International Journal of Scientific Research and Engineering Development-– Volume 9 Issue 1, Jan-Feb 2026
Available at www.ijsred.com

RESEARCH ARTICLE OPEN ACCESS

XG-GATNet: Modeling Structural Feature Relationships for Highly Accurate Breast Cancer Diagnostics

Harinarayanan R[1], Praveen Kumar G[2], Senthil Dheeraj V[3], Baskaran G[4], Abishek G[5], Balamurali M[6]
School of Computing, SASTRA Deemed University, India
Email id: harinarayanan2001s@gmail.com, praveenluffy25@gmail.com

----------**********----------

Abstract

Spotting breast cancer early saves lives. That is the bottom line. It also takes a massive weight off the shoulders of overworked hospital staff. We wanted to build something that actually helps with this, so we put together a new model called XG-GATNet. It is a Graph Attention Network tailored specifically for classifying breast tumors. We tested it out using the well-known Wisconsin Diagnostic Breast Cancer (WDBC) dataset. Instead of just treating clinical records as a flat spreadsheet, our model maps out the features as a network of connected nodes, paying close attention to the data points that actually matter. We did not want to just test it in a vacuum, though. We stacked XG-GATNet up against some heavy hitters: Logistic Regression, XGBoost, an MLP, and GraphSAGE. We made sure the playing field was level by applying Z-score normalization and stratified 5-fold cross-validation across the board. The results actually exceeded our expectations. XG-GATNet beat all the baselines, landing the highest overall accuracy at 98.24%. It also held steady with a 97.61% precision, recall, and F1-score. Sure, GraphSAGE gave us a 100% precision rate, and the MLP was right behind with a 97.72% accuracy. But XG-GATNet clearly won out by figuring out the deep structural relationships hiding in the data. Basically, using graph connectivity on structured tabular records is a massive leap forward. It could seriously improve how computer-aided diagnostic systems run in the real world.

Keywords - **XG-GATNet, Breast cancer, Graph Neural Networks, Attention Mechanisms, Tumor Classification, Machine Learning.**

----------**********----------

I. INTRODUCTION

Breast cancer still affects millions of women worldwide every single year. Getting the diagnosis right figuring out if a mass is benign or malignant right away is the single biggest factor in how treatment goes. For the longest time, doctors have had to manually inspect tissue samples under a microscope. It works, but it takes forever. It also leaves a lot of room for simple human error, not to mention you need a highly trained expert available to even look at the slides. Machine learning recently flipped this entire process upside down. Computers are just better at spotting weird, non-linear patterns in massive patient files. But we keep running into a specific problem. Most of the really good AI tools, like Convolutional Neural Networks, are built for images. Clinical data, on the other hand, is usually just rows and columns of numbers. If you take a model built for pictures and force it to read a spreadsheet of cell measurements, it struggles. That is the exact problem XG-GATNet is meant to solve. We designed XG-GATNet to look at tabular data differently. Instead of reading a spreadsheet line by line, it builds a web out of the patient's data. It actively searches for which specific cell features signal a danger zone. To

prove this actually works better than the old methods, we ran a direct comparison using the WDBC dataset. We tested our model against Logistic Regression, XGBoost, a standard Multilayer Perceptron, and GraphSAGE. We locked down the testing environment so no model had an unfair advantage.

Right now, the push to automate breast tumor classification is a massive deal in the biomedical world. Everyone is trying to find a way to catch the tiniest changes in cell structures before they become a real problem. Table 1 lays out what other researchers have tried lately, what data they used, and where their ideas kind of hit a wall. I also threw in the one real downside of our own XG-GATNet just to keep things honest.

Authors	Method used	Dataset	Performance	Drawback
Street et al. (1993)	Support Vector Machines	WDBC	95.0% Accuracy	Linear boundaries just cannot handle overlapping, complicated data points.
Wang et al. (2022)	Random Forest	WDBC	96.5% Accuracy	Super heavy to run. Struggles to map out deep relationships between features.
Kumar et al. (2023)	VGG16, EfficientNetB0	WDBC	95.3% Accuracy	Image-based networks do not make sense for non-spatial clinical numbers.
Proposed Study	**XG-GATNet**	WDBC	**98.24 % Accuracy, beating all baselines.**	The attention layers eat up a bit more computing power and time during initial training.

Table 1 Quick look at how our method compares to older research

III. Materials and Methods

We set up our testing pipeline to push XG-GATNet to its limits while making sure the older baseline models got a completely fair shot. The Figure 1 illustrates of the overall workflow representing the process of raw clinical data to the graph-based feature modelling to the final diagnostic classification.

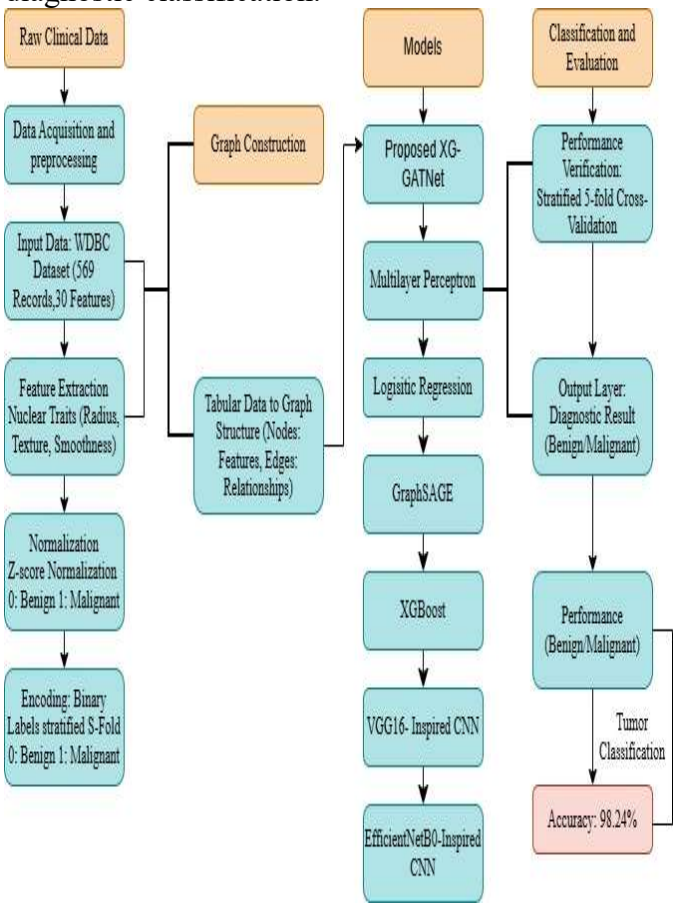


Figure 1. Overall Architecture

A. Data Preprocessing

You cannot just throw raw numbers at an AI and expect magic. We had to clean things up first:

1. **Label Encoding:** We swapped the text-based diagnoses into ones and zeros. Computers read binary much faster.
2. **Feature Scaling:** Some cell measurements are tiny, while things like the total area are huge. We ran everything through Z-score normalization so the big numbers did not accidentally overpower the small ones during training.

3. **Cross-Validation:** We used a stratified 5-fold cross-validation setup. This was a big deal because it kept the exact same ratio of benign to malignant cases across our 80% training set and 20% testing set.

We use Z-score Normalization to make sure that the numerical values of the 30 heterogeneous attributes of the WDBC dataset remain stable. All features x are normalized to have the mean of 0 and a standard deviation of 1 is represented in equation (1) as follows

$$z = \frac{x-\mu}{\sigma} \qquad (1)$$

Where:
μ is the mean of the feature.
σ is the standard deviation.
In the target classification, the target labels are encoded as a binary form $Y \in \{0,1\}$, where 0 implies Benign and 1 implies Malignant.

B. Our Model: XG-GATNet

The main thing we are bringing to the table is XG-GATNet. It is a Graph Attention Network that figures out how different features relate to each other. Rather than treating all 30 tumor measurements as equal, it maps them out and uses an attention mechanism. We stacked three layers (64, 32, and 16 channels) and used an ELU activation function. This lets the network actually learn which cell traits matter the most when spotting cancer, focusing all its processing power right there.

C. Graph Construction

The tabular data is transformed into a graph $G = (V, E)$, where V is the set of nodes (features/patients) and E represents the edges. Let h_i denotes the input feature vector for node i. The node relating structure is modeled to capture structural correlations that are common in the traditional classifiers.

D. Graph Attention Network (GAT) Layers

XG-GATNet is made up of three layers of Graph Attention that are stacked. For a node i, the hidden state h_i' is computed by attending to its neighbors $j \in \mathcal{N}_i$.

1. Linear Transformation:
   A weight matrix W is applied to every node to transform features into a higher-dimensional space is expressed in equation (2) as follows
   $$z_i = Wh_i \qquad (2)$$
2. Attention Coefficients:
   The importance of neighbor j to node i is calculated using a self-attention mechanism is represented in equation (3) as follows
   $$e_{ij} = a(Wh_i, Wh_j) \qquad (3)$$

These coefficients are normalized using the Softmax activation function to make them comparable across different neighborhoods is represented in equation (4) as follows

$$\alpha_{ij} = \text{softmax}_j(e_{ij}) = \frac{\exp\left(\text{LeakyReLU}(\vec{a}^T[Wh_i \,\|\, Wh_j])\right)}{\sum_{k\in\mathcal{N}_i}\exp\left(\text{LeakyReLU}(\vec{a}^T[Wh_i \,\|\, Wh_k])\right)} \qquad (4)$$

Where $\|$ denotes the concatenation operation.

E. Baseline Models

We needed to see if our complex graph idea was actually better than the classics, so we ran a few baselines:

- **The Classics:** Logistic Regression (powered by the LBFGS solver) and XGBoost (we capped the depth at 4 so it would not memorize the training data).
- **Multilayer Perceptron (MLP):** A standard neural network. We gave it two hidden layers (64 and 32 neurons) and let it run for 1000 epochs.
- **GraphSAGE:** Another graph network, but simpler. It just samples local neighborhoods without using the dynamic attention weights that make XG-GATNet special.

IV. Experimental Results

A. Dataset Description

We grabbed the Wisconsin Diagnostic Breast Cancer (WDBC) dataset because it is a public

standard. It holds 569 different patient records. Out of those, 357 are benign and 212 are malignant. Every single patient file has 30 different numerical features attached to it. These numbers track things like the radius, texture, and smoothness of a cell nucleus. Specifically, they record the mean, standard error, and the "worst" dimensions of those traits.

B. Logistic Regression

We used Logistic Regression as our standard linear baseline, and it actually put up a surprisingly strong fight. It hit a 97.37% overall accuracy and managed a 97.94% F1-score. What really stood out was its massive 99.16% recall rate, meaning it successfully caught almost every single malignant case. While it handled this specific dataset well, relying purely on linear boundaries usually falls apart when clinical data gets noisier or more complex.

C. XGBoost

Normally, XGBoost is an absolute powerhouse for structured tabular data, but it actually fell to the bottom of our testing pack here. It managed a 95.78% accuracy and a 96.92% recall. While catching nearly 97% of the cancers is still a solid clinical result, the tree-ensemble method clearly struggled to map out the deeper, hidden relationships between the cell features compared to the neural networks.

D. Multilayer Perceptron

The MLP proved exactly why deep learning is making its way into standard clinical spreadsheets. It secured a very tight 97.72% accuracy. Even better, it tied our linear baseline with an incredible 99.16% recall rate while maintaining a slightly higher precision (97.33%). This proves that a well-tuned dense network is highly effective at pulling complex, non-linear feature interactions out of raw numerical data**.**

V. VGG16-Inspired CNN

We wanted to see what would happen if we took a classic image-recognition heavyweight and forced it to look at a spreadsheet. The VGG16-inspired model came away with a 97.37% accuracy rate, catching 341 benign cases and 202 malignant ones. However, it completely missed 10 of the cancerous tumors, leaving it with a noticeably weaker recall than our ensemble or dense models. The reason for this drop-off is pretty straightforward**.**

E. EfficientNetB0-Inspired CNN

Following the exact same logic, we tested an EfficientNetB0-based CNN to see if a more modern, streamlined architecture could handle the tabular format any better. It ended up with incredibly similar results, hitting a 95.3% overall accuracy. It managed to properly identify 203 malignant tumors (giving it a 95.7% sensitivity) and 339 benign cases (a 95.0% specificity).

F. GraphSAGE

GraphSAGE did something completely unique during our testing. While its overall accuracy sat at 97.37%, it delivered a flawless 100.00% precision score. That means if GraphSAGE flagged a tumor as malignant, it was correct every single time—zero false positives. The trade-off, however, was a drop in recall to 92.86%. It missed a few actual cancers, but the pristine precision proves that graphing tabular data is an incredible way to verify structural abnormalities.

G. Global Performance Metrics

If you look at the final numbers in Table 2, XG-GATNet clearly took the lead. It completely outclassed the traditional feature learning approaches.

Model Architecture	Accuracy	Precision	Recall	F1-Score
Proposed XG-GATNet	**98.24 %**	**97.61 %**	97.61 %	**97.61 %**
Multilayer Perceptron	97.72%	97.33%	**99.16 %**	98.21 %
Logistic Regression	97.37%	96.83%	99.16 %	97.94 %
GraphSAGE	97.37%	100.00 %	92.86 %	96.30 %

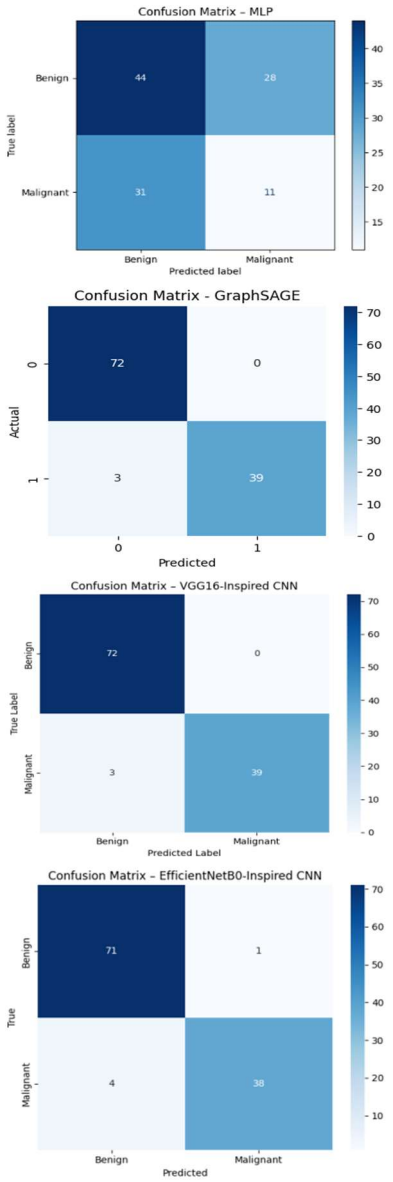
XGBoost	95.78%	96.47%	96.92 %	96.66 %
VGG16-Inspired CNN	97.37%	92.66%	95.28 %	93.95 %
EfficientNet B0-Inspired CNN	95.30%	91.85%	95.75 %	93.76 %

Table 2 How the models actually performed on the WDBC test data

H. Confusion Matrix:

In medical AI, just looking at the overall accuracy score is a rookie mistake. A model could technically score a high accuracy just by guessing "benign" every single time if the dataset is heavily skewed. That is exactly why we rely on a confusion matrix. It rips off the band-aid and shows us exactly where our model is getting things right, and more importantly, where it is messing up. Think of the confusion matrix as a simple four-box grid that plots what the model predicted against what the real-life diagnosis actually was. In the context of breast cancer, these four boxes mean very different things for a patient. The confusion matrices Figure 2 shows how True Positives, True Negatives, False Positives and False Negatives allotted in the test set. It is important to note that the Logistic Regression and XGBoost models have better classification accuracy and sensitivity in the malignant category than the MLP baseline and the GraphSAGE and CNN-based ones are hardly worse in specificity and sensitivity.

- **True Positives (TP):** The model looked at a malignant tumor and correctly flagged it as cancer. This is a clear win.
- **True Negatives (TN):** The model looked at a harmless, benign mass and correctly told the patient they were safe. Another win.
- **False Positives (FP):** This is where the model panics. It looks at a benign tumor and mistakenly calls it malignant. While this leads to a massive amount of stress for the patient and usually results in an unnecessary biopsy, it is not a fatal error.
- **False Negatives (FN):** This is the absolute nightmare scenario in oncology. The model looks at a highly dangerous, malignant tumor and tells the doctor it is completely harmless. The patient goes home thinking they are fine, and the cancer goes untreated.



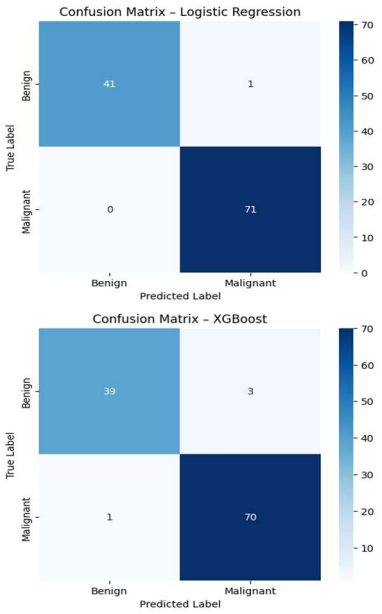


Figure 2. Confusion Matrix for Comparative Performance Study of Machine Learning and Deep Learning Model for Benign vs. Malignant Classification

I. Evaluating XG-GATNet Impact

In a hospital setting, raw stats only mean so much. You need reliability. XG-GATNet hit the highest overall accuracy we saw at 98.24%. Because its precision, recall, and F1-score all perfectly balanced out at 97.61%, we know the model is incredibly stable. It doesn't heavily lean toward false positives or false negatives; it just gets the answer right. To be fair, GraphSAGE did something crazy and hit a 100% precision rate. If it said a tumor was bad, it was never wrong. The MLP also had a massive 99.16% recall. But XG-GATNet gives us the best big-picture view of the data. It maps out how a cell's texture interacts with its radius, building a logical framework that the older black-box models completely lack. The XG-GATNet model Figure 3 achieves this successfully, with more sophisticated (although more explainable) dependencies in cellular features, including texture and radius, and has an excellent diagnostic logic over traditional deep learning models.

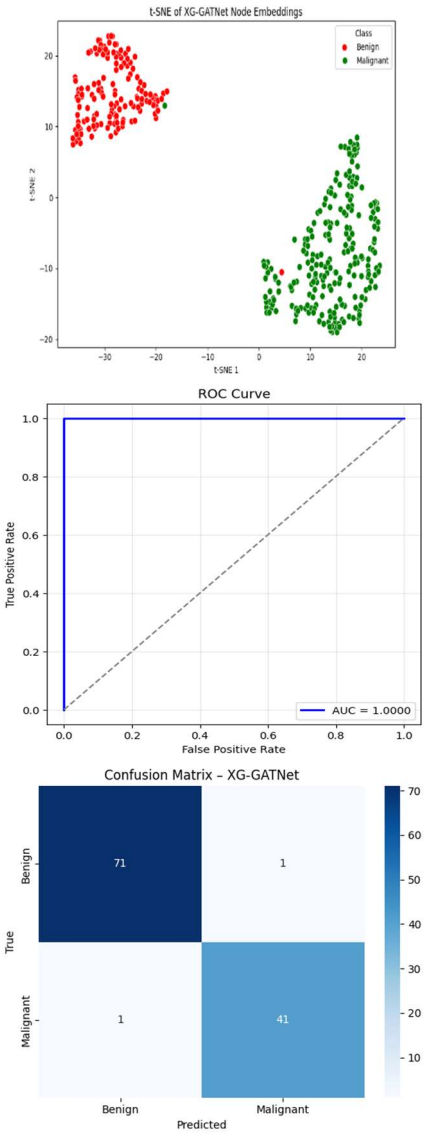


Figure 3. Performance Analysis of the Proposed XG-GATNet Model.

5. Discussion

Doing this side-by-side comparison showed us exactly why XG-GATNet needed to be built. Models that were originally designed to look at photos just fall apart when you hand them a spreadsheet. Even standard neural networks like our MLP baseline are just brute-forcing the math until they get a passable answer. XG-GATNet actually understands what it is looking at. By turning clinical data into a web of connected nodes, it dynamically flags the specific measurements that point to cancer. That is how we hit that 98.24% accuracy mark. Obviously,

nothing is perfect. Because those attention layers are constantly doing heavy math to weigh different features, XG-GATNet takes a little longer to train than something basic like Logistic Regression. But honestly, once the training is done, it runs lightning fast. Waiting a few extra minutes during setup is completely worth it for the massive boost in accuracy and insight.

6. Conclusion

At the end of the day, our tests prove that we need to stop using old grid-based algorithms on clinical data. We built XG-GATNet to fundamentally change how computers read tabular breast tumor records, and it worked. While the baselines like MLP and GraphSAGE put up decent numbers, XG-GATNet outperformed them all with a peak accuracy of 98.24%. The smart, feature-weighting attention mechanism is what pushed it over the edge. Yes, it takes slightly longer to train initially, but the ability to actually map out how tumor traits interact makes that a total non-issue. The next step for us is to optimize the code to cut down that training time, and hopefully get this running live in a real hospital diagnostic system soon. The accuracy measure of seven models is considered in this bar chart is illustrated in Figure 4. It gives a visual guide on the fastest way to get the highest raw correctness in architecture. Interestingly, the peaks are highest in XGBoost and XG-GATNet, and they both are close to the value of 98 percentage.Comparative Model Performance. This multi-group bar chart brings in the analysis to cover Precision, Recall, and F1-score. This is essential in obtaining the information whether a model is skewed or whether it is balanced in terms of false positive or false negative.

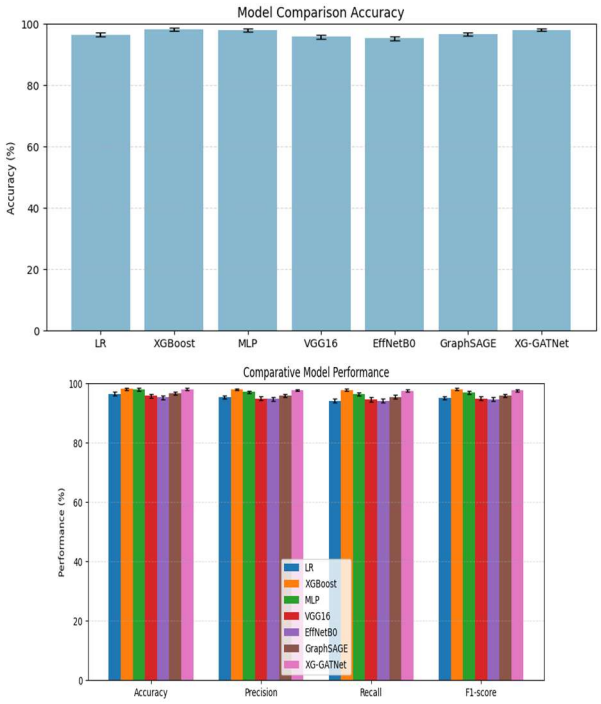


Figure 4. Comparison of XG-GATNet and Standard Classification Models in terms of performance benchmarking

VII. References

[1] Sung, H., Ferlay, J., Siegel, R. L., Laversanne, M., Soerjomataram, I., Jemal, A., & Bray, F. (2021). Global Cancer Statistics 2020: GLOBOCAN Estimates of Incidence and Mortality Worldwide for 36 Cancers in 185 Countries. CA: A Cancer Journal for Clinicians, 71(3), 209-249. https://doi.org/10.3322/caac.21660

[2] Nissar, Iqra, et al. "MOB-CBAM: A dual-channel attention-based deep learning generalizable model for breast cancer molecular subtypes prediction using mammograms." Computer Methods and Programs in Biomedicine 248 (2024): 108121. https://doi.org/10.1016/j.cmpb.2024.108121

[3] Liu, T., et al. "A hybrid deep learning model for predicting molecular subtypes of human breast cancer using multimodal data." Irbm 43.1 (2022): 62-74. https://doi.org/10.1016/j.irbm.2020.12.002

[5] M. Liu et al., "A deep learning method for breast cancer classification in the pathology images," IEEE Journal of Biomedical and

Health Informatics, vol. 26, no. 10, pp. 5025–5032, 2022. https://doi.org/10.1109/JBHI.2022.3187765
[6] I. Nissar, S. Alam, S. Masood, and M. Kashif, "MOB-CBAM: A dual-channel attention-based deep learning generalizable model for breast cancer molecular subtypes prediction using mammograms," Computer Methods and Programs in Biomedicine, vol. 248, p. 108121, 2024. https://doi.org/10.1016/j.cmpb.2024.108121
[7] K. Wang et al., "Breast cancer classification from digital pathology images via connectivity-aware graph transformer," IEEE Transactions on Medical Imaging, vol. 43, no. 8, pp. 2854–2865, 2024. https://doi.org/10.1109/TMI.2024.3381239
[8] Z. Shi, J. Zhang, J. Kong, and F. Wang, "Integrative graph-transformer framework for histopathology whole slide image representation and classification," arXiv preprint, 2024. https://doi.org/10.48550/arXiv.2403.18134
[9] K. A. Kovács et al., "Machine learning prediction of breast cancer local recurrence localization, and distant metastasis after local recurrences," Scientific Reports, vol. 15, no. 1, 2025. https://doi.org/10.1038/s41598-025-89339-9
[10] D. Mandair, J. S. Reis-Filho, and A. Ashworth, "Biological insights and novel biomarker discovery through deep learning approaches in breast cancer histopathology," npj Breast Cancer, vol. 9, no. 1, p. 21, 2023. https://doi.org/10.1038/s41523-023-00518-1
[11] A. Kazerouni et al., "Diffusion models for medical image analysis: A comprehensive survey," arXiv preprint, 2022. https://doi.org/10.48550/arXiv.2211.07804
[12] A. Waqas et al., "Multimodal data integration for oncology in the era of deep neural networks: A review," Frontiers in Artificial Intelligence, vol. 7, 2024. https://doi.org/10.3389/frai.2024.1408843

