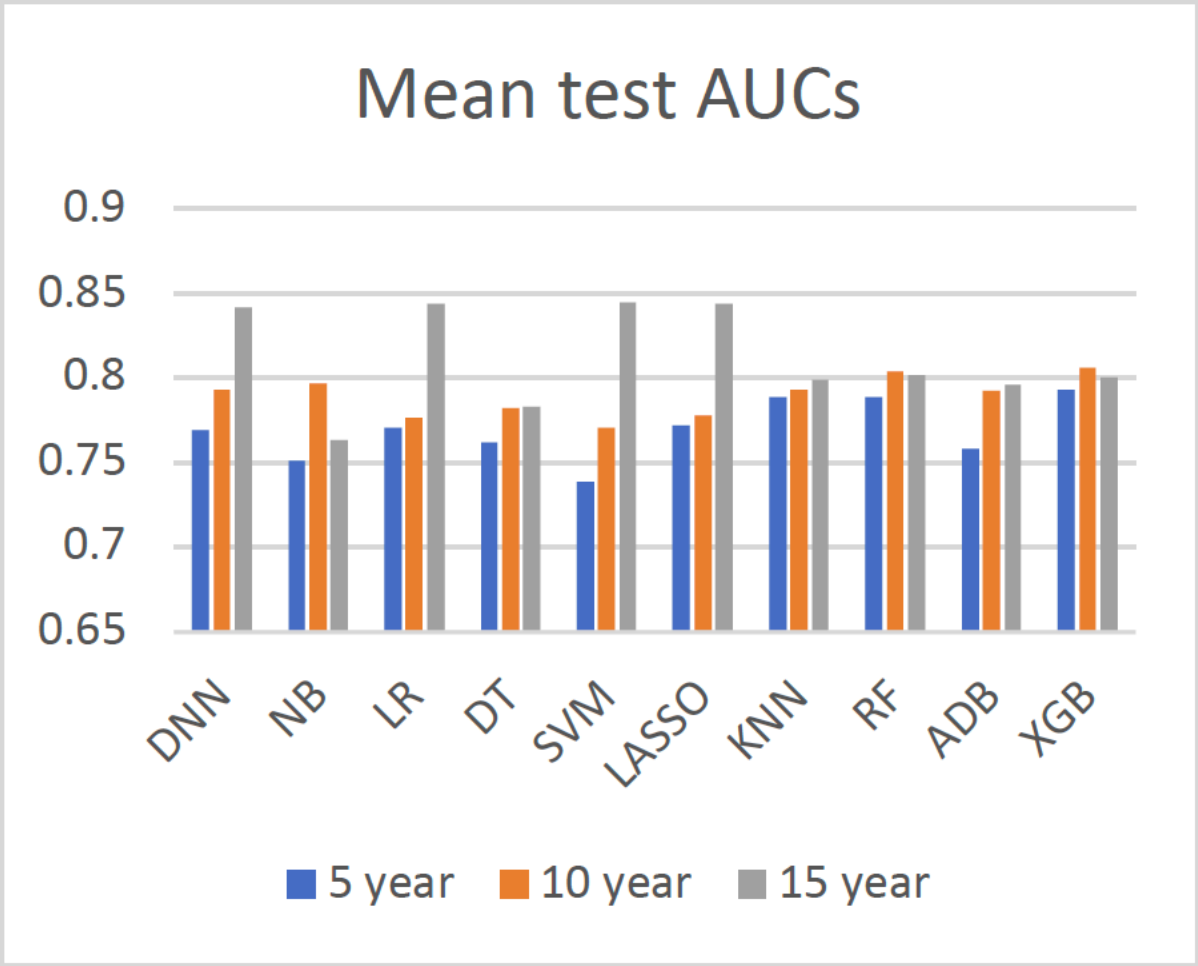

Figure 6: Side by side comparisons of the mean test AUCs of all methods when predicting 5, 10, and 15-year breast cancer metastasis

DISCUSSION

Based on the mean test AUC values shown in Table 4, XGB (1st), RF (2nd), and KNN (3rd) are the top three methods in predicting 5-year BCM. DNN ranks 6th and performs better than NB, DT, SVM, and ADB in this category. When predicting 10-year BCM, XGB (1st), RF (2nd), and NB (3rd) are the top three performers. DNN and KNN tie as the number 4 performers, so DNN performs better than LR, DT, SVM, LASSO, and ADB in this category. When predicting 15-year BCM, SVM (1st), LR and LASSO (tie for 2nd) and DNN (3rd) are the top three performers, so in this category, DNN outperforms the other 6 methods including NB, DT, KNN, RF, ADB, and XGB.

We notice that in each of the three metastasis categories, the mean test AUC values of the top performers are quite close to each other. For instance, when predicting 15-year BCM, the mean test AUC values of the top four performers are 0.842 (DNN), 0.844 (LR), 0.844 (LASSO), and 0.845 (SVM). We further look at the statistical testing results shown in Tables 5 and 6 to compare DNN with each of the 9 other machine learning methods. As shown in Table 5, the p-values we obtained for each pair methods range from 0.111 (DNN vs NB) to 0.925 (DNN vs KNN) in

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 29 June 2022 doi:10.20944/preprints202206.0394.v1

predicting 5-year BCM, which indicates that at a significance level of 0.05, we are not confident in rejecting the null hypothesis which states that DNN performs no difference from the comparison methods. Table 5 also shows that DNN performs better than both LASSO (p-value 0.028) and SVM (p-value 0.028) but no difference from other methods at a significance level of 0.05 when predicting 10-year BCM. Again according to Table 5, DNN performs better than NB, DT with a p-value of 0.011 and 0.030 each respectively, but no difference from other methods at a significance level of 0.05 when predicting 15-year BCM. Based on Table 6, DNN performs no worse than any of the comparison methods at a significance level of 0.05 for any of the three BCM categories. Overall, our statistical testing results support our conjecture that deep learning with grid search perform no worse than the comparison methods when predicting the binary status of BCM.

The Potential Effects of Imbalance Data

As demonstrated in Figure 5, the prediction performance of all methods improves in general as the number of years to metastasis increases. Concurrently, as shown in Table 1, the data become more balanced as the number of years to metastasis increases. This may indicate that data balance has in general a positive effect on the prediction performance of these machine learning methods. Additionally, we observe that the mean test AUCs of DNN, SVM, LASSO and LR, when predicting the 15-year BCM, are significantly higher than that of these methods when predicting the 5-year and 10-year BCM. An explanation for this is the 15-year dataset is much more balanced than the 5-year and 10-year dataset. This may indicate that these four methods are more sensitive to imbalanced data and potentially superior methods for predicting breast cancer metastasis when a dataset is well balanced. The two ensemble methods XGB and RF outperform all other methods when predicting the 10-year and 15-year BCM, for which data are less balanced. This may indicate that these ensemble methods tend to handle imbalanced data better.

Computation time

Table 8 shows that the average experiment time per model of DNN is way higher than that of any other method. This is perhaps because DNN has a large number of hyperparameters, and its internal parameters (weights and biases) rapidly increase as the number of hidden nodes and the number of hidden layers are increased.

CONCLUSIONS

Based on the statistical testing results, we conclude that at a significance level of 0.05, DNN performs no worse than any of the 9 comparison methods when predicting the 5, 10, and 15-year BCM. This is consistent with our conjecture that deep learning with grid search perform no worse than the comparison methods when predicting the binary status of BCM. On the other hand, it is interesting to learn that some of the other machine learning methods such as XGB, RF, and SVM are very strong competitors of DNN. Besides, obtaining the best-performing DNN models required much more computation time than doing so for the 9 comparison methods.

DECLARATIONS

Ethics approval and consent to participate

The study was approved by University of Pittsburgh Institutional Review Board (IRB # 196003)

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 29 June 2022 doi:10.20944/preprints202206.0394.v1

and the U.S. Army Human Research Protection Office (HRPO # E01058.1a).

The need for patient consent was waived by the ethics committees because the data consists only of de-identified data that are publicly available.

Consent for publication

Not applicable.

Availability of data and material

The data used in this study are available at datadryad.org (DOI 10. 5061/dryad.64964m0).

Competing interests

The authors declare that they have no competing interests.

Funding

Research reported in this paper was supported by the U.S. Department of Defense through the Breast Cancer Research Program under Award No. W81XWH-19-1-0495 (to XJ). Other than supplying funds, the funding agencies played no role in the research.

Authors' Contribution

XJ originated the study and wrote the first draft of the manuscript. XJ implemented the DNN method, and CX implemented the 9 comparison methods. Both XJ and CX conducted the experiments, and prepared and analyzed the results. XJ performed all statistical analyses of the results. All authors contributed to the preparation and revision of the manuscript.

Acknowledgement

Thank Greg Cooper and Peter Gao for proofreading the manuscript and providing their valuable comments.

REFERENCES

[1] H. Sung *et al.*, "Global Cancer Statistics 2020: GLOBOCAN Estimates of Incidence and Mortality Worldwide for 36 Cancers in 185 Countries," *CA: A Cancer Journal for Clinicians*, vol. 71, no. 3, pp. 209–249, May 2021, doi: 10.3322/caac.21660.

[2] L. Rahib, M. R. Wehner, L. M. Matrisian, and K. T. Nead, "Estimated Projection of US Cancer Incidence and Death to 2040," *JAMA Network Open*, vol. 4, no. 4, Apr. 2021, doi: 10.1001/jamanetworkopen.2021.4708.

[3] American Cancer Society, "Cancer Facts and Figures 2021." https://www.cancer.org/research/cancer-facts-statistics/all-cancer-facts-figures/cancer-facts-figures-2021.html (accessed Jun. 04, 2021).

[4] C. E. DeSantis *et al.*, "Breast cancer statistics, 2019," *CA: A Cancer Journal for Clinicians*, vol. 69, no. 6, 2019, doi: 10.3322/caac.21583.

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 29 June 2022 doi:10.20944/preprints202206.0394.v1

[5] A. M. Afifi, A. M. Saad, M. J. Al-Husseini, A. O. Elmehrath, D. W. Northfelt, and M. B. Sonbol, "Causes of death after breast cancer diagnosis: A US population-based analysis," *Cancer*, vol. 126, no. 7, pp. 1559–1567, Apr. 2020, doi: 10.1002/cncr.32648.

[6] R. L. Siegel, K. D. Miller, and A. Jemal, "Cancer statistics, 2020," *CA: A Cancer Journal for Clinicians*, vol. 70, no. 1, pp. 7–30, Jan. 2020, doi: 10.3322/caac.21590.

[7] G. P. Gupta and J. Massagué, "Cancer Metastasis: Building a Framework," *Cell*, vol. 127, no. 4. Elsevier B.V., pp. 679–695, Nov. 17, 2006. doi: 10.1016/j.cell.2006.11.001.

[8] B. Weigelt *et al.*, "Refinement of breast cancer classification by molecular characterization of histological special types," *Journal of Pathology*, vol. 216, no. 2, 2008, doi: 10.1002/path.2407.

[9] L. A. Carey *et al.*, "The triple negative paradox: Primary tumor chemosensitivity of breast cancer subtypes," *Clinical Cancer Research*, vol. 13, no. 8, 2007, doi: 10.1158/1078-0432.CCR-06-1109.

[10] D. C. Koboldt *et al.*, "Comprehensive molecular portraits of human breast tumours," *Nature*, vol. 490, no. 7418, 2012, doi: 10.1038/nature11412.

[11] R. M. Neve *et al.*, "A collection of breast cancer cell lines for the study of functionally distinct cancer subtypes," *Cancer Cell*, vol. 10, no. 6, 2006, doi: 10.1016/j.ccr.2006.10.008.

[12] B. Fisher *et al.*, "Twenty-Year Follow-up of a Randomized Trial Comparing Total Mastectomy, Lumpectomy, and Lumpectomy plus Irradiation for the Treatment of Invasive Breast Cancer," *New England Journal of Medicine*, vol. 347, no. 16, 2002, doi: 10.1056/nejmoa022152.

[13] Z. Zeng *et al.*, "Using natural language processing and machine learning to identify breast cancer local recurrence," *BMC Bioinformatics*, vol. 19, 2018, doi: 10.1186/s12859-018-2466-x.

[14] X. Zhou, K. Y. Liu, and S. T. C. Wong, "Cancer classification and prediction using logistic regression with Bayesian gene selection," *Journal of Biomedical Informatics*, vol. 37, no. 4, 2004, doi: 10.1016/j.jbi.2004.07.009.

[15] B. Cai and X. Jiang, "Computational methods for ubiquitination site prediction using physicochemical properties of protein sequences," *BMC Bioinformatics*, vol. 17, no. 1, 2016, doi: 10.1186/s12859-016-0959-z.

[16] S. Lee and X. Jiang, "Modeling miRNA-mRNA interactions that cause phenotypic abnormality in breast cancer patients," *PLoS ONE*, vol. 12, no. 8, 2017, doi: 10.1371/journal.pone.0182666.

[17] Q. Long, M. Chung, C. S. Moreno, and B. A. Johnson, "Risk prediction for prostate cancer recurrence through regularized estimation with simultaneous adjustment for nonlinear clinical effects," *Annals of Applied Statistics*, vol. 5, no. 3, 2011, doi: 10.1214/11-AOAS458.

[18] T. R. Golub *et al.*, "Molecular classification of cancer: Class discovery and class prediction by gene expression monitoring," *Science*, vol. 286, no. 5439, 1999, doi: 10.1126/science.286.5439.531.

[19] Y. Wang, F. S. Makedon, J. C. Ford, and J. Pearlman, "HykGene: A hybrid approach for selecting marker genes for phenotype classification using microarray gene expression data," *Bioinformatics*, vol. 21, no. 8, 2005, doi: 10.1093/bioinformatics/bti192.

[20] W. S. McCulloch and W. Pitts, "A logical calculus of the ideas immanent in nervous activity," *The Bulletin of Mathematical Biophysics*, vol. 5, no. 4, 1943, doi: 10.1007/BF02478259.

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 29 June 2022 doi:10.20944/preprints202206.0394.v1

[21] B. G. Farley and W. A. Clark, "Simulation of self-organizing systems by digital computer," *IRE Professional Group on Information Theory*, vol. 4, no. 4, 1954, doi: 10.1109/TIT.1954.1057468.

[22] J. Schmidhuber, "Deep learning," *Encyclopedia of Machine Learning and Data Mining*, pp. 1–11, 2016.

[23] R. E. Neapolitan and X. Jiang, "Neural Networks and Deep Learning," in *Artificial Intelligence*, 2018. doi: 10.1201/b22400-15.

[24] J. Schmidhuber, "Deep Learning in neural networks: An overview," *Neural Networks*, vol. 61. 2015. doi: 10.1016/j.neunet.2014.09.003.

[25] D. E. Rumelhart, G. E. Hinton, and J. L. Mcclelland, "A General framework for Parallel Distributed Processing," *Parallel distributed processing: explorations in the microstructure of cognition*. 1986.

[26] L. J. Lancashire *et al.*, "A validated gene expression profile for detecting clinical outcome in breast cancer using artificial neural networks," *Breast Cancer Research and Treatment*, vol. 120, no. 1, 2010, doi: 10.1007/s10549-009-0378-1.

[27] S. Belciug and F. Gorunescu, "A hybrid neural network/genetic algorithm applied to breast cancer detection and recurrence," *Expert Systems*, vol. 30, no. 3, 2013, doi: 10.1111/j.1468-0394.2012.00635.x.

[28] M. A. Fiddy, "Regularized Image Reconstruction Using SVD and a Neural Network Method for Matrix Inversion," *IEEE Transactions on Signal Processing*, vol. 41, no. 10, 1993, doi: 10.1109/78.277813.

[29] J. Hua, J. Lowey, Z. Xiong, and E. R. Dougherty, "Noise-injected neural networks show promise for use on small-sample expression data," *BMC Bioinformatics*, vol. 7, 2006, doi: 10.1186/1471-2105-7-274.

[30] I. Saritas, "Prediction of breast cancer using artificial neural networks," *Journal of Medical Systems*, vol. 36, no. 5, 2012, doi: 10.1007/s10916-011-9768-0.

[31] L. Ran, Y. Zhang, Q. Zhang, and T. Yang, "Convolutional neural network-based robot navigation using uncalibrated spherical images," *Sensors (Switzerland)*, vol. 17, no. 6, 2017, doi: 10.3390/s17061341.

[32] L. Deng, G. Tur, X. He, and D. Hakkani-Tur, "Use of kernel deep convex networks and end-to-end learning for spoken language understanding," 2012. doi: 10.1109/SLT.2012.6424224.

[33] S. Fernández, A. Graves, and J. Schmidhuber, "An application of recurrent neural networks to discriminative keyword spotting," in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 2007, vol. 4669 LNCS, no. PART 2. doi: 10.1007/978-3-540-74695-9_23.

[34] N. Naik *et al.*, "Deep learning-enabled breast cancer hormonal receptor status determination from base-level H&E stains," *Nature Communications*, vol. 11, no. 1, 2020, doi: 10.1038/s41467-020-19334-3.

[35] S. Min, B. Lee, and S. Yoon, "Deep learning in bioinformatics," *Briefings in bioinformatics*, vol. 18, no. 5. 2017. doi: 10.1093/bib/bbw068.

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 29 June 2022 doi:10.20944/preprints202206.0394.v1

[36] A. S. Lundervold and A. Lundervold, "An overview of deep learning in medical imaging focusing on MRI," *Zeitschrift fur Medizinische Physik*, vol. 29, no. 2. 2019. doi: 10.1016/j.zemedi.2018.11.002.

[37] X. Glorot and Y. Bengio, "Understanding the difficulty of training deep feedforward neural networks," in *Journal of Machine Learning Research*, 2010, vol. 9.

[38] NIH, "The Promise of Precision Medicine." https://www.nih.gov/about-nih/what-we-do/nih-turning-discovery-into-health/promise-precision-medicine (accessed Jun. 09, 2021).

[39] X. Jiang, A. Wells, A. Brufsky, and R. Neapolitan, "A clinical decision support system learned from data to personalize treatment recommendations towards preventing breast cancer metastasis," *PLoS ONE*, vol. 14, no. 3, 2019, doi: 10.1371/journal.pone.0213292.

[40] X. Jiang, A. Wells, A. Brufsky, D. Shetty, K. Shajihan, and R. E. Neapolitan, "Leveraging Bayesian networks and information theory to learn risk factors for breast cancer metastasis," *BMC Bioinformatics*, vol. 21, no. 1, 2020, doi: 10.1186/s12859-020-03638-8.

[41] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: A simple way to prevent neural networks from overfitting," *Journal of Machine Learning Research*, vol. 15, 2014.

[42] H. Chereda *et al.*, "Explaining decisions of graph convolutional neural networks: patient-specific molecular subnetworks responsible for metastasis prediction in breast cancer," *Genome Medicine*, vol. 13, no. 1, 2021, doi: 10.1186/s13073-021-00845-7.

[43] Y. W. Lee, C. S. Huang, C. C. Shih, and R. F. Chang, "Axillary lymph node metastasis status prediction of early-stage breast cancer using convolutional neural networks," *Computers in Biology and Medicine*, vol. 130, 2021, doi: 10.1016/j.compbiomed.2020.104206.

[44] N. Papandrianos, E. Papageorgiou, A. Anagnostis, and A. Feleki, "A deep-learning approach for diagnosis of metastatic breast cancer in bones from whole-body scans," *Applied Sciences (Switzerland)*, vol. 10, no. 3, 2020, doi: 10.3390/app10030997.

[45] L. Q. Zhou *et al.*, "Lymph node metastasis prediction from primary breast cancer US images using deep learning," *Radiology*, vol. 294, no. 1, 2020, doi: 10.1148/radiol.2019190372.

[46] X. Yang *et al.*, "Deep Learning Signature Based on Staging CT for Preoperative Prediction of Sentinel Lymph Node Metastasis in Breast Cancer," *Academic Radiology*, vol. 27, no. 9, 2020, doi: 10.1016/j.acra.2019.11.007.

[47] G. Litjens *et al.*, "A survey on deep learning in medical image analysis," *Medical Image Analysis*, vol. 42. 2017. doi: 10.1016/j.media.2017.07.005.

[48] M. D. Zakir Hossain, F. Sohel, M. F. Shiratuddin, and H. Laga, "A comprehensive survey of deep learning for image captioning," *ACM Computing Surveys*, vol. 51, no. 6. 2019. doi: 10.1145/3295748.

[49] S. P. Mohanty, D. P. Hughes, and M. Salathé, "Using deep learning for image-based plant disease detection," *Frontiers in Plant Science*, vol. 7, no. September, 2016, doi: 10.3389/fpls.2016.01419.

[50] B. Cai and X. Jiang, "A novel artificial neural network method for biomedical prediction based on matrix pseudo-inversion," *Journal of biomedical informatics*, vol. 48, pp. 114–121, 2014.

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 29 June 2022 doi:10.20944/preprints202206.0394.v1

[51] T. Szandała, "Review and comparison of commonly used activation functions for deep neural networks," in *Studies in Computational Intelligence*, vol. 903, 2021. doi: 10.1007/978-981-15-5495-7_11.

[52] M. J. J. Douglass, "Book Review: Hands-on Machine Learning with Scikit-Learn, Keras, and Tensorflow, 2nd edition by Aurélien Géron," *Physical and Engineering Sciences in Medicine*, vol. 43, no. 3, 2020, doi: 10.1007/s13246-020-00913-z.

[53] I. Stancin and A. Jovic, "An overview and comparison of free Python libraries for data mining and big data analysis," 2019. doi: 10.23919/MIPRO.2019.8757088.

[54] K. He, X. Zhang, S. Ren, and J. Sun, "Delving deep into rectifiers," *Proceedings of the IEEE International Conference on Computer Vision*, vol. 2015 Inter, 2015.

[55] L. S. Kim, "Understanding the difficulty of training deep feedforward neural networks Xavier," *Proceedings of the International Joint Conference on Neural Networks*, vol. 2, 1993.

[56] H. Shen, "Towards a Mathematical Understanding of the Difficulty in Learning with Feedforward Neural Networks," 2018. doi: 10.1109/CVPR.2018.00091.

[57] B. Ghojogh and M. Crowley, "The Theory Behind Overfitting, Cross Validation, Regularization, Bagging, and Boosting: Tutorial," May 2019, Accessed: Aug. 08, 2021. [Online]. Available: https://arxiv.org/abs/1905.12787v1

[58] Z. Li, K. Kamnitsas, and B. Glocker, "Overfitting of Neural Nets Under Class Imbalance: Analysis and Improvements for Segmentation," *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 11766 LNCS, pp. 402–410, Oct. 2019, doi: 10.1007/978-3-030-32248-9_45.

[59] X. Ying, "An Overview of Overfitting and its Solutions," *Journal of Physics: Conference Series*, vol. 1168, no. 2, p. 022022, Feb. 2019, doi: 10.1088/1742-6596/1168/2/022022.

[60] N. Friedman, D. Geiger, and M. Goldszmidt, "Bayesian Network Classifiers," *Machine Learning*, vol. 29, no. 2–3, 1997, doi: 10.1023/a:1007465528199.

[61] N. v. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer, "SMOTE: Synthetic minority over-sampling technique," *Journal of Artificial Intelligence Research*, vol. 16, 2002, doi: 10.1613/jair.953.

[62] R. Neapolitan, *Learning Bayesian Networks*. Upper Saddle River: Prentice Hall, 2004. Accessed: Oct. 07, 2021. [Online]. Available: https://www.amazon.com/Learning-Bayesian-Networks-Richard-Neapolitan/dp/0130125342/ref=sr_1_3?dchild=1&keywords=Learning+Bayesian+Networks&qid=1628620634&sr=8-3

[63] A. McCallum and K. Nigam, "A Comparison of Event Models for Naive Bayes Text Classification," *AAAI/ICML-98 Workshop on Learning for Text Categorization*, 1998, doi: 10.1.1.46.1529.

[64] A. Y. Ng and M. I. Jordan, "On discriminative vs. Generative classifiers: A comparison of logistic regression and naive bayes," 2002.

[65] J. Friedman, T. Hastie, and R. Tibshirani, "Additive logistic regression: A statistical view of boosting," *Annals of Statistics*, vol. 28, no. 2. 2000. doi: 10.1214/aos/1016218223.

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 29 June 2022 doi:10.20944/preprints202206.0394.v1

[66] S. R. Safavian and D. Landgrebe, "A Survey of Decision Tree Classifier Methodology," *IEEE Transactions on Systems, Man and Cybernetics*, vol. 21, no. 3, 1991, doi: 10.1109/21.97458.

[67] T. K. Ho, "Random decision forests," in *Proceedings of the International Conference on Document Analysis and Recognition, ICDAR*, 1995, vol. 1. doi: 10.1109/ICDAR.1995.598994.

[68] J. A. K. Suykens and J. Vandewalle, "Least squares support vector machine classifiers," *Neural Processing Letters*, vol. 9, no. 3, 1999, doi: 10.1023/A:1018628609742.

[69] E. Osuna, R. Freund, and F. Girosi, "Training support vector machines: An application to face detection," 1997. doi: 10.1109/cvpr.1997.609310.

[70] C. Cortes and V. Vapnik, "Support-vector networks," *Machine Learning*, vol. 20, no. 3, pp. 273–297, 1995, doi: 10.1007/bf00994018.

[71] Z. R. Yang, "Biological applications of support vector machines," *Briefings in Bioinformatics*, vol. 5, no. 4, pp. 328–338, 2004, doi: 10.1093/bib/5.4.328.

[72] C.-W. Hsu, C.-C. Chang, and C.-J. Lin, "A practical guide to support vector classification.," Department of Computer Science, National Taiwan University, 2003.

[73] H. Wang, B. Zheng, S. W. Yoon, and H. S. Ko, "A support vector machine-based ensemble algorithm for breast cancer diagnosis," *European Journal of Operational Research*, vol. 267, no. 2, pp. 687–699, 2018, doi: 10.1016/j.ejor.2017.12.001.

[74] K. S. Parikh and T. P. Shah, "Support Vector Machine – A Large Margin Classifier to Diagnose Skin Illnesses," *Procedia Technology*, vol. 23, pp. 369–375, 2016, doi: 10.1016/j.protcy.2016.03.039.

[75] R. Tibshirani, M. Saunders, S. Rosset, J. Zhu, and K. Knight, "Sparsity and smoothness via the fused lasso," *Journal of the Royal Statistical Society. Series B: Statistical Methodology*, vol. 67, no. 1, 2005, doi: 10.1111/j.1467-9868.2005.00490.x.

[76] K. Q. Weinberger, J. Blitzer, and L. K. Saul, "Distance metric learning for large margin nearest neighbor classification," 2005.

[77] Y. Yang and X. Liu, "A re-examination of text categorization methods," 1999. doi: 10.1145/312624.312647.

[78] K. Q. Weinberger and L. K. Saul, "Distance metric learning for large margin nearest neighbor classification," *Journal of Machine Learning Research*, vol. 10, 2009, doi: 10.1145/1577069.1577078.

[79] D. R. Cutler *et al.*, "Random forests for classification in ecology," *Ecology*, vol. 88, no. 11, 2007, doi: 10.1890/07-0539.1.

[80] D. Opitz and R. Maclin, "Popular Ensemble Methods: An Empirical Study," *Journal of Artificial Intelligence Research*, vol. 11, 1999, doi: 10.1613/jair.614.

[81] T. G. Dietterich, "Ensemble methods in machine learning," in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 2000, vol. 1857 LNCS. doi: 10.1007/3-540-45014-9_1.

[82] L. Breiman, "Random forests," *Machine learning*, vol. 45, no. 1, pp. 5–32, 2001.

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 29 June 2022 doi:10.20944/preprints202206.0394.v1

[83] P. Viola and M. J. Jones, "Robust Real-Time Face Detection," *International Journal of Computer Vision*, vol. 57, no. 2, 2004, doi: 10.1023/B:VISI.0000013087.49260.fb.

[84] P. Viola and M. Jones, "Rapid object detection using a boosted cascade of simple features," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2001, vol. 1. doi: 10.1109/cvpr.2001.990517.

[85] M. Zięba, S. K. Tomczak, and J. M. Tomczak, "Ensemble boosted trees with synthetic features generation in application to bankruptcy prediction," *Expert Systems with Applications*, vol. 58, 2016, doi: 10.1016/j.eswa.2016.04.001.

[86] L. Torlay, M. Perrone-Bertolotti, E. Thomas, and M. Baciu, "Machine learning–XGBoost analysis of language networks to classify patients with epilepsy," *Brain Informatics*, vol. 4, no. 3, 2017, doi: 10.1007/s40708-017-0065-7.

[87] Y. Xia, C. Liu, Y. Y. Li, and N. Liu, "A boosted decision tree approach using Bayesian hyper-parameter optimization for credit scoring," *Expert Systems with Applications*, vol. 78, 2017, doi: 10.1016/j.eswa.2017.02.017.

[88] S. R. Mousa, P. R. Bakhit, O. A. Osman, and S. Ishak, "A comparative analysis of tree-based ensemble methods for detecting imminent lane change maneuvers in connected vehicle environments," *Transportation Research Record*, vol. 2672, no. 42, 2018, doi: 10.1177/0361198118780204.

[89] H. Hu *et al.*, "HLPI-Ensemble: Prediction of human lncRNA-protein interactions based on ensemble strategy," *RNA Biology*, vol. 15, no. 6, 2018, doi: 10.1080/15476286.2018.1457935.

[90] M. H. D. M. Ribeiro and L. dos Santos Coelho, "Ensemble approach based on bagging, boosting and stacking for short-term prediction in agribusiness time series," *Applied Soft Computing Journal*, vol. 86, 2020, doi: 10.1016/j.asoc.2019.105837.

[91] A. Torres-Barrán, Á. Alonso, and J. R. Dorronsoro, "Regression tree ensembles for wind energy and solar radiation prediction," *Neurocomputing*, vol. 326–327, 2019, doi: 10.1016/j.neucom.2017.05.104.

FIGURE LEGEND

Figure 1: A feed-forward ANN that contains one hidden layer.

Figure 2: ROC curves of the best-performing models for all methods each respectively for predicting 5-year metastasis.

Figure 3: ROC curves of the best-performing models for all methods each respectively for predicting 10-year metastasis.

Figure 4: ROC curves of the best-performing models for all methods each respectively for predicting 15-year metastasis.

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 29 June 2022 doi:10.20944/preprints202206.0394.v1

Figure 5: Boxplots to compare the mean test AUCs of all methods side by side, each dataset separately and all datasets combined.

Figure 6: Side by side comparisons of the mean test AUCs when predicting 5, 10, and 15-year metastasis, for each of the methods respective.

TABLE LEGEND

Table 1: Case counts of the LSM datasets.

Table 2: The variables of the LSM datasets.

Table 3: Description of the DNN hyperparameters.

Table 4: The mean test AUCs and mean train AUCs of the best-performing models.

Table 5: Significance test results: one-tailed (greater) Wilcoxon rank sum tests.

Table 6: Significance test results: one-tailed (less) Wilcoxon rank sum tests.

Table 7: The hyperparameter values of the best-performing DNN models learned from 5-year, 10-year, and 15-year datasets, respectively.

Table 8: Experiment time per model per dataset, number of models trained, and total experiment time.