The First International Conference on
Machine Learning and Knowledge Discovery (MLKD 2024)
Amirkabir University of Technology, December 18-19, 2024

BCGNN:Machine Learning based Breast Cancer Classification improvement Using Graph Neural Network

Alireza Kazempoor Choobari* Sadegh Sulaimany†

Abstract

Breast cancer has seen a significant rise in incidence and mortality rates in recent years. Machine learning techniques have been extensively employed to diagnose and classify this disease. Ensemble learning methods, in particular, have gained prominence for binary breast cancer classification. In this study, we leveraged the Wisconsin Breast Cancer (WDBC) Dataset and introduced a novel approach using graph neural networks for breast cancer classification.In this work, we are transforming numerical data from a dataset into a graph and applying Graph Neural Networks (GNNs) to it. We are also considering the embedded graph features. Our results demonstrate a remarkable average accuracy of 98.83%, along with precision, recall, and F1-score values of 99.07%. These findings strongly suggest that graph neural networks outperform previous methods in terms of accuracy and overall performance. Our primary objective was to develop a more accurate classification model for breast cancer, and the results obtained through graph neural networks have successfully achieved this goal.

Keywords: Breast Cancer Classification, Machine Learning, Graph Neural Network, Graph Convolutional Network.

1 Introduction

Breast cancer is the most perilous type of cancer for women worldwide, making prevention a global priority. The implementation of novel and innovative techniques is crucial to mitigating the risk of developing breast cancer[1]. While lifestyle modifications such as weight management, regular physical activity, and alcohol avoidance can significantly reduce the risk of developing breast cancer, innovative diagnostic techniques are crucial for early detection and treatment. By implementing these advanced methods, we can effectively combat the rising mortality rates associated with breast cancer[2].

For breast cancer diagnosis and classification, various machine learning techniques can be employed. In the case of image-based datasets, deep learning techniques such as convolutional neural networks (CNNs) are commonly used to perform breast cancer detection on mammograms[3]. Additionally, in recent years, graph neural networks (GNNs) have been applied to image datasets of breast cancer with the aim of diagnosing this type of cancer[4]. For structured, tabular data, ensemble learning methods are prevalent, and a variety of algorithms have been utilized over the years[5].

While previous research has extensively explored image-based datasets, demographic data has often been overlooked. In our work, we focus on demographic data and disregard image-based datasets. By concentrating on demographic data and striving to improve accuracy, we anticipate achieving significant results. Figure 1 illustrates the two types of data commonly encountered in breast cancer research.

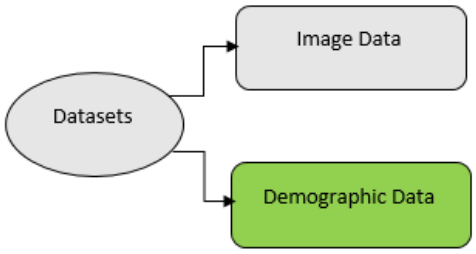

Figure 1: Breast Cancer main dataset types.

Our research will explore the effectiveness of graph neural networks in classifying breast cancer using the Wisconsin Breast Cancer Dataset(WDBC). We expect this novel approach to surpass the accuracy of existing methods. To enhance the accuracy of our binary classification model, We use the BCGNN (Breast Cancer Classification using Graph Neural Network) model. In the architecture of this model, We employ a Graph Convolutional Network (GCN), a type of graph

*Social and Biological Network Analysis Laboratory(SBNA), Department of Computer Engineering, University of Kurdistan, Sanandaj, Iran, alireza.choobari@uok.ac.ir
†Social and Biological Network Analysis Laboratory(SBNA), Department of Computer Engineering, University of Kurdistan, Sanandaj, Iran, s.sulaimany@uok.ac.ir

neural network, and believe that the unique properties of GCNs can enhance accuracy compared to previous work on the WDBC dataset. We specify the following sctions of the paper as follows:

In section 2, we delve into previous studies and research related to our work, focusing on the methodologies employed for breast cancer detection and classification.section 3 offers a comprehensive overview of the dataset utilized in our research, along with a detailed explanation of the proposed methodology. We will delve into the dataset's characteristics, including its features and samples. Additionally, we will provide a thorough description of the employed architecture, its various components, and an outline of the implementation process.In section 4, we will present and analyze the results obtained from our experiments. A comparative analysis will be conducted to evaluate the performance of our proposed method against existing approaches. Finally, in section 5 summarizes the key findings of our research. We will discuss the overall contributions of our work, highlight the achieved outcomes, and propose potential avenues for future research.

2 Related Work

Different machine learning models are frequently employed for detection and classification purposes. The primary concern, however, lies in the model's ability to deliver satisfactory accuracy for the classification task at hand. Furthermore, mitigating data leakage is paramount. Our objective is to surpass the accuracy of previous studies and demonstrate that novel models can genuinely outperform traditional methods[6].

In breast cancer classification, these points are also of great importance and have always been considered. In[7], the authors used and evaluated five different machine learning models. Decision trees(DT), random forests(RF), support vector machines(SVM), neural networks(NN), and logistic regression(LR) were the machine learning models utilized in this study. These models were applied to both the WDBC and BCCD[8] datasets and are widely used for breast cancer classification.

Also in [9], The authors employed a stacking classification method on the Wisconsin Breast Cancer Dataset and achieved a commendable accuracy of 97.20% in their classification task. In addition to [10], A higher classification accuracy was obtained by applying an Hierarchical Clustering Random Forest (HCRF) model to the WBCD[11] and WDBC datasets. Feature selection was carried out using Variable Importance Measure (VIM), demonstrating improved performance over AdaBoost, Decision Trees, and Random Forests.

In [12], The authors achieved a high accuracy of 94.35% by K-Nearest Neighbor model to the WDBC dataset. Moreover [13], By comparing decision trees and AdaBoost, it was found that AdaBoost yielded the best results, achieving an accuracy score of 92.53%. Then [14], Among the seven classifiers employed for breast cancer detection, the XGBoost (gradient boosting algorithm) outperformed the others, reaching an impressive accuracy of 96%.
Finally, in [15], The authors employed an ensemble learning method, dubbed ELRL, comprising four base models: Extra Trees, LightGBM, Ridge Classifier, and LDA. These models independently perform classification tasks and their outputs are fed into a voting classifier to determine the final prediction. When applied to the WDBC dataset, this ensemble achieved a remarkable accuracy of 97.66%. Inspired by the high accuracies obtained using machine learning and ensemble methods on the WDBC dataset, we delve into the use of cutting-edge graph-based deep learning models for breast cancer classification. A summary of the related studies is presented in Table 1.

3 Materials and Methods

In subsection 3.1, we will describe the dataset and its details.In subsection 3.2, we provide an explanation of GNN. In subsection 3.3, we describe the overall implementation of our work. Finally, In subsection 3.4, we discuss the specific type of GNN used, which is BCGNN, and explain how it was applied in our implementation.

3.1 Dataset

The dataset used in this study is the Wisconsin Breast Cancer (WDBC) Dataset, a widely used benchmark in machine learning for cancer diagnosis.In [16],a link to access the provided dataset file has been included. It comprises 569 instances, each representing a cell nucleus from a fine needle aspirate (FNA) of a breast mass. These instances are characterized by 32 features, derived from digitized images of the cell nuclei.

The dataset includes two categorical features:

- ID: A unique identifier for each instance.
- Diagnosis: The target variable, indicating whether the tumor is malignant (M) or benign (B).

Additionally, there are 10 categorical features with three possible values each, and 30 continuous features. These 30 continuous features represent measurements

Table 1: Comparison of Machine Learning based works on the WDBC Dataset.

Reference	Year	Model	Dataset	Samples	Features	Accuracy
[12]	2021	KNN	WDBC	569	32	94.35%
[13]	2021	Adaboost	WDBC	569	31	92.53%
[14]	2024	XGBoost	WDBC	569	30	96%
[15]	2024	ELRL-E	WDBC	569	9	97.66%

of cell nuclei characteristics such as radius, texture, perimeter, and symmetry. Each continuous feature is calculated as the mean, standard error (SE), or worst value of the corresponding measurement. For instance, the 'radius-mean' feature represents the average radius of the cell nuclei in a given sample.

In [15], only the mean features were utilized, excluding the 'compactness' feature. Thus, the study in [15] employed nine features, all representing mean values. In contrast, our study leverages all 30 continuous features, along with the target variable, for classification. Table 2 provides a list of the dataset's features along with their brief descriptions.

3.2 GNN

Typically, when we use a neural network, our goal is to extract significant features from the data. We seek a vector representation for the input vector that encapsulates all the essential features. But, in a graph neural network (GNN), we aim to find a representation for the input graph that captures both the structural and content information of the graph. A GNN is a type of artificial neural network specifically designed to process graph-structured data. In a GNN, our input is a graph constructed from a dataset with its samples and corresponding features, and our output is a representation of this input graph. This representation can be subsequently employed for classification tasks[17].

Using GNNs offers both advantages and disadvantages. On the plus side, GNNs typically reduce time complexity compared to traditional models, have lower computational costs, and exhibit better scalability and efficiency when dealing with large graphs. However, a potential drawback is that all nodes within the graph are updated at each iteration, including those that are not direct neighbors. This can lead to increased memory consumption[18].

Our objective is to leverage GNNs to generate a novel representation of the input graph and subsequently apply this representation to perform binary classification of breast cancer on the WDBC dataset. GNNs employ two primary functions: aggregator and combinator. The aggregator function aggregates information from the neighbors of each node, updating the node's features. The combinator function merges the aggregated information and representation of a node's neighbors with the node's own information and representation obtained from the preceding layer. Various operations can be used for this combination, such as mean, sum, maximum, or minimum. Finally, we can apply an activation function to determine the class labels and perform the classification task.

3.3 Research Workflow

Our method can be viewed as an 8-step process, which will be detailed further below:

Graph Neural Networks (GNNs) for Graph-Structured Data:

GNNs are specifically designed to process data represented as graphs, where nodes represent entities and edges represent relationships between them. Among the popular GNN variants, GraphSAGE, GCN, and GAT each have their distinct characteristics. This study will focus on GCNs.

Data Preparation and Preprocessing:

Before constructing the BCGNN, the dataset will be loaded, cleaned to remove outliers and standardize features, and visualized for better understanding. The data will then be split into training and testing sets, with 70% allocated for training and 30% for testing. Inductive learning has been used for data splitting. In this setting, the model is trained on an independent training dataset and then evaluated on a separate test dataset.

Graph Construction:

A graph will be created from the dataset, where each data sample becomes a node. The feature vector associated with each sample will represent its node attributes. Edges will connect nodes based on their Euclidean distance. If two nodes are within a specified threshold (e.g., 0.5), they will be connected. Nodes with no connections will be equipped with self-loops.

Table 2: Dataset features and brief descriptions for each feature. Each feature has three states: standard error(se), mean, and worst.

Feature	Description	Feature	Description
1.Radius	Radius of tumor cells. Measurement of the distance from the center of the cell to its edge.	**6.Compactness**	Calculation of cell Compactness.
2.Texture	Description of the tumor's texture, surface, and structural features.	**7.Concavity**	Assessment of cell membrane invaginations.
3.Perimeter	Measuring the circumference of tumor cells.	**8.Concave-points**	Counting the number of invaginations at cell edges.
4.Area	Measuring the surface area of tumor cells.	**9.Symmetry**	Assessment of cell shape regularity.
5.Smoothness	Assessment of the regularity and smoothness of cell boundaries.	**10.Fractal-dimension**	Assessment of tumor cell morphological complexity.

$$d(x, y) = \sqrt{\sum_{i=1}^{n}(x_i - y_i)^2} \quad (1)$$

BCGNN Model and Classification:

This constructed graph will serve as the input to the BCGNN model. The BCGNN will process the graph and output a classification, determining whether the sample is benign or malignant. Subsection 3.4 will delve into the BCGNN's architecture and its operational principles.

Evaluation and Comparison:

To assess the model's performance, various evaluation metrics will be employed, and the results will be compared to those reported in existing studies. Figure 2 provides a comprehensive overview of the entire process.

3.4 BCGNN architecture

Graph Convolutional Networks (GCNs) are a specific type of Graph Neural Network (GNN) designed to process graph-structured data. Inspired by convolutional neural networks (CNNs), GCNs offer a significant advantage over CNNs by enabling the processing of non-Euclidean data represented as graphs, unlike CNNs which are limited to Euclidean data[19].

In GCNs, aggregator and combinator functions play pivotal roles in processing neighborhood information of nodes. These functions are employed to gather and combine features of neighboring nodes. The aggregator function is responsible for pooling information and features from neighboring nodes. Although this can be accomplished in various ways, GCNs typically employ a mean-based aggregation strategy. This means that all neighboring nodes are treated equally, and the node's features are updated using an average of its neighbors' features.

Following the feature update, the combinator function merges the node's updated features with the aggregated features from its neighbors[20]. Figure 3, visually illustrates the functioning of the aggregator function and how a node's features are updated.

In our BCGNN model, we employed three layers of GCN convolutions. These layers progressively transform input features into hidden representations. After each convolutional layer, we applied a ReLU activation function and a dropout layer. Activation functions, such as ReLU, introduce non-linearity into the model, enabling it to learn complex patterns. Dropout is used as a regularization technique to prevent overfitting. Overfitting occurs when a model becomes too specialized to the training data, hindering its ability to generalize to new data.

Following the third convolutional layer, a global mean pooling layer is applied. This layer aggregates the node features by computing their mean, providing

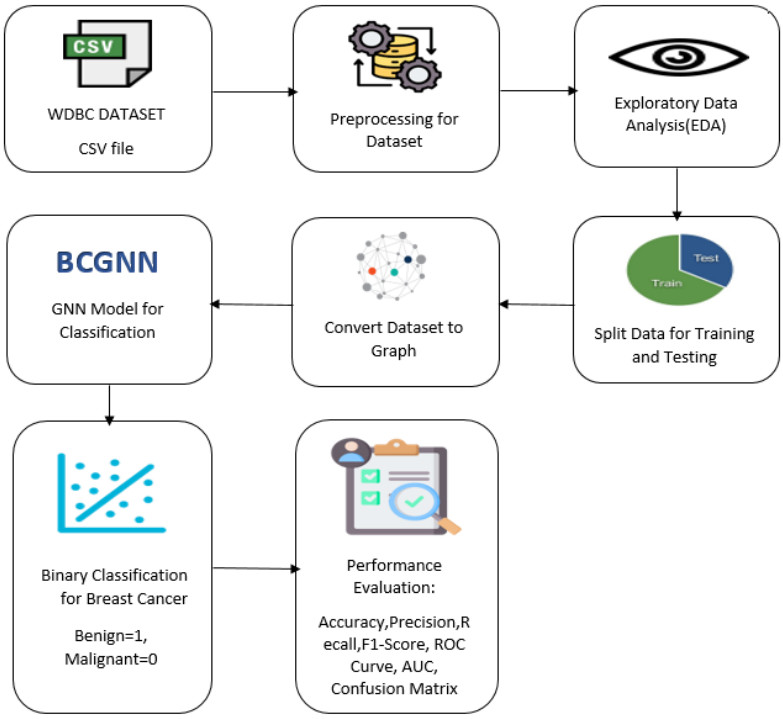

Figure 2: Model Overview. This overview can provide a good, overall understanding of the implementation process from start to finish.

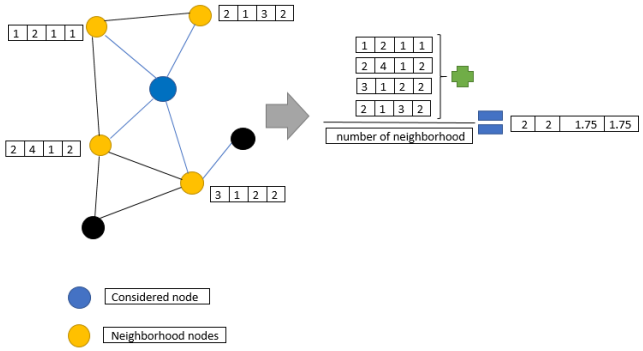

Figure 3: Aggregate approach structure for GCN.

a holistic representation of the graph's features. Our model also includes two linear layers. The first linear layer further transforms the aggregated features into hidden representations, followed by a ReLU activation and dropout. Finally, the second linear layer maps the hidden representations to the output classes, and a log-softmax function is used to produce class probabilities.

In the first convolutional layer of the GCN, the layer input consists of 35 features: 30 continuous features from the dataset and 5 local features related to the nodes. In the first convolutional layer, we have 128 output features that are passed to the next layer. The number of hidden dimensions is considered to be 128. In the second and third convolutional layers as well, the input and output features are 128. The first linear layer also receives 128 input features and helps to reach a new representation by combining features. The second linear layer, which is the last layer of the model, receives 128 features as input and converts them to 2 output features. These 2 outputs represent two classes in the classification problem. Figure 4 illustrates the overall architecture of the model used in this study.

4 Result and Discussion

4.1 Result

We have used various evaluation metrics to assess the performance of BCGNN model. Accuracy, precision, recall, and F1-score are commonly used metrics in evaluating the performance of classification models. The formulas for each of these metrics are provided in this section. Accuracy is defined as the ratio of the number of correct predictions to the total number of predictions, both correct and incorrect:

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN} \tag{2}$$

TP refers to the number of true positives, which are cases that were correctly predicted as positive. TN represents the number of true negatives, or cases that were correctly predicted as negative. FP denotes false positives, which are cases that were incorrectly predicted as positive. Finally, FN stands for false negatives, representing cases that were incorrectly predicted as negative.

The next evaluation metric is precision, which is defined as the ratio of true positives to all predicted positives. This metric tells us the proportion of instances that the model predicted as positive that were actually correct:

$$\text{Precision} = \frac{TP}{TP + FP} \tag{3}$$

Another evaluation metric is recall, which is calculated as the ratio of true positives to the total number of actual positives. This metric tells us what proportion of the actual positive cases were correctly identified by the model:

$$\text{Recall} = \frac{TP}{TP + FN} \tag{4}$$

The F1-score is a metric that combines both precision and recall, providing a balance between the two. It is particularly useful in situations where the classes are imbalanced:

$$\text{F1-Score} = 2 * \frac{Precision * Recall}{Precision + Recall} \tag{5}$$

We also utilized two additional evaluation metrics: the confusion matrix and the ROC curve. The confusion matrix is a valuable and essential evaluation metric for classification models, allowing us to identify the number of correct and incorrect predictions for each class. This matrix consists of four values: TP, TN, FP, and FN. The ROC curve, on the other hand, is another metric for evaluating the performance of binary classification models. It's a curve that illustrates the relationship between the true positive rate and the false positive rate. To evaluate the performance of our proposed model, Table 3 presents the precision, recall, and F1-score values associated with the average accuracy.

In Table 4, the accuracy achieved using the GNN model, as well as the number of features used from the dataset, are compared with previous and related works. In the accuracy calculation, the average accuracy is 98.83% with a standard deviation of 0.58%. The standard deviation of 0.58% means that, when running the code multiple times, the accuracy can vary. In some cases, the accuracy might be as low as 98.25%, while in others, it could be as high as 99.41%. However, the average accuracy across all these runs is 98.83%. By comparing the average accuracy with that of previous works, we observe an improvement in accuracy obtained by using the proposed GNN model.

A bar chart in Figure 5 visually compares the accuracy of our model to those found in previous research. This visual representation allows for a more granular comparison. Figure 6 and Figure 7 depict the ROC curve and confusion matrix, respectively, which were used to assess our model's performance.

4.2 Discussion

In the Results, the obtained results are presented using the aforementioned evaluation metrics. Our proposed model achieved an average accuracy of 98.83%. Additionally, precision, recall, and F1-score were all 99.07%. With these results, the GNN model outperformed the highest accuracy reported in previous related works by 1.17%. Even considering a standard deviation of 0.58% and comparing it to an accuracy of 98.25%, we still observe an improvement of 0.59%.
In [12], Utilizing KNN with 32 features from the WDBC dataset, the authors achieved an accuracy of 94.35%. Also In [13],The authors employed both Decision Trees (DT) and AdaBoost algorithms, with AdaBoost achieving a higher accuracy of 92.53%. Then [14], the highest accuracy was obtained using the XGBoost gradient boosting algorithm with 30 features, resulting in an accuracy of 96%. Finally, in [15], the authors achieved the highest accuracy in previous studies using an ensemble learning method with the

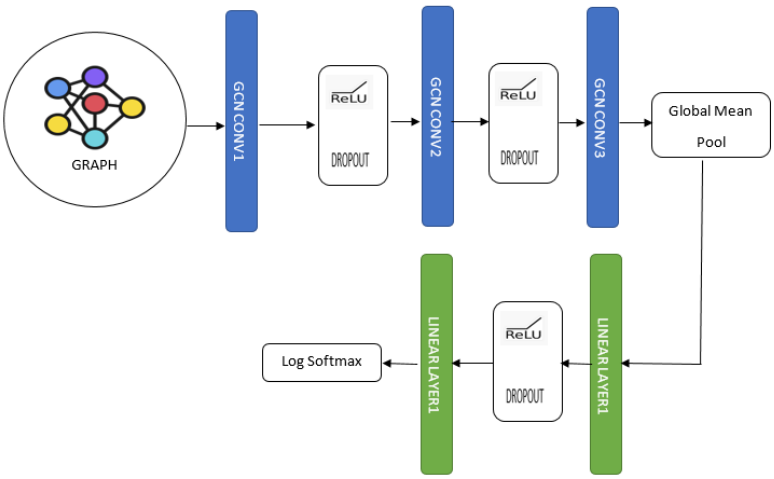

Figure 4: BCGNN Model architecture.

Table 3: Evaluation metrics values based on the average accuracy of 98.83% for BCGNN model.

Model	Accuracy	Precision	Recall	F1-Score
BCGNN	98.83%	99.07%	99.07%	99.07%

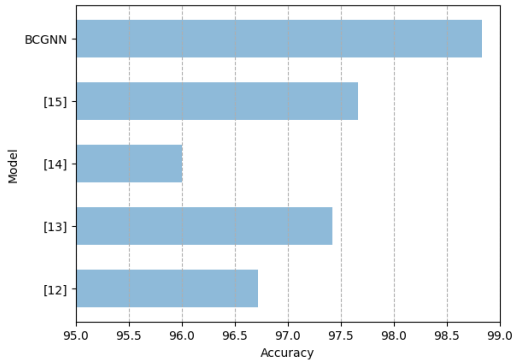

Figure 5: A bar chart to compare the accuracy achieved by BCGNN model and previous models.

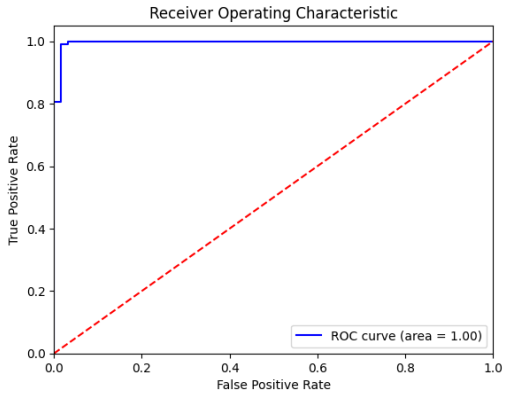

Figure 6: ROC curve to show the ability of BCGNN model to discriminate between classes(Accuracy=98.83%).

ELRL model and 9 features, reaching 97.66%.
Based on our findings, we have successfully achieved our goal of improving binary breast cancer classification accuracy. Our study has a positive impact on this field. However, a limitation of our research is the small sample size of the WDBC dataset, which contains only 569 samples. This relatively small sample size can be considered a limitation for classification tasks. Ultimately, our results can lead to more accurate and effective breast cancer diagnosis, facilitating cancer treatment.

5 Conclusion and Future Works

Diagnosis and classification of breast cancer have become very important in recent years due to the high incidence and mortality rates of this type of cancer among women.
In recent years, various machine learning models have been used to classify breast cancer. Also, one of the most common methods is ensemble learning, which by combining several models and using the features of each of those models, has been able to lead to good

Table 4: Comparison of BCGNN's model performance with those of previous and related works.

Reference	Model	Dataset	Samples	Features	Split Data	Accuracy
[12]	KNN	WDBC	569	32	Train=0.7,Test=0.3	94.35%
[13]	Adaboost	WDBC	569	31	Train=0.75,Test=0.25	92.53%
[14]	XGBoost	WDBC	569	30	Train=0.8,Test=0.2	96%
[15]	ELRL-E	WDBC	569	9	Train=0.7,Test=0.3	97.66%
Our Model	BCGNN	WDBC	569	31	Train=0.7,Test=0.3	**98.83±0.58%**

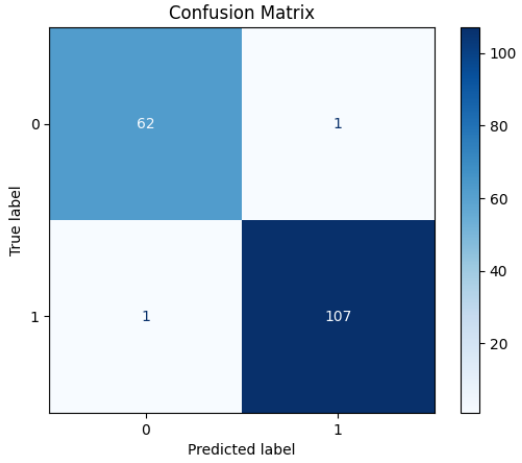

Figure 7: Confusion matrix as an evaluation metric to measure the classification accuracy of BCGNN model(Accuracy=98.83%).

results. In this study, we used BCGNN and due to the precise and layered structure of GNN, we achieved better results for binary classification of breast cancer. The average accuracy of 98.83%, precision=99.07%, recall=99.07%, F1-Score=99.07% are the results obtained from our use of the GNN method.
In the future works, we intend to address the standard deviation in accuracy. We will examine the main reasons for standard deviation and strive to mitigate or reduce its impact on model evaluation.we aim to combine our method with demographic and image data, aiming for more robust and reliable results. Through hyperparameter tuning and sensitivity analysis, we hope to achieve better performance. We will also further investigate the graph construction method and threshold value.

Code Availability

The code related to the implementation of our paper is accessible through this link: https://github.com/alireza-k-ch/BCGNN

Acknowledgements

We would like to express our sincere gratitude to Aso Mafakheri and Parsa Azimi, members of the Social and Biological Network Analysis Laboratory(SBNA) at University of Kurdistan, for their invaluable support.

References

[1] K. M. Cuthrell, N. Tzenios and K. Morton Cuthrell "Article no.IRJO.100349 Review Article Cuthrell and Tzenios," 2023. [Online]. Available: https://www.researchgate.net/publication/371069531

[2] A. N. Giaquinto et al. "Breast Cancer Statistics, 2022," *CA: A Cancer Journal for Clinicians*, vol. 72, no. 6, pp. 524–541, Nov. 2022, doi: 10.3322/caac.21754

[3] B. Nassif, A. Talib, Q. Nasir, Y. Afadar and O. Elgendy "Breast cancer detection using artificial intelligence techniques: A systematic literature review," Elsevier, 2022. [Online]. Available: https://www.cancer.org/

[4] L. T. Duong, C. Q. Chu, P. T. Nguyen, S. T. Nguyen and B. Q. Tran "Edge detection and graph neural networks to classify mammograms: A case study with a dataset from Vietnamese patients," *Applied Soft Computing*, vol. 134, Feb. 2023, doi: 10.1016/j.asoc.2022.109974

[5] M. R. Mohebian, H. R. Marateb, M. Mansourian, M. A. Mañanas and F. Mokarian "A Hybrid Computer-aided-diagnosis System for Prediction of Breast Cancer Recurrence (HPBCR) Using Optimized Ensemble Learning," *Computational and Structural Biotechnology Journal*, vol. 15, pp. 75–85, 2017, doi: 10.1016/j.csbj.2016.11.004

[6] S. Kapoor and A. Narayanan "Leakage and the reproducibility crisis in machine-learning-based science," *Patterns*, vol. 4, no. 9, Sep. 2023, doi: 10.1016/j.patter.2023.100804

[7] Y. Li "Performance Evaluation of Machine Learning Methods for Breast Cancer Prediction," *Applied and Computational Mathematics*, vol. 7, no. 4, p. 212, 2018, doi: 10.11648/j.acm.20180704.15

[8] M. Patrcio, J. Pereira, J. Crisstomo, P. Matafome, R. Seia and F. Caramelo "Breast Cancer Coimbra," UCI Machine Learning Repository, 2018. [Online]. Available: https://doi.org/10.24432/C52P59

[9] M. R. Basunia, I. A. Pervin, M. Al Mahmud, S. Saha and M. Arifuzzaman "On Predicting and Analyzing Breast Cancer using Data Mining Approach," in *2020 IEEE Region 10 Symposium (TENSYMP)*, Dhaka, Bangladesh, 2020, pp. 1257-1260, doi: 10.1109/TENSYMP50017.2020.9230871

[10] Z. Huang and D. Chen "A Breast Cancer Diagnosis Method Based on VIM Feature Selection and Hierarchical Clustering Random Forest Algorithm," *IEEE Access*, vol. 10, pp. 3284–3293, 2022, doi: 10.1109/ACCESS.2021.3139595

[11] W. Wolberg "Breast Cancer Wisconsin (Original)," UCI Machine Learning Repository, 1990. [Online]. Available: https://doi.org/10.24432/C5HP4Z

[12] T. A. Assegie "An optimized K-Nearest neighbor based breast cancer detection," *Journal of Robotics and Control (JRC)*, vol. 2, no. 3, pp. 115–118, May 2021, doi: 10.18196/jrc.2363

[13] T. A. Assegie, R. L. Tulasi and N. K. Kumar "Breast cancer prediction model with decision tree and adaptive boosting," *IAES International Journal of Artificial Intelligence*, vol. 10, no. 1, pp. 184–190, 2021, doi: 10.11591/ijai.v10.i1.pp184-190

[14] S. Laghmati, S. Hamida, K. Hicham, B. Cherradi and A. Tmiri "An improved breast cancer disease prediction system using ML and PCA," *Multimedia Tools and Applications*, vol. 83, no. 11, pp. 33785–33821, Mar. 2024, doi: 10.1007/s11042-023-16874-w

[15] A. Batool and Y. C. Byun "Toward Improving Breast Cancer Classification Using an Adaptive Voting Ensemble Learning Algorithm," *IEEE Access*, vol. 12, pp. 12869–12882, 2024, doi: 10.1109/ACCESS.2024.3356602

[16] W. Wolberg, O. Mangasarian, N. Street and W. Street "Breast Cancer Wisconsin (Diagnostic)," UCI Machine Learning Repository, 1993. [Online]. Available: https://doi.org/10.24432/C5DW2B

[17] Z. Wu, S. Pan, F. Chen, G. Long, C. Zhang and P. S. Yu "A Comprehensive Survey on Graph Neural Networks," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 32, no. 1, pp. 4-24, Jan. 2021, doi: 10.1109/TNNLS.2020.2978386

[18] J. Wang "A survey on graph neural networks," *EAI Endorsed Transactions on e-Learning*, vol. 8, no. 3, Jul. 2023, doi: 10.4108/eetel.3466

[19] X. Xu, X. Zhao, M. Wei and Z. Li "A comprehensive review of graph convolutional networks: approaches and applications," *Electronic Research Archive*, vol. 31, no. 7, pp. 4185–4215, 2023, doi: 10.3934/ERA.2023213

[20] S. Zhang, H. Tong, J. Xu and R. Maciejewski "Graph convolutional networks: a comprehensive review," *Computational Social Networks*, vol. 6, no. 1, Dec. 2019, doi: 10.1186/s40649-019-0069-y