Statistical Considerations When Combining Artificial Intelligence-Enabled Diagnostic Devices

Manasi Sheth[1], Daniel Erchul[1] , Gene Pennello[1]
[1]Food and Drug Administration, 10903 New Hampshire Ave, Silver Spring, MD 20993

Abstract
Artificial intelligence (AI) is increasingly being incorporated into medical devices in efforts to provide results used alone or with other information to help assess a subject's present or future state of health. Increased computing power, greater availability of data, and the availability of deep learning and other AI methods have opened many opportunities for developing AI-enabled medical devices with increased capabilities. Performance validation of AI-enabled medical devices is essential in order to assess safety and efficacy. We focus on statistical methods to assess the combination of two or more AI diagnostic devices in order to improve diagnostic accuracy – e.g., sensitivity, specificity, positive and negative predictive value, diagnostic likelihood ratio. After providing a brief discussion of AI machine learning, including convolutional neural networks with application to diagnostic medical imaging, we present some basic probabilistic considerations and statistical techniques regarding combination of two binary-output AI-enabled devices in order to produce combined outputs with improved performance. We also review common methods for calculating a confidence interval on an accuracy proportion, e.g., sensitivity. Our goal is to provide information to non-statistician device developers and device users for their consideration of ways to increase clinical decision accuracy.

Key Words: Medical Devices, Endpoints, Machine Learning, Combination Rule, Convolutional Neural Network, Diagnostic Test Accuracy

1. Introduction

Adaptive computational (i.e. computerized "machine learning") models, if incorporated into medical devices, offer great potential to improve patient outcomes in many areas of healthcare. Although a variety of adaptive computational techniques have existed for many decades, further refinement of neural networks such as multi-layered convolutional neural networks, increased computational power, and greater amounts of data, now allow such models to be considered by some measures to be as good as humans for certain tasks [1,2,3].

The United States Food and Drug Administration (FDA) is seeing a dramatic increase in interest from both medical device developers and healthcare providers in pursuing adaptive machine learning methods for more capable medical devices. Among the most advanced and powerful machine learning techniques is the neural network model, which uses adaptive weighting of computer-modeled "neurons" to produce functions that map input to a desired output. One important application of neural networks is mapping data into separate output classes. (See Figure 1.) One may consider the neurons in these models to be "networked" via their weighted connections to other neurons, in a way that

have similarities to how axons and dendrites of biological neurons connect to other biological neurons. Although the detailed ways of how computer neural networks (that generally use integrated circuit [IC] chips) learn and operate can be substantially different from the way biological neural networks (e.g. the human brain), both computer neural networks and the human brain include neuron units that adapt (i.e. learn) in order to produce a desired result. Other "non-neural network" machine learning techniques also involve parameters that are adjusted/learned. For these techniques however, the adaptable parameters were generally not called "neurons". While some may use the terms "artificial intelligence", "AI", and "machine learning" interchangeably, others take great care when deciding which term to use, since for example, not all adaptive models use neural network "artificial intelligence" architectures [4,5].

Deep learning is a type of machine learning based on artificial neural networks, in which multiple layers of processing are used to extract progressively higher level features from data. The increase in efforts to incorporate deep learning into imaging devices is one important area of interest for many involved with medical device use and development. One important and powerful example is the convolutional neural network (CNN or "ConvNet"), a type of a deep learning architecture that is being incorporated into medical imagers, including computed tomography (CT) imagers, magnetic resonance (MR) imagers, ultrasound imagers, retinal imagers, and devices to analyze skin lesions, the latter for example, to classify skin lesions as cancerous or non-cancerous The latter is also an example where devices have been designed so that either a licensed healthcare provider or a layperson can take an image of skin lesion, and then send the image to an "AI server" in the computer server cloud which then provides its skin condition classification. In addition to image analysis, AI-enabled devices can be used to analyze signals in the time domain, for example in electrocardiograms (ECG), electroencephalograms (EEG), lung sounds, etc. Speech recognition is another example where AI can be used to analyze time-domain signals. While one may consider speech analysis to be very different than ECG, EEG, lung sounds, etc., often many of the basic concepts involved can be similar with regard to producing AI models of these time-domain signals.

Binary output (or "dichotomous") decisions – e.g.., positive test result/decision to take a clinical action or negative test result/decision to not take a clinical action – are a routine part of clinical practice. More generally, classification is not a new problem in statistics. A variety of approaches for classification are available [6], including linear discriminant analysis [7], regression trees, artificial neural networks [8], [9] and nearest neighbor techniques. While AI-enabled devices can be used to classify outputs into more than just two categories (i.e. binary output classification), given the importance of binary test results used by healthcare, this article focuses mainly on statistical validation considerations for binary-output AI-enabled devices.

We focus on statistical measures for combining AI-enabled devices, with the goal of appropriate combinations in order to produce results that increase measures of diagnostic accuracy such as sensitivity, specificity, positive predictive value (PPV), negative predictive values (NPV), positive- and negative-likelihood rations, and area under the ROC curve. We believe that this focus with regard to combining separate AI-enabled devices is important at least for the following reasons: While many AI-enabled device developers may focus on methods to produce a single AI-enabled device that uses multiple data types as input (e.g., a device with both imaging and time-domain inputs), there are situations where such a single device is not the optimal choice or is not available

to the decision maker. While our examples involve combination of neural network device outputs, many of the statistical concepts discussed in this article can also be used for combining outputs from devices other than AI-enabled devices.

1.1 Neural Networks

A gentle introduction to neural networks that includes R programming code is Chapter 8 in [12], which is freely available at http:/www.comp-approach.com. Here, we present a brief overview on neural networks that is hoped to provide helpful information to those new to the subject. To point out an important theoretical concept on the power of artificial neural networks, we encourage the reader to gain some familiarity with the "universal approximation theorem" [20-22].

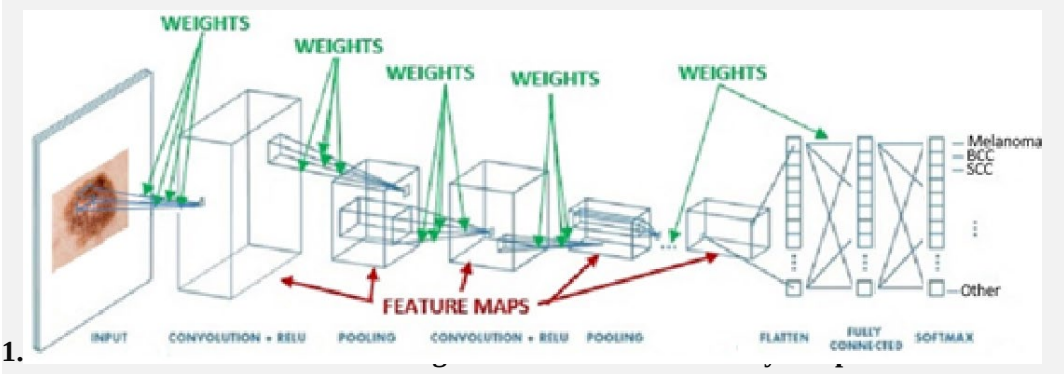
Neural networks are algorithms that utilize weighted "neural connections" of "neurons" to map input datapoints to desired output datapoints. For many computer neural network models, one may consider a neuron as an element which takes a weighted sum of its inputs and uses this weighted sum as input to a mathematical function, such as a sigmoid function, hyperbolic tangent function, step function, piecewise linear function (eg. Relu function), etc. The output of this neuron can then serve as an input into the next layer of neurons, and so on. In a fully-connected neural network, the outputs of each and every neuron serve as inputs to each and every neuron of the next layer. In some cases, this may work sufficiently, while in other cases, all of the connections are not present in the initial architecture by design, or are eliminated during the process of adjusting the model parameters (e.g. pruning of connections). Neurons in a neural network also often include weighted "bias" inputs to neurons that do not have inputs from other neurons. A bias input provides a constant input to its respective neuron. The numbers of layers, the numbers of neurons in the layers, the type of output functions that the neurons use, etc. all can vary for the particular task. In binary-output AI models, a single output layer neuron is often used, consisting of a weighted sum of neuronal outputs from the previous layer, which we refer to as an "index", which is then inputted to a sigmoid function. In our discussion, we consider models where a threshold value can be applied to the index values that are produced when the trained model is presented with input samples. The threshold serves as a separation point to determine whether the index value produced will be considered a positive or negative test output. In our discussion, the threshold value can be varied with the goal of producing the best desired output characteristics.

There are a variety of techniques used to adjust (i.e. "train") the parameters (e.g. weights) in order to map the input to the desired output. We refer the reader to the literature for a detailed discussion on this subject, and only briefly mention a "supervised learning" approach. For simplicity, one may wish to consider supervised learning as adjusting the parameters by first using a training data set with known inputs and known outputs. This input data is fed initially into an untrained network (e.g. with random initial weights), which first likely does not produce the desired output. The difference (i.e. error) in the actual output and the desired output is a function (i.e. error function) involving the neuronal weights and the non-linear functions in the various layers of the network. The variables are then adjusted with the goal of reducing the error to a minimum, or to find minima or maxima of other functions in order optimize the desired results. Again, there are many specific methods, and we refer the reader to the literature. The adjustment of the model's variable parameters typically takes numerous iterations, or epochs, of adjustment to the weights. While cloud servers or optimized hardware can be used by

deep-learning device developers, even some commonly-available laptop computers have sufficient computational speed to train some useful deep learning neural network classifiers. After training, the network is tested with a test set to validate the device's model and accuracy.

A convolutional neural network (CNN or "ConvNet" – See Figure 1) is a type of neural network (typically a "deep learning" multiple layer neural network model) that contains "convolutional filters". We refer the reader to the internet and literature with many sources of detailed information regarding CNNs. However, for a basic example you may wish to consider the following example explanation of at least some CNN components: An input layer convolutional filter typically multiplies individual pixel (or "voxel") values of a small area of the image by the filter's weights. The results of the individual multiplications are summed and then fed into a mathematical function (e.g. Relu function) in order to produce a value (typically a voxel element) in the next neural network layer. The convolutional filter is then moved to another location on the image, where the filter weights are again multiplied by the pixel values of the new local area on the image, and again summed and fed into the output function to produce another voxel element value. This continues until the image is scanned. The new layer of voxels is called a "feature map" layer. It is not unusual to have many convolutional filters which in turn produce many feature-map layers. There can be other operations involved, including taking averages and medians of area of feature maps, moving (i.e. "rastering") the convolutional filters by steps greater than one along the input image or feature maps), etc. (4, 23). At some layer in the CNN, often the feature maps are "unwound" ("flattened") into a single layer which may then be connected to successive connected layers with neuron weights, etc. Finally, for binary output CNN devices, the output layer are typically fed as a linear combination (again which we refer to as an "index") and often into a single sigmoid function, which is thresholded to produce a binary output. During training, images are presented to the network, so that the weights can be adjusted (i.e. learn) to minimize the error function, or to find minima or maxima of other functions (which are functions of the network's convolution filter weights, neuron output function [e.g. sigmoid, Relu, etc.], etc.) with a goal of optimizing production of correct binary outputs (e.g. "0" for a negative disease condition and "1" for a positive disease condition). After training the network is tested with a test set to validate the device's accuracy.

Figure 1: Example of a Convolutional Neural Network (CNN, ConvNet)



1.

When a binary-output device needs to be assessed for its accuracy, a common practice is for the device to be tested in a prospective study on samples or subjects that have been

"ground truthed" using a "gold standard", or reference method to be either positive or negative for the health condition of interest. The device is tested with the reference positive and negative samples to determine the fractions (percentages) of each that the device determines correctly. The analyses are often characterized with a table that is similar to the one shown in Table 1.

Table 1: 2 x 2 Confusion Matrix for Performance of a Diagnostic Test

		True Conditions		
		Reference Positive	Reference Negative	Row Totals
Test Results	Positive Test Result	TP	FP	TP + FP
	Negative Test Result	FN	TN	FN + FP
	Column Totals	TP + FN	FP + TN	

Reference standard samples tested by the device generate test results assigned to one of the four cells labeled True Positive (TP), False Positive (FP), True Negative (TN), and False Negative (FN) in Table 1. The device's sensitivity and specificity estimates are as follows:
Sensitivity = [TP / (TP + FN)] x 100%
Specificity = [TN / (FP + TN)] x 100%
Test Accuracy or Overall Agreement = [(TP + TN)/(TP + FN + FP + TN) x 100%
Positive predictive value (PPV) = [TP / (TP + FP)] x 100%
Negative predictive value (NPV) = [TN / (FN + TN)] x 100%

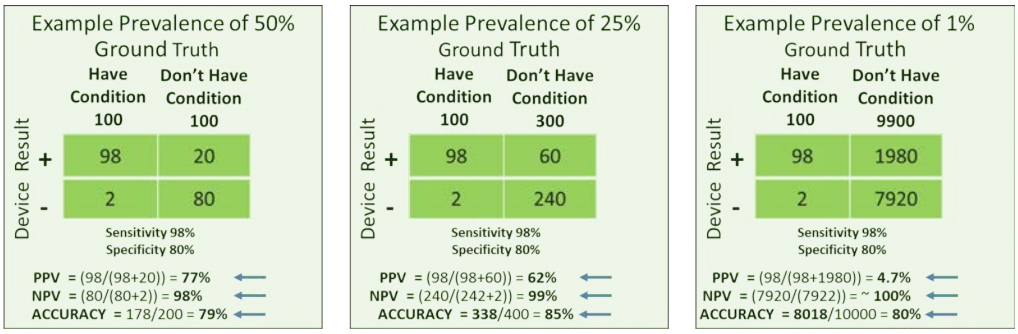
These point estimates are usually cited as fractions or percentages. Confidence intervals for these performance measures can be derived from the test results and are essential for quantifying estimation uncertainty. Given the variety of statistical methods, distributional characteristics, and study designs that may be involved, we later describe some common methods for obtaining confidence interval for these measures.

The sensitivity of a device is the probability that it returns a positive test results on a subject with the health condition (i.e. as determined by the reference). Specificity of the device is the probability that it returns a negative test result on a subject without the health condition. In a clinical study, the sample estimate of sensitivity is the proportion of subjects with the condition who test positive. In probability notation, P (T = 1 | D = 1) is the population parameter for the estimated sensitivity where T = test condition {1 = if positive, 0 = otherwise}, and D = Disease condition {D = 1 if disease present, 0 otherwise}. Similarly, specificity is a measure of the probability of a testing device to return a negative test result when the device has been given a sample that has been confirmed to be negative. In probability notation, P (T = 0 | D = 0) is the population parameter for the estimated specificity.

The positive predictive value (PPV) is a measure of the post-test probability of the condition being present, given that a positive test result was returned by the device. In probability notation this is P (D = 1 | T = 1). Similarly, the negative predictive value (NPV) is a measure of the post- test probability of the condition not being present, given

that a negative test result was returned by the device. In probability notation, this is P (D = 0 | T = 0). We note that with PPV, NPV, and accuracy, it is essential that prevalence, the a priori probability of the condition, be included. In Figure 2 we show examples of how prevalence can affect PPV, NPV, and accuracy below with the same sensitivity and specificity.

Figure 2: Examples of Prevalence Affecting PPV, NPV and Accuracy



Another important pair of measures are the diagnostic likelihood ratios. The positive likelihood ratio (PLR) is defined by Sensitivity divided by (1 – Specificity), and negative likelihood ratio (NLR) is defined by (1 – Sensitivity) divided by Specificity of a test. By Bayes theorem, PLR confers the relative change in the odds of disease given a positive result compared with the pre-test odds p / (1 – p), where p = prevalence of the disease. Thus, a PLR of 10 means that the odds of disease are 10 times greater for subjects who test positive compared to the odds in the population (i.e.. the pre-test odds). Likewise, NLR confers the relative change in the odds of disease given a negative result compared with the pre-test odds p / (1 – p), where p = prevalence of disease. Thus, an NLR of 0.1 means the odds of disease are 10 times less for subjects who test negative than the odds in the population.

2. Combining Devices

2.1 Considerations Regarding Correlation of Neural Network Output Layers

While continuing refinement of AI devices has led to increased accuracy, many AI devices still are substantially less than 100% sensitive and 100% specific. To improve performance, AI device developers may wish to consider using data from multiple data modalities for input into one AI device. For example, a device developer may have a camera to take images of the skin lesions, and also a non-imaging infrared spectral analyzer which gathers infrared spectra for some sample points on lesions. In addition, the developer has training data set for positive and negative skin cancer lesions with images and spectra for each sample. The developer devises and trains an AI-enabled device that can use both the images and spectra for the input to a single deep learning neural network with a binary output (positive or negative for skin cancer). However, there are situations where the data is at physically separate locations, and the device may not have access to both sets of data. For this situation with a single device with two modalities for inputs (e.g. image and spectroscopy), it still may be possible to train a single neural network for situations where data is missing, and that such a network can

still perform with only one data modality. Further, one may argue that such a large single device with potentially more interconnected neurons may achieve better performance than the combination of two separate neural networks. Thus one might conclude that considering combination of separate devices is not particularly worthwhile. However, as a matter of reliability, there is still is the potential situation where one single device (e.g., one physical device that has two data modalities for input) is unable to operate (e.g., power failure, device malfunction, device is damaged, etc.) or provide a useful result (e.g., due to missing input data) . With two separate neural networks however, if only one device goes into a failure mode, the other device can still provide its standalone classification output. Further, there are healthcare providers/device users who will only have separate devices, and there are developers who only will develop devices for one modality. Indeed too, developers may only focus in one technology area, for example in image analysis, and only develop their devices to classify images, while other developers may also be experts in another modality, for example with infrared spectral analysis. So not only can users (who find themselves with separate devices) and their patients potentially benefit from increased classification accuracy from an analysis of combining outputs from separate devices, but device developers may also find their devices in higher demand from the combined increased performance.

We note here that our discussion focuses on the situation for improved combined performance where both devices are always available to report their results. Otherwise, when only one device is present with the strategies we will discuss, the sensitivity and specificity is that for one single device. We leave it to the reader to consider other strategies such as serial strategy where the second device's output may or may not be requested, pending the output of the first device. One may also wish to consider such a serial strategy along with the second device's reliability, for example if the second device is available less than 100% of the time to provide a result after being requested to do so. Again, at the present we leave analysis of such strategies for the reader to consider.

Figure 3: Examples of conditionally uncorrelated, conditionally positively-correlated, and conditionally negatively-correlated index values. The index values are inputs to the devices' binary output function (e.g. sigmoid function).

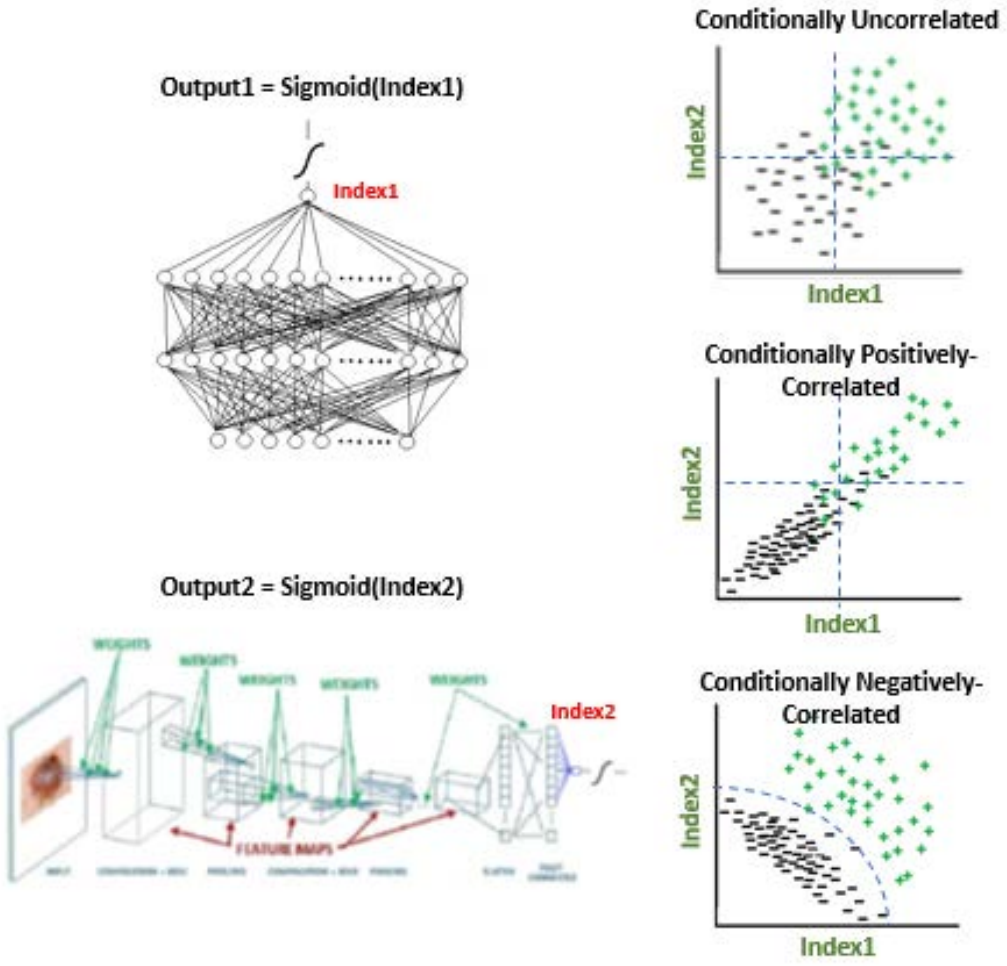


Figure 3 shows two separate neural network devices. As an example, device 1 uses infrared spectra of skin lesions as input, and it is trained with the goal to produce an output of 1 when the spectra presented is from a skin lesion that is positive for skin cancer, and to produce an output of 0 when the spectra presented is from a skin lesion that is negative for skin cancer. Device 2 uses images of skin lesions as input, and is trained with the goal to also produce an output of 1 when the image is from a skin lesion that is positive for skin cancer and to produce an output of 0 when the image is from a skin lesion that is negative for skin cancer.

2.1 Performance Metrics for Combining Devices/Tests

When evaluating performance of two devices, we consider it essential to know the strategy being utilized for the system, e.g. parallel or series. In a parallel strategy, a positive result from either device 1or device 2 is considered a positive combined result. This may also be referred to as a "believe the positive" strategy. Also with the parallel strategy, both devices must return a negative for the combined result to be considered negative. Two tests are said to be in series when both tests need to be positive in order for the combined result to be considered positive. With this latter serial strategy, only one of the two devices needs to return a negative result for the combined result to be

considered negative. Thus, this series strategy may also be referred to as a "believe the negative" serial strategy.

Above we have introduced the basic performance measures derived from the confusion matrix. The confusion matrix for a binary output device is a two-by-two table (See Table 1 above.) that contains four outcomes. Sensitivity and specificity are derived from the confusion matrix. Receiver operating characteristic (ROC) curves can be generated from this 2 x 2 confusion matrix from binary output devices by varying the output threshold. That is, the dividing point on the index scale that divides positive outputs from negative outputs. When the true positives and true negatives are weighted by prevalence, other important measure such as positive predictive value (PPV), negative predictive value (NPV), and accuracy are also derived from the 2 x 2 confusion matrix. Estimates for the overall sensitivity (sometimes called "net sensitivity") and overall specificity (sometimes called "net specificity") for the two tests in combination can be obtained using probability and statistical methods.

2.1.1. "Series" Strategy Combination of Device Outputs for Sensitivity

As mentioned above, sensitivity is an estimate of the ability of the device to correctly produce a positive result when a positive sample confirmed by a gold standard method has been given to the device. The sensitivity of applying tests A and B in series is represented by the Table 2. Algebraically, combined sensitivity for A and B in series equals sensitivity of A times sensitivity of B. In probability notation, this is $Pa(T = 1 \mid D = 1) \times Pb(T = 1 \mid D = 1)$ where subscript a indicates device A and subscript b indicates device B.

Table 2: Data Matrix for Systems in Series

Device A	Device B	Test Result
Positive	Positive	Positive
Positive	Negative	Negative
Negative	Positive	Negative
Negative	Negative	Negative

2.1.2. Parallel Combination of Device Outputs for Sensitivity

If, instead, tests A and B are applied in parallel, so that a positive result on either test causes the overall result is to be classified as positive, then the combined sensitivity for the combination is sensitivity of A + sensitivity of B – sensitivity of A x sensitivity of B, assuming the two tests are conditionally independent (uncorrelated) among subjects with the health condition. In probability notation, this is $Pa(T = 1 \mid D = 1) + Pb(T = 1 \mid D = 1) - Pa(T = 1 \mid D = 1) \times Pb(T = 1 \mid D = 1)$. Conditional independence may be plausible if tests A and B are based on different technologies and different biological input modalities, for example, combining lung sound spectral signals, respiration rate, oxygen saturation, skin temperatures, and previous lung x-ray scans, to screen for a respiratory condition.

Table 3: Data Matrix for Parallel Systems

Device A	Device B	Test Result
Positive	Positive	Positive
Positive	Negative	Positive

Negative	Positive	Positive
Negative	Negative	Negative

2.1.3 Correct classification of non-cases – combining specificities:

Specificity evaluates the ability to identify non-cases. If A and B are applied in parallel, then only the non-cases that are correctly classified by both tests will be considered a negative result in the combined classification.

In order to have a correct classification of non-cases with two tests read in parallel, both tests must be negative. So, the overall probability of correct classification of non-cases (the overall specificity) from applying tests A and B is in parallel is Specificity of A x Specificity of B. In probability notation, this is Pa(T = 0 | D = 0) x Pb(T = 0 | D = 0).

If instead tests A and B are applied in series, only one correct negative result from either test is needed for the overall result to be classified as negative. Thus, the specificity of the combination is represented algebraically by Specificity of A + Specificity of B – (Specificity of A x Specificity of B). In terms of probability, this is Pa(T = 0 | D = 0) + Pb(T = 0 | D = 0) – Pa(T = 0 | D = 0) x Pb(T = 0 | D = 0), assuming the tests are conditionally independent among subjects without the health condition.

Below is a table that provides combined sensitivities and combined specificities for different thresholds for two example devices. From the examples in Table 4, one can observe that a desired combined sensitivity and combined specificity can depend not only on the specific thresholds, but also on the particular strategy.

Table 4: Combining Deep Learning Outputs – Changing Thresholds

Threshold	Device 1		Device 2		Parallel Strategy		Serial Strategy	
	Sens	Spec	Sens	Spec	Combined Sens	Combined Spec	Combined Sens	Combined Spec
1	0.90	0.80	0.90	0.80	0.99	0.64	0.81	0.96
2	0.90	0.55	0.90	0.60	0.99	0.33	0.81	0.82
3	0.70	0.85	0.70	0.95	0.91	0.81	0.49	0.99
4	0.95	0.60	0.95	0.55	0.9975	0.33	0.90	0.82

However, when combining devices using convolutional neural networks and determining their combined performance measures, there are three possibilities; the two systems could be (1) conditionally uncorrelated (independent) (2) conditionally correlated, and (3) conditionally negatively-correlated. In each of these possibilities, the performance metrics may be computed differently than the traditional methods. The first of the possibilities has been presented above, the latter two will be presented elsewhere.

2.1.1 Confidence Intervals Procedures for Net Performance Measures

A binomial proportion confidence interval is a confidence interval for the probability of success calculated from the outcome of a series of Bernoulli trials. In other words, a

binomial proportion confidence interval is an interval estimate of a success probability p when only the number of experiments n and the number of successes x's are known.

There are several formulae for a binomial confidence interval, but all of them rely on the assumption of a binomial distribution. In general, a binomial distribution applies when an experiment is repeated a fixed number of times, each trial of the experiment has two possible outcomes (success and failure), the probability of success is the same for each trial, and the trials are statistically independent. To calculate this confidence interval, a variety of large sample approximations and exact methods are commonly used, all with their own trade-offs in accuracy and computational intensity.

A commonly used formula for a binomial confidence interval relies on approximating the distribution of error about a binomially-distributed observation, $\hat{p}$, with a normal distribution. This approximation is based on the central limit theorem and is unreliable when the sample size is small or the success probability is close to 0 or 1. Using the normal approximation, the success probability (i.e. combined sensitivity) is estimated as $\hat{p} \pm z\sqrt{\frac{\hat{p}(1-\hat{p})}{n}}$. This normal approximation interval is sometimes even called the Wald's interval. Although easy to compute the Wald interval for a binomial proportion can have surprisingly poor coverage compared with the nominal confidence level even in moderately sized samples [13].

The Wilson score interval [11] is an improvement over the normal approximation interval in that the actual coverage probability is closer to the nominal value. The success probability p is estimated as $\frac{\hat{p}+\frac{z^2}{2n}}{1+\frac{z^2}{n}} \pm \frac{z}{1+\frac{z^2}{n}}\sqrt{\frac{\hat{p}(1-\hat{p})}{n}+\frac{z^2}{4n^2}}$. This interval has good properties even for a small number of trials and/or extreme probability.

The Clopper-Pearson interval is a very common method for calculating binomial confidence intervals. This is often called an 'exact' method because it is based on the cumulative probabilities of the binomial distribution (i.e. exactly the correct distribution rather than an approximation). Because of a relationship between the binomial distribution and the beta distribution, the Clopper-Pearson interval is sometimes presented in an alternate format that uses quantiles from the beta distribution. $B\left(\frac{\alpha}{2}; x, n-x+1\right) < \theta < B(1-\frac{\alpha}{2}; x+1, n-x)$ where x is the number of successes, n is the number of trials, and B(p, v, w) is the pth quantile from a Beta distribution with shape parameters v and w. The Clopper-Pearson interval is an exact interval since it is based directly on the binomial distribution rather any approximation to the binomial distribution. This interval never has less than the nominal coverage for any population proportion, but that means that it is usually conservative. For example, the true coverage rate of a 95% Clopper-Pearson interval may be well above 95%, depending on n and θ. Thus, the interval may be wider than it needs to be to achieve 95% confidence.

Suppose, in a sample of 114 diseased patients, net sensitivity is 98.25% (112/114). Then the Clopper-Pearson interval for this estimate is (93.8%, 99.8%) and the Wilson's score interval is (93.8%, 99.5%). Below the table shows the variety of confidence intervals for estimates for net sensitivity. Similar approach can also be used to compute confidence interval for net specificities.

For positive and negative likelihood ratios, the Wald confidence interval method is described in [14]. A score method is described in [15].

Table 5: Confidence Interval Procedures for Net Sensitivity Estimates

Combined Sensitivity Estimate	95% Wald's Confidence Interval	95% Clopper-Pearson Confidence Interval	95% Wilson's Score Interval
4.00% (5/114)	(0.63%, 8.15%)	(1.44%, 9.94%)	(1.89%, 9.86%)
21.93% (25/114)	(14.33%, 29.53%)	(14.72%, 30.65%)	(15.32%, 30.37%)
39.47% (45/114)	(30.50%, 48.45%)	(30.45%, 49.06%)	(30.98%, 48.65%)
57.02% (65/114)	(47.93%, 66.11%)	(47.41%, 66.25%)	(47.85%, 65.73%)
74.56% (85/114)	(66.57%, 82.56%)	(65.55%, 82.25%)	(65.86%, 81.66%)
92.11% (105/114)	(87.16%, 97.06%)	(85.54%, 96.33%)	(85.67%, 96.79%)

3.0 Concluding Remarks

There are some quantitative tests with two cut-offs/thresholds instead of a single threshold based on the clinical context where the need was to have a high rule-in and rule-out claim. However, these thresholds need to be fixed prior to clinical validation studies to evaluate performance of the test in a clinical study. It should be noted that this also applies when combining two tests/devices since combining the data prior to cut-off may or may not result in same distributional assumption as with one test.

We have focused on combining two tests in parallel or serially that are assumed to be conditionally uncorrelated. Conditional independence might be plausible if the two tests are based on different technologies, concepts, and inputs. For example, to diagnose presence or absence or prostate cancer, the assumption that a prostate specific antigen test is conditionally independent of a digital rectal exam by a physician is plausible. However, regardless of plausibility, the condition independence assumption needs to be evaluated for adequacy in any given study. The conditional independence assumption could be exploited in serial combinations of tests to obtain confidence intervals on combined sensitivity and combined specificity, but was not explored.

In a prospective study of diagnostic test accuracy, sample estimates PPV and NPV are unbiased, assuming that the study is well-conducted. However, for health conditions of low prevalence, many if not most diagnostic accuracy studies are retrospective so that the study can be enriched with subjects having the condition. In retrospective studies, PPV and NPV are distorted because prevalence is much higher in the study than in the population intended for the test. For retrospective enriched studies, an external estimate of prevalence can be combined with study estimates of sensitivity and specificity to obtain valid estimates of PPV and NPV via Bayes Theorem. Alternatively, study estimates of positive and negative likelihood ratios are unbiased in retrospective enriched studies and, as mentioned, confer the relative change in the odds of having the condition given the test result compared with the pre-test odds.

We have considered believe-the-positive (BTP) and believe-the-negative (BTN) combinations of tests in parallel or serially, with a focus on AI-enabled tests. BTP and BTN combinations are special cases of logic rules for combining two or more test results [16] – [19]. For example, Wolf et al [19] considers ordinal logic regression.

Acknowledgements

We would like to show our gratitude to Dr. Bipasa Biswas for sharing her insight. We would also like to thank our colleagues in the Office of Clinical Evidence and Analysis in Center for Devices and Radiological Health for their valuable discussion.

.

References

[1] Hinton G. (2018) Deep Learning - A Technology With the Potential to Transform Health Care. Journal of the American Medical Association. 320(11), 1101–1102
[2] Kiani, A., Uyumazturk, B., Rajpurkar, P. et al. (2013) Impact of a deep learning assistant on the histopathologic classification of liver cancer. npj Digital Medicine 3, 23
[3] LeCun, Y., Bengio, Y. & Hinton, G. (2015) Deep learning. Nature 521, 436–444
[4] Schank, R. C. (1987). What Is AI, Anyway?. AI Magazine, 8(4), 59
[5] Du-Harpur X, Watt FM, Luscomb NM, Lynch MD (2020) What is AI? Applications of artificial intelligence to dermatology British Journal of Dermatology (2020) 183, 423–430
[6] Dudoit S., Fridland, J. and Speed 2000 TP. Comparison of Discrimination Methods for the Classification of Tumors Using Gene Expression Data. Technical Report 576, University of California, Berkeley.
[7] Su and Liu (1993). Linear Combinations of multiple diagnostic markers. Journal of American Statistical Association, 88, 1350 – 1355.
[8] McIntosh, M & Pepe, M.S. (2002). Combining Several Screening Tests: Optimality of the Risk Score, Biometrics 657-664.
[9] Zhang et al. (1999). A Simple Statistical Parameter for Use in Evaluation and Validation of High Throughput Screening Assays, Journal of Biomolecular Screening, 4(2), 67-73.
[10] Pepe, Margaret. The Statistical Evaluation of Medical Tests for Classification and Prediction, Oxford University Press 2003.
[11] Wilson, E.B. (1927). Probably inference, the law of succession, and statistical inference, Journal of the American Statistical Association, 22, (158), 209- 212.
[12] Arnold, T. Kane, M. Lewis BW. A Computational Approach to Statistical Learning, CRC Press, 2020.
[13] Brown, LD, Cai TT, DasGupta A, Interval Estimation for a Binomial Proportion. Statist. Sci. 16(2): 101-133 (May 2001). DOI: 10.1214/ss/1009213286.
[14] Simel DL, Samsa GP, Matchar DB. Likelihood ratios with confidence: sample size estimation for diagnostic test studies. J. Clin Epidemiol. 1991; 44(8): 763 - 70. DOI: 10.1016/0895-4356(91)90128-v.PMID:1941027
[15] Gart JJ., Nam J. Approximate interval estimation of the ratio of binormal parameters: A review and corrections for skewness. Biometrics 1988; 44: 323-328.
[16] Baker SG. Identifying combinations of cancer markers for further study as triggers of early intervention. Biometrics. 2000 Dec; 56(4): 1082 - 7. DOI: 10.1111/j.0006-341x2000.01082.x.PMID:11129464

[17] Etzioni R. Kooperberg C. Pepe M. Smith R. Gann PH. Combining biomarkers to detect disease with application to prostate cancer. Biostatistics. 2003 Oct; 4(4): 523-38.DOI: 10.1093/biostatistics/4.4.523. PMID: 14557109.
[18] Wang MC. Li S. Bivariate marker measurements and ROC analysis. Biometrics. 2012. Dec; 68(4): 1207 - 18. DOI: 10.1111/j/1541-0420.2012.01783.x. Epub 2012 Sep 24. PMID: 23005264; PMCID: PMC3533066.
[19] Wolf BJ, Slate EH, and Hill EG. Ordinal Logic Regression: A classifier for discovering combinations of binary markers for ordinal outcomes. Comput Stat Data Anal. 2015. February 1; 82: 152-163.
[20] Cybenko, G. Approximation by superpositions of a sigmoidal function. Math. *Control Signal Systems* 1989; 2, 303–314. https://doi.org/10.1007/BF02551274.
[21] Husmeier D., Allen D., Taylor J.G. (1997) A Universal Approximator Network for Learning Conditional Probability Densities. In: Ellacott S.W., Mason J.C., Anderson I.J. (eds) Mathematics of Neural Networks. Operations Research/Computer Science Interfaces Series, vol 8. Springer, Boston, MA. https://doi.org/10.1007/978-1-4615-6099-9_32.
[22] Russell G, Fausett LV. Comparison of function approximation with sigmoid and radial basis function networks, *Proc. SPIE 2760, Applications and Science of Artificial Neural Networks II*, 22 March 1996; https://doi.org/10.1117/12.235903
[23] S. Albawi, T. A. Mohammed and S. Al-Zawi, "Understanding of a convolutional neural network," 2017 International Conference on Engineering and Technology (ICET), 2017, pp. 1-6, doi: 10.1109/ICEngTechnol.2017.8308186.