A Comparative Study and Extension of a Deep Neural Network Model for Breast Cancer Diagnosis

Soraya Ezazipour
Dept. of Industrial Eng. & Management
Oklahoma State University
Stillwater, OK, USA
sezazip@okstate.edu

Somayeh Shojaeikhah
Dept. of Civil & Environmental Eng.
Oklahoma State University
Stillwater, OK, USA
somayeh.shojaeikhah@okstate.edu

Soroosh Farsiani
Dept. Mechanical & Aerospace Eng.
Oklahoma State University
Stillwater, OK, USA
soroosh.farsiani@okstate.edu

***Abstract*—Breast cancer is a leading cause of mortality among women worldwide. While early detection greatly improves survival rates, conventional screening approaches have limitations that technology may address. This study presents a comprehensive exploration and extension of a deep neural network model proposed for breast cancer diagnosis. The original framework, exists in [1], employs a feedforward architecture with diverse activation functions, specifically Swish, LeakyReLU, ReLU, and Sigmoid. Our work aimed to replicate these findings and introduce architectural variations to investigate performance improvements. The Wisconsin Diagnostic Breast Cancer dataset was utilized for model development and 10-fold Cross-Validation. In replicating the multi-activation model, our experiments slightly validated the effectiveness of complementary functions, achieving up to 97.72% testing accuracy. However, our proposed two-hidden-layer model with ReLU activations attained 98.60% accuracy, surpassing established benchmarks. This architecture demonstrated precision in identifying both malignant and benign cases per sensitivity and specificity metrics nearing 99%. The improved precision emphasizes these models' potential as robust clinical tools for early breast cancer diagnosis. Advancing computational methods remains integral in saving lives through enhanced screening.**

***Index Terms*—Deep neural networks, multiple activation functions, breast cancer diagnosis, Swish, Leaky ReLU.**

I. Introduction

Breast cancer represents an urgent public health issue, as the most common cancer in women worldwide. In 2020 alone, over 2 million new breast cancer cases were diagnosed, and nearly $700,000$ patients died from the disease [2]. While survival rates have improved thanks to early detection and better treatments, conventional breast cancer screening approaches have notable limitations that technology may help address. For instance, mammography, the standard screening modality, may overlook existing cancers in dense breast tissue, as well as yielding false positive results that lead to unneeded procedures [2]. These screening failures contribute to later-stage diagnoses as well as overdiagnosis, necessitating improved early diagnostic accuracy.

Advanced computational analytics, particularly deep learning, shows immense promise to enhance breast cancer detection from imaging data. Deep neural networks (DNNs) possess unrivaled capacity to learn from complex multidimensional datasets. By modeling intricate relationships between imaging features and cancer outcomes, DNNs can recognize distinct morphological patterns beyond human perception. Specialized architectures enable customized modeling of dependencies critical for medical applications. The layered composition facilitates hierarchical learning, while diverse activation functions introduce key nonlinearities. Carefully tailored deep learning systems thus have potential to address current pitfalls by boosting diagnostic precision.

This study investigates activation diversity within deep neural networks for enhanced breast cancer classification from imaging data. We developed and evaluated a layered neural network framework activating hidden layers with complementary Swish, Leaky ReLU, ReLU and Sigmoid functions. Architectural adaptations were introduced and assessed using the Wisconsin Diagnostic Breast Cancer (WDBC) benchmark dataset. Our experiments demonstrate state-of-the-art discrimination ability, with certain proposed variants achieving testing accuracy over 98%. The findings provide insights to guide future machine learning explorations towards maximizing early diagnostic performance. Advancing computational analytics remains pivotal in saving patient lives through improved screening.

II. RELATED WORKS

The application of artificial intelligence (AI) for breast cancer diagnosis has garnered substantial research attention. Sechopoulos et al. [3] provided an overview applying deep learning to mammography and tomosynthesis, addressing limitations of traditional techniques. Meanwhile, Sadoughi et al. [4] systematically reviewed AI-based approaches from $2007-2017$, finding support vector machines achieved over 95% accuracy across various modalities. Abd Elgader and Hamza [5] compared neural architectures on the WBCD dataset, observing multi-layer perceptrons (MLPs) outperformed alternatives. Utomo et al. [6] employed Artificial Neural Networks (ANN) with extreme learning techniques to diagnose breast cancer using the Breast Cancer Wisconsin Dataset. Their findings indicated that the Extreme Learning Machine Neural Networks (ELM ANN) outperformed the Backpropagation Neural Networks (BP ANN) in terms of a more effective generalization classifier model.

Beyond breast cancer, AI integration shows immense potential to enhance healthcare. Lakhani and Sundaram [7] proposed a deep convolutional neural network for tuberculosis detection via X-ray, achieving 98.81% accuracy surpassing experienced radiologists. Xu et al. [8] designed a DNN-based model for glaucoma diagnosis attaining sensitivity and specificity both exceeding 93% by extracting retinal nerve fiber layer features.

DNN advancements for breast cancer concentrate on enhancing discrimination capability. Akselrod-Ballin et al. [9] adopted a region-based convolutional neural network using biopsy-proven lesions, achieving AUC of 0.9 for malignancy detection. Multi-task learning has also shown promise, with Ribli et al. [10] developing an integrated classifier for lesion detection and diagnosis yielding 0.895 AUC.

Our work aligns with these efforts in tailoring DNNs to boost breast cancer diagnostic precision. Evaluating activation diversity and architectural adaptations provides insights to guide ongoing enhancements as AI integration offers immense life-saving potential. Advancing computational tools remains pivotal in improving early detection and survival.

III. TECHNICAL DESCRIPTION

The promising deep neural network framework presented in [1] served as the foundation for our experimental methodology. This feedforward architecture consists of an input layer, three hidden layers with diverse activation functions (AF), two intermittent dropout layers for regularization, and a sigmoid output layer for binary cancer/non-cancer classification.

The raw WDBC dataset contains 569 instances with 30 real-valued input features computed from digitized fine-needle aspirate images. Each feature corresponds to a morphometric attribute (e.g. radius, texture) indicative of cell nucleus properties. We standardized this dataset using Scikit-learn's StandardScaler to rescale features to zero mean and unit variance. This normalization prevents skewed ranges from biasing model fitting.

Our TensorFlow/Keras implementation allowed testing various activations. The multi-function DNN employed Swish (1st hidden layer), LeakyReLU (2nd hidden layer), ReLU (3rd hidden layer) following the original paper. We introduced a simplified variant with just two hidden layers and ReLU activations, found to enhance performance. Hidden nodes were determined via grid search hyperparameter tuning. Dropout layers used rate of 0.3 for regularization. All networks were trained for 50 epochs with early stopping, using batch size of 16.

Models were optimized using Adam, RMSprop, Adagrad and Nadam algorithms available in Keras, to assess impact on learning. A repeated stratified 10-fold cross-validation approach enabled robust performance estimation. We logged comprehensive metrics including accuracy, AUC, sensitivity, specificity, F1 score and MCC to quantify model discrimination ability from multiple perspectives. By considering such multifaceted indicators, we aimed for balanced cancer/non-cancer predictive power beyond raw accuracy.

A. Sigmoid AF

The sigmoid activation function is highly popular and widely used in neural networks due to its nonlinear nature. This function transforms input values into a range between 0 and 1. The main disadvantage of this AF is that the addition of an increased number of hidden layers with the Sigmoid AF in a DNN leads to a substantial reduction in the gradients of the loss function as they propagate backward through the network. This diminishing trend in gradients poses a significant challenge in training the DNN effectively. This is known as vanishing gradient problem [11]. The AF can be formulated as:

$$f(x) = \frac{1}{1+\exp(-x)} \tag{1}$$

B. ReLU AF

ReLU, short for Rectified Linear Unit, is a widely employed nonlinear activation function in neural networks. ReLU is considered more efficient compared to other activation functions because it activates only a specific number of neurons at a time, rather than all neurons simultaneously. However, in certain cases, the gradient value can be zero. This can result in the weights and biases not being updated during the backpropagation step in neural network training, which is the main disadvantages of this AF, and is called dead ReLU [12]. This activation function exhibits a faster learning pace when contrasted with logistic and hyperbolic tangent Afs [13], [14]. ReLU function can be defined as:

$$f(x) = \begin{cases} 0 & \text{for } x < 0, \\ x & \text{for } x \geq 0. \end{cases} \tag{2}$$

C. Leaky ReLU AF

Leaky ReLU represents an enhanced version of the ReLU function. In Leaky ReLU, for negative values of x, rather than setting the function's value to zero, it is defined as an extremely small linear component of x [12]. As a result, the issue of dead ReLU is addressed. This AF can be defined as:

$$f(x) = \begin{cases} ax & \text{for } x < 0, \\ x & \text{for } x \geq 0. \end{cases} \tag{3}$$

Where "a" represents a very small value introduced to create a slight slope for the negative values, preventing a complete zero output.

D. Swish AF

The Swish function, a relatively recent activation function discovered by researchers at Google, possesses a distinctive characteristic of being non-monotonic. Unlike monotonic functions, Swish may exhibit a decrease in its output value even when the input values are increasing [12]. It was demonstrated that the Swish activation function exhibits superior performance on more challenging datasets when compared to ReLU, especially in the context of deeper models [15]. It can be provided by the following formula:

$$f(x) = \frac{x}{1+\exp(-x)} = x.\,\mathit{sigmoid}(x) \tag{4}$$

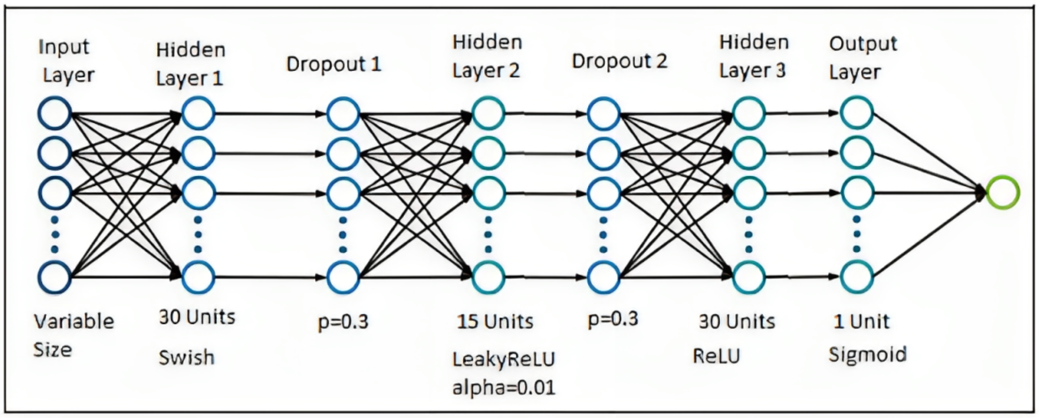

Fig. 1. Proposed model: multiple activation DNN. [1]

E. DNN Models

The DNN introduced by Vijayakumar et al. [1] is designed to explore the impact of different activation functions on model performance. The proposed architecture consists of an initial input layer, followed by three hidden layers and a final output layer. Between each pair of consecutive hidden layers, dropout layers are incorporated for regularization purposes, decreasing the risk of overfitting. The activation functions applied to the hidden layers include Swish for the first layer, LeakyReLU for the second layer, and ReLU for the third layer. The final output layer is activated using the Sigmoid AF.

As illustrated in Fig. 1, the architecture includes dropout layers after the first two hidden layers, contributing to the regularization of the model. Dropout, a well-established regularization technique, involves randomly ignoring certain units during training to minimize interdependent learning between them [16]. To facilitate a comprehensive analysis, four additional DNN models, each sharing the same architecture but employing distinct activation functions, were proposed for comparison. These models are named as follows:

- Sigmoid DNN: All hidden layer AFs are set to Sigmoid, and the output layer AF is Sigmoid.
- ReLU DNN: All hidden layer AFs are set to Rectified Linear Unit (ReLU), and the output layer AF is Sigmoid.
- LeakyReLU DNN: All hidden layer AFs are set to LeakyReLU with an alpha of 0.01, and the output layer AF is Sigmoid.
- Swish DNN: All hidden layer AFs are set to Swish, and the output layer AF is Sigmoid.

A detailed breakdown of the architecture and AFs for each DNN model is presented in Table I. It provides insights into the configuration of the networks for the purpose of comparative analysis. This comprehensive experimental setup aims to evaluate the performance of the DNN models under different AFs, representing the influence of activation functions on model behavior and predictive accuracy.

F. Project Extension

As an extension to the study, we introduced a variant of the proposed DNN architecture by altering the number of hidden layers. Specifically, we implemented a simpler two-hidden-layer architecture to investigate how this modification influences the overall performance of the model. This extension aims to provide insights into the impact of architectural variations on predictive accuracy and model behavior. Our process includes employing the same training parameters, such as utilizing optimizers like Adam, Adagrad, Nadam, and RMSprop. The training process spanned 50 epochs, with a batch size of 16, reflecting the same conditions as the original study. Furthermore, we adopt a repeated stratified 10-fold Cross-Validation approach to robustly evaluate the models' performance and account for potential biases.

IV. EXPERIMENTAL RESULTS

The experimental analysis conducted by Vijayakumar et al. [1] aimed to assess the performance of DNN models using the UCI dataset WDBC. Preprocessing involved standardizing the training dataset using Scikit-learn's StandardScaler to achieve zero mean and unit variance. Keras, a powerful Python library with a TensorFlow backend, facilitated the customization of DNN models, with additional support from Matplotlib, NumPy, and Scikit-learn for visualization, numerical operations, and machine learning functionalities.

Seven distinct optimizers, namely SGD, Adam, Adamax, Adagrad, Adadelta, RMSprop, and Nadam, available in Keras, were employed to train each DNN model for 50 epochs. This comprehensive approach resulted in 35 models, enabling a thorough exploration of the impact of different optimization algorithms. The performance evaluation adopted a repeated stratified 10-fold Cross-Validation method to ensure robustness and mitigate bias. Key performance indices, including accuracy, sensitivity, specificity, precision, recall, Matthews Correlation Coefficient (MCC), and F1 score, were computed for a comprehensive assessment.

The study not only prioritized high accuracy but also aimed for a balanced trade-off between sensitivity and specificity,

TABLE I
DETAILED CONFIGURATION OF DNN MODELS [1].

Layer	Units#	Sigmoid DNN	ReLU DNN	Leaky ReLU DNN	Swish DNN	Proposed DNN
Input	Variable size	-	-	-	-	-
Dense	30	Sigmoid	ReLU Alpha=0.01	LeakyReLU	Swish	Swish
Dropout $\rho = 0.3$	-	-	-	-	-	-
Dense	15	Sigmoid	ReLU Alpha=0.01	LeakyReLU	Swish	LeakyReLU
Dropout $\rho = 0.3$	-	-	-	-	-	-
Dense	30	Sigmoid	ReLU Alpha=0.01	LeakyReLU	Swish	ReLU
Dense (Output)	1	Sigmoid	Sigmoid	Sigmoid	Sigmoid	Sigmoid

along with elevated MCC and F1 score values. The performance metrics were defined mathematically, in equations (5)−(11), incorporating true positive (TP), true negative (TN), false positive (FP), and false negative (FN) values. The utilization of these diverse metrics provided a holistic understanding of the models' performance across various dimensions.

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \times 100 \tag{5}$$

$$Sensitivity = \frac{TP}{TP + FN} \times 100 \tag{6}$$

$$Specifity = \frac{TN}{TN + FP} \times 100 \tag{7}$$

$$Precision = \frac{TP}{TP + FP} \tag{8}$$

$$Recall = \frac{TP}{TP + FN} \tag{9}$$

$$F1\ Score = 2 \times \frac{Precision \times Recall}{Precision + Recall} \tag{10}$$

$$\text{MCC} = \frac{TPTN - FPFN}{\sqrt{(TP + FP)(TP + FN)(TN + FP)(TN + FN)}} \tag{11}$$

The results of DNNs proposed by Vijayakumar et al. [1] is presented in Table II, highlighting the accuracy and performance of five DNN models with different optimization algorithms. Notably, their findings revealed variations in performance across different activation functions and optimizers. Sigmoid DNN performs the worst overall, with maximum accuracy of only 80.28% with Adagrad optimizer. The vanishing gradient issue with Sigmoid likely prevents effective learning. ReLU, LeakyReLU, and Swish DDN models exhibit comparable high performance - exceeding even 97% accuracy using Adam, Nadam, RMSProp optimizers. The proposed model with diverse activations outperforms the individual activation DNNs. The complementary modeling capabilities appear to boost overall accuracy.

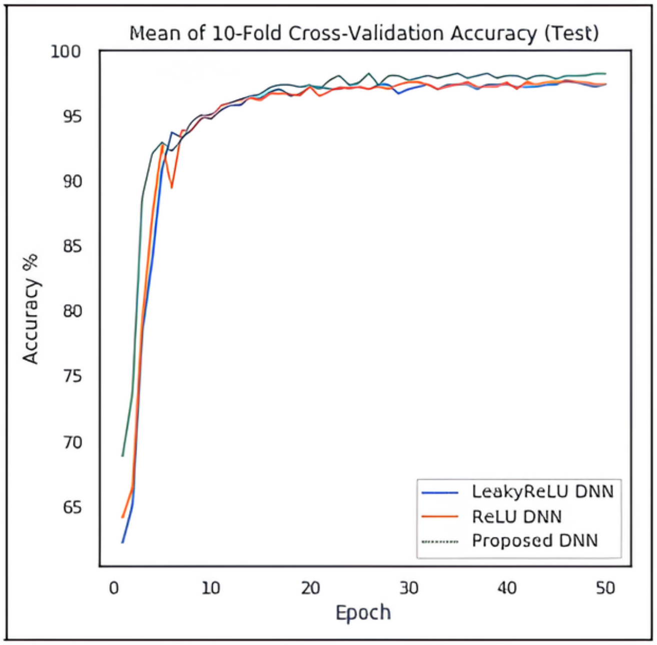

Fig. 2. The average testing accuracy at each epoch in the 10−fold CV experiments for top three DNN models investigated by Vijayakumar et al. [1]

The mean testing accuracy at each epoch during the 10-fold Cross-Validation experiments for their three DNN models with the highest accuracy is illustrated in Fig. 2.

In our attempt to replicate and extend the study, we followed all steps to to replicate the models, and the comparison of results for five DNNs with different optimizers, along with all introduced performance indices. Our obtained results are represent in Table III. Our experiments validate slightly the effectiveness of combining complementary activation functions, with the multi-activation DNN achieving testing accuracy up to 97.72%. The Swish, LeakyReLU, and ReLU hidden layers appear to model distinct data relationships that prove synergistic. However, choice of optimization algorithm impacts model convergence. For example, RMSprop yielded much higher validation accuracy (97%) compared to Adadelta (54%) for this DNN.

Examining the confusion matrices provides insight into the predictive patterns. The high specificity of 99% for the model

TABLE II
Results in [1] for comparing the five DNN models with different optimization algorithms.

DNN	Optimizer	Accuracy	F1-score	MCC	Sensitivity	Specificity	Precision	Recall
Sigmoid	Adadelta	62.74	0	0	0	100	0	0
	Adagrad	80.28	0.645	0.594	48.11	99.38	0.979	0.481
	Adam	62.92	0.009	0.054	0.47	100	1	0.005
	Adamax	62.74	0	0	0	100	0	0
	Nadam	71.55	0.389	0.398	24.29	99.61	0.974	0.243
	RMSprop	62.81	0.004	0.034	0.19	100	1	0.002
	SGD	61	0.12	0.003	7.17	92.97	0.377	0.072
ReLU	Adadelta	97.77	0.97	0.952	95.99	98.82	0.98	0.96
	Adagrad	97.94	0.972	0.956	96.56	98.77	0.979	0.966
	Adam	94.76	0.929	0.888	91.37	96.78	0.944	0.914
	Adamax	95.1	0.933	0.895	91.84	97.03	0.948	0.918
	Nadam	97.75	0.97	0.952	96.51	98.49	0.974	0.965
	RMSprop	96.77	0.956	0.931	95.05	97.79	0.962	0.95
	SGD	62.74	0	0	0	100	0	0
Leaky ReLU	Adadelta	97.77	0.97	0.952	95.75	98.96	0.982	0.958
	Adagrad	97.93	0.972	0.956	96.37	98.85	0.98	0.964
	Adam	94.94	0.931	0.891	91.56	96.95	0.947	0.916
	Adamax	94.87	0.93	0.89	91.6	96.81	0.945	0.916
	Nadam	97.42	0.965	0.945	95.9	98.32	0.971	0.959
	RMSprop	96.75	0.956	0.93	95.09	97.73	0.961	0.951
	SGD	62.74	0	0	0	100	0	0
Swish	Adadelta	95.24	0.934	0.898	90.05	98.32	0.97	0.9
	Adagrad	97.49	0.966	0.946	95.42	98.71	0.978	0.954
	Adam	92.62	0.898	0.841	87.64	95.57	0.922	0.876
	Adamax	92.28	0.893	0.834	86.6	95.66	0.922	0.866
	Nadam	97.52	0.966	0.947	95.61	98.66	0.977	0.956
	RMSprop	94.31	0.922	0.878	90.38	96.64	0.941	0.904
	SGD	62.74	0	0	0	100	0	0
Proposed	Adadelta	97.86	0.971	0.954	95.94	98.99	0.983	0.959
	Adagrad	98.21	0.976	0.962	96.65	99.13	0.985	0.967
	Adam	94.15	0.922	0.875	93.21	94.71	0.913	0.932
	Adamax	94.04	0.92	0.873	92.36	95.04	0.917	0.924
	Nadam	97.33	0.964	0.943	96.13	98.04	0.967	0.961
	RMSprop	95.98	0.946	0.914	95.14	96.47	0.941	0.951
	SGD	62.74	0	0	0	100	0	0

indicates very few false positives, reflecting excellent identification of true negative cases. Sensitivity remains reasonably balanced at 96%, showing accurate malignant case detection.

Upon comparing Table II and Table III, distinct disparities in reported results become evident in certain instances. Take, for instance, the Sigmoid DNN with the Adam optimizer. According to the reference [1], the reported accuracy is 62.92%. In contrast, our experimental findings indicate a significantly higher accuracy of 97.54%. These discrepancies emphasizes the critical need for careful examination and consideration of various factors in the process of designing and training neural network models.

In addition, we designed a DNN architecture inspired by their work and introduced modifications to enhance discriminative capability. The proposed model, illustrated in Fig. 3. It is a simple two-hidden-layer architecture with 128 and 64 units. Utilizes ReLU AF in hidden layers. Applies dropout with a rate of 0.3 after each hidden layer to prevent overfitting. Uses a binary sigmoid AF in the output layer. We utilized the four different optimizer, Adam, Adagrad, Nadam and RMSprop and trained the DNN for 50 epochs with a batch size of 16. Implements early stopping to prevent overfitting. The corresponding results are presented in Table III.

Across all the activation functions and optimizers tested, our proposed DNN model outperforms the other DNN models on nearly all evaluation metrics. This indicates our model is more effective. For our model, the Nadam optimizer achieves the highest testing accuracy (98.60%), F1 score (0.98), MCC (0.97), sensitivity (0.98), specificity (0.99), precision (0.99), and recall (0.98). Thus Nadam optimizes our model the best. Adam, Adagrad, and RMSprop also optimize our model well, with testing accuracy and F1 scores above 95% and MCC near 0.95% or higher. SGD optimizes the baseline proposed model best but is not as effective at optimizing our model. Our model achieves particularly high specificity scores ($0.98 - 0.99$), indicating it rarely misclassifies negative cases. Sensitivity is also strong at $0.96 - 0.98$ showing good detection of positive cases. Additionally, the mean testing accuracy at each epoch during the 10-fold Cross-Validation experiments for replicated DNN models and our proposed model are illustrated in Fig. 4 and Fig. 5, respectively.

V. Conclusion

This project explored and extended a deep feedforward neural network model for breast cancer diagnosis. By replicating and extending the original study, we investigated the impact of multiple activation functions and evaluated the model's performance using diverse optimizers. The replicated models

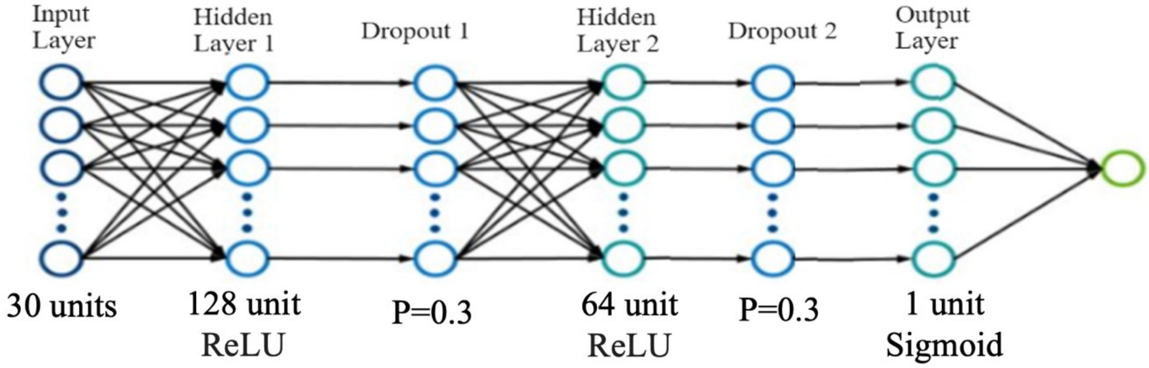

Fig. 3. The proposed DNN model in this project

TABLE III
Our results for comparing THE FIVE REPLICATED DNN MODELS AND OUR PROPOSED DNN MODEL

DNN	Optimizer	Accuracy	F1-Score	MCC	Sensitivity	Specificity	Precision	Recall
Sigmoid	Adadelta	54.98	0.16	0.00	0.30	0.70	0.81	0.30
	Adagrad	62.74	0.00	0.00	0.00	1.00	1.00	0.00
	Adam	97.54	0.97	0.95	0.96	0.98	0.97	0.96
	Adamax	97.72	0.97	0.95	0.96	0.99	0.98	0.96
	Nadam	97.37	0.96	0.95	0.95	0.99	0.98	0.95
	RMSprop	97.72	0.97	0.95	0.96	0.99	0.98	0.96
	SGD	62.74	0.00	0.00	0.00	1.00	1.00	0.00
Relu	Adadelta	52.93	0.31	-0.01	0.36	0.63	0.49	0.36
	Adagrad	91.02	0.86	0.81	0.80	0.97	0.96	0.80
	Adam	97.37	0.96	0.94	0.96	0.98	0.97	0.96
	Adamax	97.19	0.96	0.94	0.96	0.98	0.97	0.96
	Nadam	97.72	0.97	0.95	0.96	0.99	0.98	0.96
	RMSprop	97.72	0.97	0.95	0.96	0.99	0.98	0.96
	SGD	96.84	0.96	0.93	0.95	0.98	0.97	0.95
Leakyrelu	Adadelta	60.76	0.34	0.14	0.38	0.74	0.59	0.38
	Adagrad	94.73	0.92	0.89	0.87	0.99	0.99	0.87
	Adam	97.19	0.96	0.94	0.96	0.98	0.97	0.96
	Adamax	97.72	0.97	0.95	0.96	0.99	0.98	0.96
	Nadam	97.54	0.97	0.95	0.96	0.99	0.98	0.96
	RMSprop	96.84	0.96	0.93	0.96	0.97	0.96	0.96
	SGD	97.02	0.96	0.94	0.95	0.98	0.97	0.95
Swish	Adadelta	54.34	0.45	0.08	0.51	0.56	0.46	0.51
	Adagrad	93.51	0.91	0.86	0.87	0.97	0.95	0.87
	Adam	97.02	0.96	0.94	0.95	0.98	0.97	0.95
	Adamax	97.72	0.97	0.95	0.96	0.99	0.98	0.96
	Nadam	97.02	0.96	0.94	0.95	0.98	0.97	0.95
	RMSprop	97.19	0.96	0.94	0.95	0.98	0.97	0.95
	SGD	96.31	0.95	0.92	0.93	0.98	0.97	0.93
Proposed model [1]	Adadelta	54.67	0.57	0.18	0.79	0.40	0.49	0.79
	Adagrad	94.73	0.92	0.89	0.89	0.98	0.97	0.89
	Adam	97.37	0.96	0.94	0.96	0.98	0.97	0.96
	Adamax	97.72	0.97	0.95	0.96	0.99	0.98	0.96
	Nadam	97.37	0.96	0.94	0.96	0.98	0.97	0.96
	RMSprop	97.37	0.96	0.94	0.95	0.99	0.98	0.95
	SGD	97.54	0.97	0.95	0.96	0.99	0.98	0.96
Our proposed model	Adam	97.37	0.96	0.95	0.96	0.98	0.97	0.96
	Adagrad	95.44	0.93	0.91	0.91	0.98	0.97	0.91
	Nadam	98.60	0.98	0.97	0.98	0.99	0.99	0.98
	RMSprop	97.89	0.97	0.96	0.97	0.99	0.98	0.97

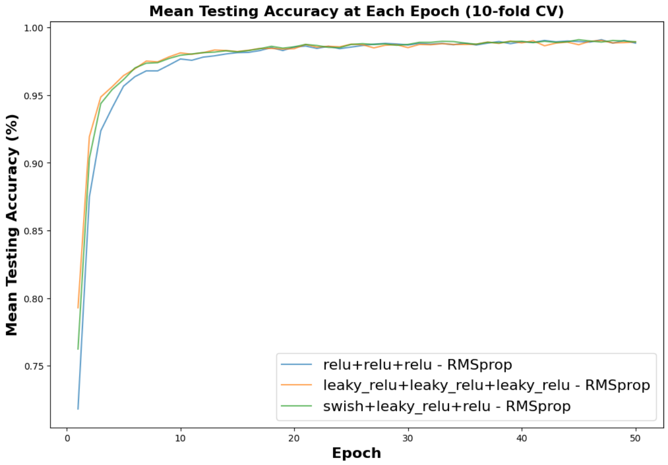

Fig. 4. The average testing accuracy at each epoch in the 10−fold CV experiments for our new DNN model with RMSprop optimizer.

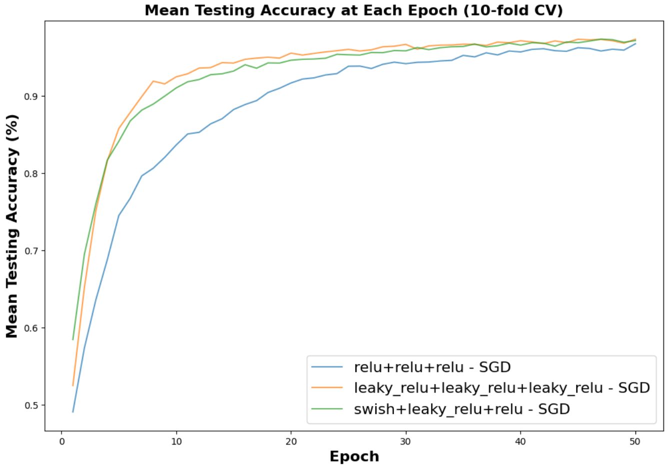

Fig. 5. The average testing accuracy at each epoch in the 10−fold CV experiments for our new DNN model with SGD optimizer.

provide a subtle validation of the efficacy gained by blending complementary activation functions in hidden layers. Notably, the proposed model from the original paper demonstrates this synergy by attaining an impressive testing accuracy of 97.72% with the Adamax optimizer.

In addition, we proposed a DNN model with different hidden layer numbers. Our proposed two-hidden-layer model with ReLU activations surpassed the established models, attaining 98.60% testing accuracy with the Nadam optimizer. This highlights the potential of architectural adaptations to further improve discriminative capability. Performance metrics revealed our model has excellent sensitivity (0.98) and specificity (0.99), precisely identifying both malignant and benign cases. The high F1 score (0.98) and MCC (0.97) also demonstrate strong predictive power. While most optimizers proved effective, Nadam optimized our model the best, indicating the importance of testing different optimization algorithms.

The enhanced performance emphasizes the potential of carefully designed and optimized DNNs as precise clinical classification tools. Our findings provide insights to guide future explorations focused on improving early diagnostic accuracy. Advancing such computational methods remains pivotal in the fight against breast cancer.

References

[1] Vijayakumar, K., Kadam, V. J., & Sharma, S. K. (2021). Breast cancer diagnosis using multiple activation deep neural network. Concurrent Engineering, 29(3), 275–284.

[2] Houssami, N., Lee, C. I., Buist, D. S., & Tao, D. (2017). Artificial intelligence for breast cancer screening: opportunity or hype?. The Breast, 36, 31–33.

[3] Sechopoulos, I., Teuwen, J., & Mann, R. (2021, July). Artificial intelligence for breast cancer detection in mammography and digital breast tomosynthesis: State of the art. In Seminars in cancer biology (Vol. 72, pp. 214–225). Academic Press.

[4] Sadoughi, F., Kazemy, Z., Hamedan, F., Owji, L., Rahmanikatigari, M., & Azadboni, T. T. (2018). Artificial intelligence methods for the diagnosis of breast cancer by image processing: a review. Breast Cancer: Targets and Therapy, 219–230.

[5] Abd Elgader, H. A., & Hamza, M. H. (2011). Breast cancer diagnosis using artificial intelligence neural networks. J. Sc. Tech, 12(1), 159–171.

[6] Utomo, C. P., Kardiana, A., & Yuliwulandari, R. (2014). Breast cancer diagnosis using artificial neural networks with extreme learning techniques. International Journal of Advanced Research in Artificial Intelligence, 3(7), 10–14.

[7] Lakhani P, and Sundaram B (2017) Deep learning at chest radiography: automated classification of Pulmonary Tuberculosis by using convolutional neural networks. Radiology 284(2):574-–582.

[8] Xu, Yongli, et al. ”A hierarchical deep learning approach with transparency and interpretability based on small samples for glaucoma diagnosis.” NPJ digital medicine 4.1 (2021): 48.

[9] Akselrod-Ballin, A., Karlinsky, L., Alpert, S., Hasoul, S., Ben-Ari, R., & Barkan, E. (2016). A region based convolutional network for tumor detection and classification in breast mammography. In Deep Learning and Data Labeling for Medical Applications: First International Workshop, LABELS 2016, and Second International Workshop, DLMIA 2016, Held in Conjunction with MICCAI 2016, Athens, Greece, October 21, 2016, Proceedings 2 (pp. 197–205). Springer International Publishing.

[10] Ribli, D., Horváth, A., Unger, Z., Pollner, P., & Csabai, I. (2018). Detecting and classifying lesions in mammograms with deep learning. Scientific reports, 8(1), 4165.

[11] Hochreiter, S., Bengio, Y., Frasconi, P., & Schmidhuber, J. (2001). Gradient flow in recurrent nets: the difficulty of learning long-term dependencies.

[12] Sharma, S., Sharma, S., & Athaiya, A. (2017). Activation functions in neural networks. Towards Data Sci, 6(12), 310–316.

[13] Glorot, X., Bordes, A., & Bengio, Y. (2011, June). Deep sparse rectifier neural networks. In Proceedings of the fourteenth international conference on artificial intelligence and statistics(pp. 315–323). JMLR Workshop and Conference Proceedings.

[14] LeCun, Y., Bengio, Y., & Hinton, G. (2015). Deep learning. nature, 521(7553), 436–444.

[15] Ramachandran, P., Zoph, B., & Le, Q. V. (2017). Searching for activation functions. arXiv preprint arXiv:1710.05941.

[16] Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., & Salakhutdinov, R. (2014). Dropout: a simple way to prevent neural networks from overfitting. The journal of machine learning research, 15(1), 1929–1958.

The link of the code in Google Colab: https://colab.research.google.com/drive/1QMWcqfHpaUWZ6CR2-aKRQOzfmHo5mv9K?usp=sharing