Optimizing Computer-Aided Diagnosis with Cost-Aware Deep Learning Models

Charmi Patel[1†], Yiyang Wang[2], Thiruvarangan Ramaraj[1], Roselyne Tchoua[1], Jacob Furst[1], Daniela Raicu[1]

[1] *DePaul University, Chicago, IL, 60604, U.S.A*
[2] *Milwaukee School of Engineering, Milwaukee, WI, 53202, U.S.A*
[†] *E-mail: cpatel54@depaul.edu*

Classical machine learning and deep learning models for Computer-Aided Diagnosis (CAD) commonly focus on overall classification performance, treating misclassification errors (false negatives and false positives) equally during training. This uniform treatment overlooks the distinct costs associated with each type of error, leading to suboptimal decision-making, particularly in the medical domain where it is important to improve the prediction sensitivity without significantly compromising overall accuracy. This study introduces a novel deep learning-based CAD system that incorporates a cost-sensitive parameter into the activation function. By applying our methodologies to two medical imaging datasets, our proposed study shows statistically significant increases of 3.84% and 5.4% in sensitivity while maintaining overall accuracy for Lung Image Database Consortium (LIDC) and Breast Cancer Histological Database (BreakHis), respectively. Our findings underscore the significance of integrating cost-sensitive parameters into future CAD systems to optimize performance and ultimately reduce costs and improve patient outcomes.

Keywords: Misclassification errors; Cost-sensitive activation function; Convolutional neural network

1. Introduction

Machine learning (ML) models have been developed to identify patterns in data across various domains, including computer-aided diagnosis,[1] public health,[2] and defect detection.[3] Generally, these ML models are optimized based on overall prediction accuracy across all classes and data points, assuming that misclassification errors are equal.[4] However, this assumption can be perilous in classification problems where misclassifying a positive instance carries a higher cost than misclassifying a negative instance. Particularly in the medical domain, a false negative error will likely have much greater consequences than a false positive.

To address this challenge, we propose a novel cost-aware deep learning-based CAD system that incorporates different cost values into the activation function to boost the model's sensitivity. By fine-tuning the cost values associated with false positive and false negative instances, we can significantly increase true positives. Our contributions are twofold: 1) a CAD training framework designed to enhance sensitivity while maintaining overall accuracy, and 2) a proof-of-concept demonstrating the value of incorporating cost values as hyperparameters in future CAD systems.

2. Related Work

2.1. *Cost-Sensitive Learning in Non-Medical Domains*

In recent years, cost-sensitive learning has gained popularity as a valuable tool in non-medical domains to tackle class imbalance[4–10] and to address its associated costs of misclassification.

Prior research by Zhu and Wan[11] proposed a cost-sensitive learning method for semi-supervised hit-and-run analysis to handle the data imbalance issue which significantly improved model's performance even with a small proportion of labeled historical data. Khoshgotaar et al.[12] introduced cost-sensitive learning into Software Defect Prediction (SDP) and used a boosting method to build software quality models. Le et al.[13] implemented a hybrid approach by combining oversampling techniques and cost-sensitive learning, which significantly improved bankruptcy prediction performance. Devi et al.[14] proposed a cost-sensitive weighted random forest algorithm for effective credit card fraud detection. The model assigns more weight to minority instances during training, resulting in improved performance compared to existing random forest techniques. Xiao et al.[15] integrated a group handling neural network-based cost-sensitive semi-supervised selective ensemble model for credit-scoring problems.

Other prior work focused on improving the overall prediction performance by modifying the loss function to consider different cost values for various misclassifications. Li et al.[16] proposed a pixel-based adaptive weighted cross-entropy loss function to facilitate road crack detection. Wang et al.[17] also introduced a novel cost-sensitive loss function for semantic segmentation of remote sensing images. More recently, Li et al.[18] constructed a new cost-sensitive loss function that incorporates the cost difference caused by misclassification between different classes proving its ability to enhance the model's effectiveness.

2.2. *Cost-Sensitive Learning Applied to Medical Diagnosis*

Research studies on the ML application to medical diagnosis typically employ traditional ML algorithms and advanced algorithms via ensemble learning,[19] evolutionary algorithms,[20] sparse autoencoders (SAE).[21] However, few research works have conducted cost-sensitive learning in medical diagnosis. Recently, Manop[5] developed a cost-sensitive XGBoost model for breast cancer detection and evaluated it on four breast cancer datasets with uneven class distribution, achieving accuracy ranging from 95.99% to 96.43%. Ali et al.[6] developed a method that combines cost-sensitive learning and ensemble learning techniques to predict breast cancer. The ensemble learning methods include GentleBoost, Bagging, and Adaptive Boosting, resulting in a 3.91% improvement. Zieba et al.[7] proposed the combination of ensemble learning and cost-sensitive Support Vector Machine (SVM) to address the lung cancer patients' post-operation life expectancy. They observed that patients not covered by the minority rules have a 97% chance of surviving the considered survival period. Ali et al.[22] applied cost-sensitive ensemble methods in the classification of chronic kidney disease (CKD) which incorporates feature ranking capabilities instead of enhancing predictive accuracy. Cost-sensitive deep neural networks (CSDNN) have also been developed by Wang et al.[23] to predict hospital readmission, achieving a significant improvement in accuracy of 6% and 4% for 1-year and 30-day readmission prediction, respectively. Mienye and Sun[10] developed robust cost-sensitive classifiers

for predicting medical diagnosis by modifying the objective functions of algorithms such as logistic regression, decision trees, extreme gradient boosting, and random forest. They tested these classifiers on four medical datasets and demonstrated that cost-sensitive methods yield improvements ranging from 1% to 4% compared to the standard algorithms.

These works[5–7,10,23] suggest that incorporating cost-sensitive learning with specific cost values during misclassifications into ML models can improve overall classification performance. However, these studies focus solely on improving overall classification results without determining the impact of different cost values on sensitivity and specificity metrics. Our study expands the integration of the misclassification costs into the learning algorithm by optimizing the diagnostic interpretation sensitivity without sacrificing the overall diagnostic performance.

3. Methodology

3.1. *Convolutional Neural Networks*

A classical CNN model (Figure 1) comprises convolutional layers and fully connected layers. The convolutional layers are designed to extract features, while the fully connected layers are responsible for classification. For our study, we focus on the binary classification problem distinguishing between malignant and benign cases. To achieve this, we employed a single neuron in the output layer with the sigmoid activation function. The output of the sigmoid activation function determines the predicted label.

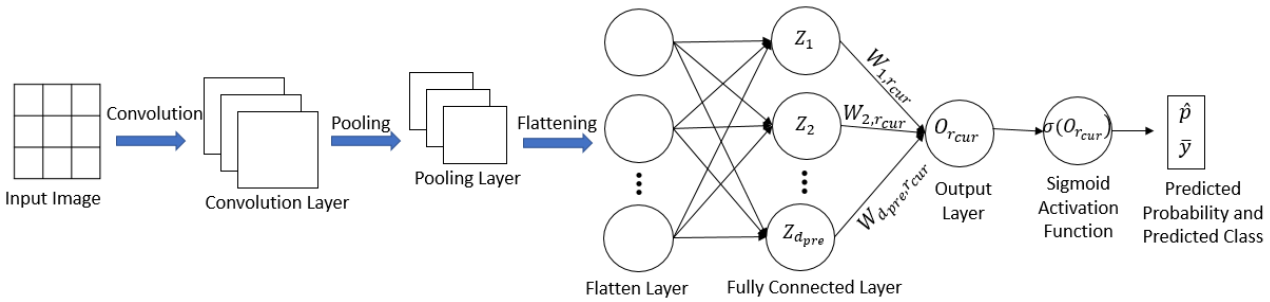

Fig. 1: CNN architecture overview.

Let $O^{(t)}_{r_{\text{cur}}}$ be the output of the current layer's neuron with index r_{cur} at epoch t for any image X_i:

$$O^{(t)}_{r_{\text{cur,i}}} = w^{(t)}_{0,r_{\text{cur,i}}} + \sum_{r_{\text{pre}}=1}^{d_{\text{pre}}} g(z^{(t)}_{r_{\text{pre}}}) w^{(t)}_{r_{\text{pre}} r_{\text{cur,i}}} \tag{1}$$

where $w^{(t)}_{r_{\text{pre}} r_{\text{cur}}}$ represents the weights between the neuron indexed as r_{pre} in the previous layer and the neuron indexed as r_{cur} in the current layer at epoch t. d_{pre} represents the number of neurons in the previous layer, and g represents the activation function applied to each output value of a neuron $z^{(t)}_{r_{\text{pre}}}$ from the previous layer r_{pre}.

Given an image X_i, where i represents the image index, we denote y_i as its actual label

and $\hat{p}_i$ as its prediction probability:

$$\hat{p}_i^{(t)} = \sigma(O_{r_{\text{cur,i}}}^{(t)}) = \frac{1}{1 + e^{-O_{r_{\text{cur,i}}}^{(t)}}} \tag{2}$$

where σ represents the sigmoid activation function applied to the output layer neuron. The predicted $\bar{y}_i$ is calculated as follows:

$$\bar{y}_i = \begin{cases} 1, & \text{if } \sigma(O_{r_{\text{cur,i}}}^{(t)}) \geq 0.5 \\ 0, & \text{if } \sigma(O_{r_{\text{cur,i}}}^{(t)}) < 0.5 \end{cases} \tag{3}$$

Using the actual label y_i and its predicted probability $\hat{p}_i$, the loss for instance X_i at epoch t is calculated using Binary Cross Entropy (BCE) loss:

$$Loss_{BCE_i}^{(t)} = -[y_i \log_2(\hat{p}_i) + (1 - y_i) \log_2(1 - \hat{p}_i)] \tag{4}$$

During the backpropagation, the model updates its weights to minimize the overall loss $L_{ERM}^{(t)}$ (Equation 5) at epoch t, which is the average loss across all N training instances.

$$L_{ERM}^{(t)} = \frac{1}{N} \sum_{i=1}^{N} Loss_{BCE_i}^{(t)}(\hat{p}_i) \tag{5}$$

where ERM denotes Empirical Risk Minimization.

3.2. *Cost-aware CNN model*

To direct the performance of the CNN model towards the true positives (malignant cases denoted by 1), we propose a cost-sensitive activation function $\sigma(O_{r_{\text{cur}}}^{(t)}, c(y, \bar{y})^{(t)})$ that penalizes more false negatives than the false positives by assigning higher costs to outcomes that are misclassifications of true positives and lower costs for misclassifying true negatives (benign cases denoted by 0). Grounded in the work by Li et al.,[18] we define an inverse relationship between the cost of false negatives $c(1,0)$ and the cost of false positives $c(0,1)$ and restrict the values of $c(1,0)$ to be greater than 1:

$$c(0,1) = \frac{1}{c(1,0)} \tag{6}$$

The activation function remains the same for correctly classified cases, and therefore, the costs for true positives $c(1,1)$ are equal to the costs for true negatives $c(0,0)$ and are equal to 1.

By integrating the cost values into the activation function, Equation (2) is transformed into Equation (7), denoting that the predicted probabilities $\hat{p}^{(t)}$ at epoch t are now influenced by the costs associated with each type of output:

$$\hat{\hat{p}}^{(t)} = \sigma(O_{r_{\text{cur}}}^{(t)}, c(y, \bar{y})^{(t)}) = \frac{1}{1 + e^{-O_{r_{\text{cur}}}^{(t)} \cdot c(y,\bar{y})^{(t)}}} \tag{7}$$

Implicitly, the new $L_{ERM\_Cost}^{(t)}$ loss function at epoch t, which is based on the predicted probabilities, will be dependent on the cost values:

$$L_{ERM\_Cost}^{(t)} = \frac{1}{N} \sum_{i=1}^{N} Loss_{BCE_i}^{(t)}(\hat{\hat{p}}_i) \tag{8}$$

3.3. *Cost Analysis w.r.t. Sensitivity and Specificity*

Maximizing both sensitivity and specificity simultaneously is not possible as they are inversely related.[24,25] However, introducing cost-values in the activation function of the outcome layer allows optimizing the performance of each without a decline in the overall accuracy. Here we illustrate two examples that show the impact of the costs on the performance of a CNN model.

3.3.1. *Handling False Negative Cases*

In a false negative case, where the actual label $y = 1$ and the predicted label $\bar{y} = 0$, to achieve $\bar{y} = 0$, $O_{r_{\text{cur}}}$ must be a negative value. Using Equations (2) and (4), and if $c(1,0) = 1$, which represents no cost value being used, we obtain the loss function:

$$Loss_{BCE_i^{(t)}} = -\left[\log_2\left(\frac{1}{1+e^{-O^{(t)}_{r_{\text{cur,i}}}}}\right)\right] \tag{9}$$

If we introduce $c(1,0) > 0$ in the activation function, which is a cost value associated with the false negative situation, we obtain the modified loss function:

$$Loss_{BCE\_Cost_i^{(t)}} = -\left[\log_2\left(\frac{1}{1+e^{-O^{(t)}_{r_{\text{cur,i}}}\cdot c(y,\bar{y})^{(t)}}}\right)\right] \tag{10}$$

$$Loss_{BCE\_Cost_i^{(t)}} > Loss_{BCE_i^{(t)}} \tag{11}$$

After introducing a cost value of $c(1,0)$, the loss value obtained from Equation (10) demonstrates an increase, as shown in Equation (11). This cost value impacts the 'false negative' case during the training process, leading to a decrease in false negative cases and an increase in sensitivity.

3.3.2. *Handling False Positive Cases*

In a false positive case, where $y = 0$ and $\bar{y} = 1$, to achieve $\bar{y} = 1$, $O_{r_{\text{cur}}}$ must be a positive value. If $c(0,1) = 1$, which means no cost values being used, we obtain the loss function:

$$Loss_{BCE_i^{(t)}} = -\left[1 - \log_2\left(\frac{1}{1+e^{-O^{(t)}_{r_{\text{cur,i}}}}}\right)\right] \tag{12}$$

If we introduce $c(0,1) = \frac{1}{c(1,0)}$ with $c(1,0) > 1$, which represents a cost value associated with the false positive situation, we obtain the modified loss function:

$$Loss_{BCE\_Cost_i^{(t)}} = -\left[1 - \log_2\left(\frac{1}{1+e^{-O^{(t)}_{r_{\text{cur,i}}}\cdot c(y,\bar{y})^{(t)}}}\right)\right] \tag{13}$$

$$Loss_{BCE\_Cost_i^{(t)}} < Loss_{BCE_i^{(t)}} \tag{14}$$

After introducing a cost value of $c(0,1)$, the loss value obtained from Equation (13) demonstrates a decrease, as shown in Equation (14). This cost value influences the training process, leading to a decrease in false positive cases and a reduction in specificity.

By incorporating cost-sensitive learning for both "false negative" and "false positive" cases, the model can effectively update its weights to improve performance based on the specific misclassifications encountered during training.

4. Applications

We apply the cost-sensitive activation function approach to deep learning CAD models for lung cancer and breast cancer.

For the lung cancer application, we use the NIH/NCI LIDC[26] data and for the breast cancer application, we use the BreakHis.[27] For both applications, the data was split into training, validation and testing sets using stratified random sampling, with proportions of 70%, 10%, and 20%, respectively. To ensure more robust results, we repeated the process of data splitting and model development for 30 times. The classification performance on the testing set was reported with the mean value across all 30 trials and a 95% confidence interval.

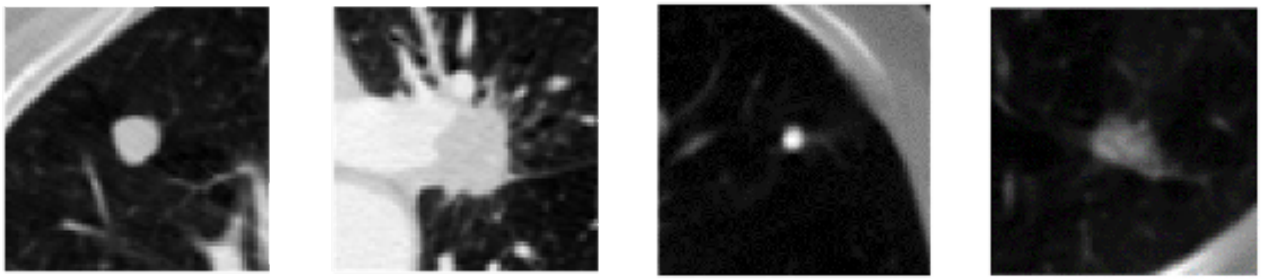

Fig. 2: Multiple visual appearances of cropped lung nodules: The two nodules on the left exhibit malignant features, characterized by spiculated contours and larger size, while the two on the right are benign, displaying smaller, smoother nodules indicative of non-malignancy.

4.1. *LIDC Dataset*

The LIDC[26] dataset contains 2,680 distinct nodules found in Computed Tomography (CT) scans from 1,010 patients. In this study, we implemented the following data preprocessing steps: First, we cropped nodules into images of size 71 x 71 (Figure 2), which is the size of the largest nodule in the dataset. Third, we assigned malignancy classification labels, where nodules with malignancy ratings of 1 (highly unlikely) and 2 (moderately unlikely) were labeled as 'Benign', while nodules with malignancy ratings of 4 (moderately suspicious) and 5 (highly suspicious) were labeled as 'Malignant'. After data pre-processing, which includes normalization and removal of indeterminate nodules with malignancy rating 3, we were left with 1,605 nodules. This dataset is imbalanced, comprising 699 malignant and 906 benign ones.

4.2. *BreakHis Dataset*

BreakHis[27] comprises of 7,909 histological images of breast tumor tissue collected from 82 patients using varying magnification levels (40X, 100X, 200X, and 400X). It contains imbalanced data with 2,480 benign and 5,429 malignant images. The images were 3-channel RGB, 8-bit depth each channel and dimension of 700 x 460 pixels. In this study, we normalized images using min-max normalization. A few sample images are shown in Figure 3.

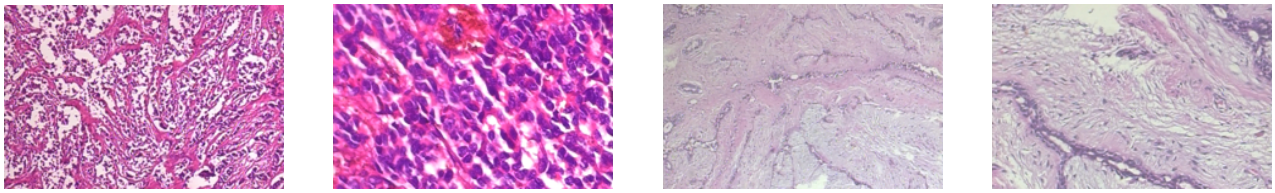

Fig. 3: Breast cancer histological images from the BreakHis Dataset. The left two images depict malignancy, while the right two show benign samples, illustrating critical distinctions for diagnosis.

4.3. *Design and Architecture of the Deep CNN Model*

The transfer learning method overcomes the limitation of having a small amount of training data by initially pre-training a deep learning model on a publicly available large dataset. As part of our study's cost-sensitive algorithm classification model, we fine-tune[28] a pre-trained ResNet18 convolutional neural network from ImageNet[29] on our own dataset.

For this study, we followed Nibali et al.'s recommendation[30] and utilized the ResNet18 CNN architecture.[31] In the intermediate layers of the architecture, the Rectified Linear Unit (ReLU) activation function is used. Given the objective of addressing a binary classification problem (malignant vs. benign), we implemented a single neuron with the sigmoid activation function as the output layer. Consequently, during training, if the output of the sigmoid activation function is greater than or equal to 0.5, we classify the instance as Malignant (positive class); otherwise, we classify it as Benign (negative class).

4.4. *Experimental Results*

The accuracy, sensitivity, and specificity assessment of the two datasets across 30 trials are tabulated in Table 1 through Table 4, showing the performance when the classifier is trained with different misclassification costs on the LIDC and BreakHis datasets. The first column and second column indicate the cost value given for false positives and false negatives, while the various metrics are listed from the third to the last columns. The numbers in bold indicate a significant difference compared to the results obtained without any cost values, whereas non-bold numbers indicate no significant difference.

From the experimental results, the cost-sensitive classifier indeed supports the analysis in Section 3.3. Table 1 reveals that we can enhance sensitivity while maintaining the same level of accuracy. Notably, in both datasets, sensitivity increases with a decrease in specificity for higher values of c(1,0), which is associated with false negatives, and lower values of c(0,1), which is associated with false positives. For the LIDC dataset, the highest sensitivity was achieved when c(0,1) = 0.33 and c(1,0) = 3, with accuracy and specificity remaining unaffected. We observed a significant improvement of 3.84% compared to the baseline model, where c(0,1) = c(1,0). Table 2 presents the classification performance of the BreakHis dataset, showing a 5.4% significant improvement when c(0,1) = 0.067 and c(1,0) = 15. Additionally, we observe a trend of significantly increasing sensitivity with decreasing specificity while keeping the overall accuracy within the range of 86.55% to 88.10%. In Tables 1 to 4, numbers within

Table 1: The classification performance on the LIDC testing data was assessed through 30 trials with increasing $c(1,0)$ cost values.

$c(0,1)$	$c(1,0)$	Accuracy	Sensitivity	Specificity
1	1	85.26% (84.63%, 85.89%)	76.22% (74.62%, 77.83%)	91.25% (90.33%, 92.17%)
0.5	2	85.25% (84.39%, 86.11%)	76.89% (75.17%, 78.61%)	90.79% (89.91%, 91.67%)
0.33	3	85.80% (85.19%, 86.41%)	**80.06% (78.81%, 81.30%)**	89.61% (88.54%, 90.69%)
0.2	5	84.93% (84.37%, 85.48%)	78.22% (76.76%, 79.69%)	**89.37% (88.39%, 90.35%)**
0.1	10	84.81% (84.06%, 85.55%)	**78.97% (77.33%, 80.62%)**	**88.67% (87.31%, 90.04%)**
0.067	15	84.56% (83.90%, 85.23%)	**78.58% (76.83%, 80.33%)**	**88.53% (87.23%, 89.83%)**

Table 2: The classification performance on the BreakHis testing data was assessed through 30 trials with increasing $c(1,0)$ cost values.

$c(0,1)$	$c(1,0)$	Accuracy	Sensitivity	Specificity
1	1	88.10% (88.03%, 88.16%)	90.38% (90.29%, 90.47%)	83.10% (82.98%, 83.21%)
0.5	2	**88.38% (88.32%, 88.45%)**	**91.84% (91.75%, 91.93%)**	**80.81% (80.63%, 80.99%)**
0.33	0.3	**88.39% (88.33%, 88.46%)**	**93.79% (93.69%, 93.88%)**	**76.58% (76.40%, 76.76%)**
0.2	5	**87.82% (87.74%, 87.91%)**	**95.05% (94.92%, 95.18%)**	**72.00% (71.76%, 72.23%)**
0.1	10	**87.15% (87.04%, 87.26%)**	**95.71% (95.58%, 95.84%)**	**68.42% (68.00%, 68.83%)**
0.067	15	**86.55% (86.51%, 86.79%)**	**95.78% (95.64%, 95.92%)**	**66.66% (66.09%, 67.23%)**

parentheses represent 95% confidence intervals, with bold numbers indicating significant distinctions compared to the baseline model (c(0,1) = c(1,0)).

An increase in sensitivity is observed with decreasing c(0,1) values, which shows significant improvement compared to the baseline models. Table 3 illustrates the results, focusing on increasing c(0,1) values, which will lead to a decrease in sensitivity significantly and an increase

Table 3: The classification performance on the LIDC testing data was assessed through 30 trials with increasing $c(0,1)$ cost values.

$c(0,1)$	$c(1,0)$	Accuracy	Sensitivity	Specificity
1	1	85.26% (84.63%, 85.89%)	76.22% (74.62%, 77.83%)	91.25% (90.33%, 92.17%)
2	0.5	84.83% (84.13%, 85.52%)	75.06% (73.24%, 76.88%)	91.31% (90.24%, 92.38%)
3	0.33	84.39% (83.70%, 85.07%)	**73.25% (71.44%, 75.06%)**	91.77% (90.66%, 92.88%)
5	0.2	83.00% (82.53%, 84.08%)	**69.56% (67.24%, 71.88%)**	92.41% (91.17%, 93.65%)
10	0.1	81.93% (81.03%, 82.82%)	**64.11% (61.89%, 66.33%)**	**93.74% (92.70%, 94.78%)**
15	0.067	80.48% (79.38%, 81.58%)	**60.69% (57.28%, 64.11%)**	**93.59% (92.36%, 94.83%)**

Table 4: The classification performance on the BreakHis testing data was assessed through 30 trials with increasing $c(0,1)$ cost values.

$c(1,0)$	$c(0,1)$	Accuracy	Sensitivity	Specificity
1	1	88.10% (88.03%, 88.16%)	90.38% (90.29%, 90.47%)	83.10% (82.98%, 83.21%)
2	0.5	**87.43% (87.35%, 87.50%)**	**88.75% (88.64%, 88.87%)**	**84.52% (84.33%, 84.71%)**
3	0.33	**84.70% (80.93%, 88.46%)**	**84.43% (78.48%, 90.39%)**	**85.27% (84.22%, 86.32%)**
5	0.2	**81.34% (76.26%, 86.41%)**	**79.02% (70.99%, 87.05%)**	**86.41% (85.02%, 87.80%)**
10	0.1	**70.62% (63.16%, 78.10%)**	**61.39% (49.71%, 73.08%)**	**90.85% (89.06%, 92.65%)**
15	0.067	**69.41% (62.17%, 76.66%)**	**59.33% (48.02%, 70.63%)**	**91.50% (89.76%, 93.20%)**

in specificity. We observe that with c(0,1) = 15 and c(1,0) = 0.067, the highest specificity is achieved, albeit with a trade-off in sensitivity. When we examine the impact of decreasing c(0,1) values as illustrated in Table 4, we observe a reverse relationship, with decreasing specificity and increasing sensitivity. This leads to a notable 8.4% improvement in specificity; however, it comes at the expense of a reduction in sensitivity.

For both the LIDC and BreakHis datasets, we observe an increasing trend of sensitivity with increasing c(1,0) values, and an increasing trend of specificity with increasing c(0,1) values. Notably, in both datasets, sensitivity increases with the decrease in specificity for higher c(1,0) values and lower c(0,1) values.

Our results indicate that by tuning the cost values, we can achieve higher sensitivity or specificity. Significantly, the overall accuracy remains consistent in the majority of cases.

5. Conclusion and Future Work

In this study, we proposed the incorporation of a cost-sensitive values into the activation function for deep learning-based CAD systems. Specifically, it addresses one of the most common problems in CAD, which is improving true positives measured using sensitivity, by adjusting the cost values without significantly impacting accuracy. The effectiveness and robustness of the model are demonstrated through theoretical analysis and experiments on different datasets. Compared with previous work on LIDC and BreakHis, this is the first study that utilizes cost values in activation functions to enhance sensitivity. Our findings strongly suggest that incorporating cost values as hyperparameters in future CAD systems holds promising benefits, with statistically significant increases of 3.84% and 5.4% in sensitivity, while maintaining overall accuracy, for LIDC and BreakHis Data.

While our study has yielded valuable insights, certain constraints, limited to the datasets used in this research, may impact the generalizability of our findings. Furthermore, our study predominantly relied on a single activation function and the use of binary cross entropy loss.

Future investigations can involve the inclusion of datasets from Medical Imaging and Data Resource Center[32] (MIDRC), The Cancer Imaging Archive[33] (TCIA), and other medical-related databases, which can provide a broader perspective on model performance. To improve model performance and address imbalanced datasets, we can explore various activation functions and loss functions, such as focal loss.[34] In addition to this, we will also explore different thresholds for classifying instances as malignant or benign based on the sigmoid activation function output. Currently, we use a default threshold of 0.5. We will also investigate the integration of the proposed cost values with the group Distributionally Robust Optimization[35] (gDRO) algorithm. This approach aims to enhance the worst group performance while preserving the overall CAD system's effectiveness. Additionally, we plan to delve into the impact of cost-sensitive activation functions on multi-class classification. These endeavors collectively aim to improve the accuracy and effectiveness of the cost-sensitive learning approach in medical diagnosis, ultimately benefiting diagnostic decision-making and patient outcomes.

References

1. H. Li, N. Zeng, P. Wu and K. Clawson, Cov-net: A computer-aided diagnosis method for recognizing covid-19 from chest x-ray images via machine vision, *Expert Systems with Applications* **207**, p. 118029 (2022).
2. P. Wu, H. Li, N. Zeng and F. Li, Fmd-yolo: An efficient face mask detection method for covid-19 prevention and control in public, *Image and Vision Computing* **117**, p. 104341 (2022).
3. N. Zeng, P. Wu, Z. Wang, H. Li, W. Liu and X. Liu, A small-sized object detection oriented

multi-scale feature fusion approach with application to defect detection, *IEEE Transactions on Instrumentation and Measurement* **71**, 1 (2022).
4. C. X. Ling and V. S. Sheng, Cost-sensitive learning and the class imbalance problem, *Encyclopedia of Machine Learning* **2011**, 231 (2008).
5. M. Phankokkruad, Cost-sensitive extreme gradient boosting for imbalanced classification of breast cancer diagnosis, in *2020 10th IEEE International Conference on Control System, Computing and Engineering (ICCSCE)*, 2020.
6. S. Ali, A. Majid, S. G. Javed and M. Sattar, Can-csc-gbe: Developing cost-sensitive classifier with gentleboost ensemble for breast cancer classification using protein amino acids and imbalanced data, *Computers in Biology and Medicine* **73**, 38 (2016).
7. M. Zieba, J. M. Tomczak, M. Lubicz and J. Świątek, Boosted svm for extracting rules from imbalanced data in application to prediction of the post-operative life expectancy in the lung cancer patients, *Applied Soft Computing* **14**, 99 (2014).
8. N. Thai-Nghe, Z. Gantner and L. Schmidt-Thieme, Cost-sensitive learning methods for imbalanced data, in *The 2010 International Joint Conference on Neural Networks (IJCNN)*, 2010.
9. S. Wang, W. Liu, J. Wu, L. Cao, Q. Meng and P. J. Kennedy, Training deep neural networks on imbalanced data sets, in *2016 International Joint Conference on Neural Networks (IJCNN)*, 2016.
10. I. D. Mienye and Y. Sun, Performance analysis of cost-sensitive learning methods with application to imbalanced medical data, *Informatics in Medicine Unlocked* **25**, p. 100690 (2021).
11. S. Zhu and J. Wan, Cost-sensitive learning for semi-supervised hit-and-run analysis, *Accident Analysis & Prevention* **158**, p. 106199 (2021).
12. T. M. Khoshgoftaar, E. Geleyn, L. Nguyen and L. Bullard, Cost-sensitive boosting in software quality modeling, in *7th IEEE International Symposium on High Assurance Systems Engineering, 2002. Proceedings*, 2002.
13. T. Le, M. T. Vo, B. Vo, M. Y. Lee and S. W. Baik, A hybrid approach using oversampling technique and cost-sensitive learning for bankruptcy prediction, *Complexity* **2019**.
14. D. Devi, S. K. Biswas and B. Purkayastha, A cost-sensitive weighted random forest technique for credit card fraud detection, in *2019 10th International Conference on Computing, Communication and Networking Technologies (ICCCNT)*, 2019.
15. J. Xiao, X. Zhou, Y. Zhong, L. Xie, X. Gu and D. Liu, Cost-sensitive semi-supervised selective ensemble model for customer credit scoring, *Knowledge-Based Systems* **189**, p. 105118 (2020).
16. K. Li, B. Wang, Y. Tian and Z. Qi, Fast and accurate road crack detection based on adaptive cost-sensitive loss function, *IEEE Transactions on Cybernetics* (2021).
17. E. Wang, Y. Jiang, Y. Li, J. Yang, M. Ren and Q. Zhang, Mfcsnet: Multi-scale deep features fusion and cost-sensitive loss function based segmentation network for remote sensing images, *Applied Sciences* **9**, p. 4043 (2019).
18. D. Li, X. Ma, Y. Ren and S.-W. Teng, Rectified softmax loss with all-sided cost sensitivity for age estimation, *IEEE Access* **8**, 32551 (2020).
19. I. D. Mienye, Y. Sun and Z. Wang, An improved ensemble learning approach for the prediction of heart disease risk, *Informatics in Medicine Unlocked* **20**, p. 100402 (2020).
20. H. Yaghoobi, E. Babaei, B. M. Hussen and A. Emami, Ebst: an evolutionary multi-objective optimization based tool for discovering potential biomarkers in ovarian cancer, *IEEE/ACM Transactions on Computational Biology and Bioinformatics* **18**, 2384 (2020).
21. I. D. Mienye, Y. Sun and Z. Wang, Improved sparse autoencoder based artificial neural network approach for prediction of heart disease, *Informatics in Medicine Unlocked* **18**, p. 100307 (2020).
22. S. I. Ali, H. S. M. Bilal, M. Hussain, J. Hussain, F. A. Satti, M. Hussain, G. H. Park, T. Chung and S. Lee, Ensemble feature ranking for cost-based non-overlapping groups: A case study of chronic kidney disease diagnosis in developing countries, *IEEE Access* **8**, 215623 (2020).

23. H. Wang, Z. Cui, Y. Chen, M. Avidan, A. B. Abdallah and A. Kronzer, Predicting hospital readmission via cost-sensitive deep learning, *IEEE/ACM Transactions on Computational Biology and Bioinformatics* **15**, 1968 (2018).
24. R. Parikh, A. Mathai, S. Parikh, G. C. Sekhar and R. Thomas, Understanding and using sensitivity, specificity and predictive values, *Indian Journal of Ophthalmology* **56**, p. 45 (2008).
25. D. M. Naeger, M. P. Kohi, E. M. Webb, A. Phelps, K. G. Ordovas and T. B. Newman, Correctly using sensitivity, specificity, and predictive values in clinical practice: how to avoid three common pitfalls., *AJR. American Journal of Roentgenology* **200**, W566 (2013).
26. S. G. Armato III, G. McLennan, L. Bidaut, M. F. McNitt-Gray, C. R. Meyer, A. P. Reeves, B. Zhao, D. R. Aberle, C. I. Henschke, E. A. Hoffman *et al.*, The lung image database consortium (lidc) and image database resource initiative (idri): a completed reference database of lung nodules on ct scans, *Medical Physics* **38**, 915 (2011).
27. F. A. Spanhol, L. S. Oliveira, C. Petitjean and L. Heutte, A dataset for breast cancer histopathological image classification, *IEEE Transactions on Biomedical Engineering* **63**, 1455 (2015).
28. W. Rawat and Z. Wang, Deep convolutional neural networks for image classification: A comprehensive review, *Neural Computation* **29**, 2352 (2017).
29. O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein *et al.*, Imagenet large scale visual recognition challenge, *International Journal of Computer Vision* **115**, 211 (2015).
30. A. Nibali, Z. He and D. Wollersheim, Pulmonary nodule classification with deep residual networks, *International Journal of Computer Assisted Radiology and Surgery* **12**, 1799 (2017).
31. K. He, X. Zhang, S. Ren and J. Sun, Deep residual learning for image recognition, in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016.
32. N. Baughan, H. M. Whitney, K. Drukker, B. Sahiner, T. Hu, M. McNitt-Gray, K. Myers, M. L. Giger *et al.*, Sequestration of imaging studies in midrc: a multi-institutional data commons, in *Medical Imaging 2022: Image Perception, Observer Performance, and Technology Assessment*, 2022.
33. K. Clark, B. Vendt, K. Smith, J. Freymann, J. Kirby, P. Koppel, S. Moore, S. Phillips, D. Maffitt, M. Pringle *et al.*, The cancer imaging archive (tcia): maintaining and operating a public information repository, *Journal of digital imaging* **26**, 1045 (2013).
34. M. Mulyanto, M. Faisal, S. W. Prakosa and J.-S. Leu, Effectiveness of focal loss for minority classification in network intrusion detection systems, *Symmetry* **13**, p. 4 (2020).
35. S. Sagawa, P. W. Koh, T. B. Hashimoto and P. Liang, Distributionally robust neural networks for group shifts: On the importance of regularization for worst-case generalization, *arXiv preprint arXiv:1911.08731* (2019).