Breast Cancer Subtype Classification Based on Optimized Backpropagation Neural Network

Joung Min Choi
Department of Computer Science
Virginia Tech
Blacksburg, VA, US
joungmin@vt.edu

Haoqiu Song
Department of Computer Science
Virginia Tech
Blacksburg, VA, US
haoqiu@vt.edu

Jiayue Zhou
Department of Computer Science
Virginia Tech
Blacksburg, VA, US
jiayuez@vt.edu

Wei Liu
Department of Statistics
Virginia Tech
Blacksburg, VA, US
wliu3@vt.edu

Yuze Li
Department of Computer Science
Virginia Tech
Blacksburg, VA, US
lyuze@vt.edu

Zhen Shen
Department of Computer Science
Virginia Tech
Blacksburg, VA, US
zeroshen@vt.edu

Weitong Li
Department of Computer Science
Virginia Tech
Blacksburg, VA, US
weitongli@vt.edu

Abstract

Breast cancer is one of the highly heterogeneous diseases composed of multiple biological factors. Due to this diversity, patients have shown different prognosis and clinical outcomes. To provide a better therapeutic decision, a standardized breast cancer subtyping system, PAM50, has been developed. Early diagnosis and precise subtype prediction play a critical role in providing proper treatment for each patient. To accurately predict the breast cancer subtype, neural network-based classification models have been proposed utilizing gene expression profiles of breast cancer patients. However, they still suffer from the model optimization showing low-performance improvement and one of the issues could be due to the low searching efficiency and the slow convergence of backpropagation (BP). Recently, some optimization algorithms to initialize model parameters with optimal values for later training have been suggested for the BP neural network have been suggested. These approaches have been adopted in several fields including image classification, and the case studies showed that those techniques showed the potential to improve the optimization and prediction performance. Regarding this, those algorithms also might be helpful for breast cancer subtype classification tasks.

In this paper, we presented optimized backpropagation neural network models for breast cancer subtype classification based on six optimization algorithms utilizing gene expression profiles. The optimization algorithms were investigated for their potential to improve the performance of breast cancer subtype prediction tasks.

They were evaluated and compared to the widely-used Glorot and random initialization in the aspect of f1-score and accuracy. From the experiment results, the classification performance showed high variance under different optimization, and the grey wolf optimization and Glorot showed stable performance compared to the others. Our implementation of BP-optimized breast cancer subtype classification models is publicly available at https://github.com/joungmin-choi/CS5824-final.

1 Introduction

Breast cancer is one of the leading causes of death among women worldwide [1]. It is a highly heterogeneous disease composed of diverse genetic, epigenetic, and phenotypic modifications, which is also affected by the metabolism and micro-environment [2]. This diversity caused different clinical outcomes for breast cancer patients, and it has been a challenge to provide accurate diagnosis and proper treatment. To better predict the prognosis of patients, approaches have been made to divide breast tumors into several subtypes showing distinct patterns leading to different treatment responses. Sorlie et al. classified breast cancer into five intrinsic subtypes, i.e., luminal A, luminal B, HER2 over-expression, basal and normal-like tumors based on the molecular profiles utilizing the microarrays technique [3]. These subtypes have been repeated by several other studies based on gene expression profiling, and Parker et al. presented the PAM50 model based on the signature genes related to a hormone receptor, proliferation, and myoepithelial and basal features [4], which has become a standardized breast cancer subtyping system. Precise prediction of breast cancer subtypes based on the PAM50 subtypes has become a key role in cancer prognosis, which helps for therapeutic decisions and patient outcome improvement [5].

In recent years, to accurately predict the subtypes of breast tumor patients, several classification methods based on machine learning have been presented utilizing gene expression profiles. Yu et al. identified the differentially expressed genes which are biologically important in gene regulatory networks for each breast cancer subtype and constructed binary classifiers for each subtype based on several machine learning algorithms including K-nearest neighborhood, support vector machine, and random forest [6]. Gao et al. proposed an algorithm to measure the enrichment score for each gene utilizing the patient sample's gene expression profiles and developed a cancer molecular subtype classification based on the fully-connected neural network model using those scores, presenting a case study in breast cancer [7]. Beykikhoshk et al. presented a framework to classify luminal A and luminal B subtypes by calculating the personalized biomarker scores based on the attention mechanism [8]. Moreover, several approaches to incorporate biological knowledge and gene relations have been suggested for predicting breast cancer subtypes. Rhee et al. suggested a hybrid framework based on the graph convolution neural network (GCN) and relation network integrate the gene expression profiles and protein-protein interaction network [9]. Lee et al. developed a multi-attention-based GCN model for cancer subtype prediction utilizing pathway networks and presented a case study for classifying breast cancer subtypes [10].

The presented neural network-based models have shown comparable classification results to the existing classifiers. However, they still suffer from model optimization showing low-performance improvement. One of the main reasons could be the high degree of heterogeneity of breast tumors and various biological factors. Another issue would be that those models use backpropagation (BP)-based optimization, which is a classical and fundamental method. But, with regard to the low searching efficiency and the slow convergence speed in the BP algorithm, some optimization algorithms have been proposed for the BP neural network to achieve better results [11, 12]. The main idea is to make the initial parameters (e.g. weights and biases) in BP neural network optimal for the later training phase based on the algorithms, rather than using the randomly initialized parameters. Recently, this approach has been adopted in some fields such as image classification, and several case studies have been performed. Song et al. showed that the sine cosine algorithm (SCA)-optimized BP neural network improved the prediction accuracy for classifying multi-class images [13]. Bacanin et al. also investigated the performance of the convolutional neural network model for image classification when utilizing the sine cosine algorithm and firefly algorithm with different benchmark datasets, and obtained the experimental results indicating that the proposed method outperforms other baseline methods [14]. These results suggest the potential of the optimized BP algorithm to be utilized in other domains for performance improvement, which leads us to think of its application to the breast cancer subtype classification task.

In this paper, we presented the optimized backpropagation neural network models for breast cancer subtype classification utilizing gene expression profiles and investigated their potential for prediction performance improvement. After data preprocessing and feature selection, we implemented six optimization algorithms of SCA, whale, ant lion, firefly, moth-flame, and grey wolf optimizer for the BP parameter initialization and constructed a neural network-based breast cancer classifier. The performance of each optimizer was compared with the random and Glorot (also known as Xavier) initialization and we showed how each algorithm works for optimization.

2 Materials and Methods

2.1 Data collection

The breast cancer (BRCA) datasets were collected from The Cancer Genome Atlas (TCGA) [15], where gene expression profiles of breast cancer patients based on RNA-seq were used in this study. Breast cancer subtype information based on PAM50 for each TCGA BRCA sample was retrieved from [4], and the total of 1,059 samples was divided into 5 subtypes as shown in Table 1. For feature selection, 113 normal samples were also obtained.

Table 1: The number of breast cancer samples for each subtype

Breast cancer subtypes	Number of samples
Luminal A	556
Luminal B	200
Her2-enriched	182
Basal-like	81
Normal-like	40

2.2 Preprocessing

RNA-seq datasets typically reflect biological heterogeneity and technical biases [16]. To eliminate the effects caused by these issues, we first removed genes having no read count for all samples. After that, size factors were calculated, and read counts were normalized by library size, and log-transformed using DESeq2 R package [17].

2.3 Feature selection

To prevent the possibility of overfitting and expensive computational cost in training the neural network model, feature selection was performed. First, the genes remaining after preprocessing were evaluated for their informativity as a breast cancer signature by performing differential analysis that compared to the normal samples using DESeq2. Genes with an absolute value of log fold-change greater than 2 and an adjusted p-value less than 0.01 were considered as differentially expressed genes (DEG). We selected 1,000 DEGs having the highest log fold-change values and the lowest adjusted p-value and used them as final features for breast cancer subtype classification.

2.4 Optimization

2.4.1 Sine cosine algorithm

Sine cosine algorithm (SCA)[18] is a population-based optimization algorithm. By taking full advantage of the characteristics of randomness and trigonometric function, this algorithm has managed to make the search scope more comprehensive during the whole search process. When there is a large number of particles aiming to search for the global optimum solution of an objective function, each of them will use a particular position updating equation to determine its next destination randomly. Through transmitting and recording messages in every iteration process, all the particles will update their own information and then compare their information with the global optimum solution. After that, one can transmit the information to others and memorize it for future comparisons. Over and over again, through enough iteration times, all the particles may eventually reach the most optimal solution of the objective function.

To be more specific, suppose that there is a set of particles, which are used to find the global optimal solution. When using SCA to search for the optimal solution of the chosen objective function, first, initialize all these particles and randomly choose one to be the temporary best solution. Then make evaluations of the current position through the objective function, and use all information that the particles have got to update the current global optimal value. Finally, use the particular position updating function to determine the position of their next movements.

To determine the next destination of each particle, four parameters are used to find it. Since the searching space of SCA for each particle is a circular area, for those parameters, their usages are to determine the direction of the next search movement, get different sine or cosine values, and decide whether the destination point should be an important factor in determining the next position, and makes it possible to use the sine function or the cosine function with equal probability.

2.4.2 Whale optimization algorithm

Whale optimization algorithm (WOA)[19] is a recently proposed stochastic optimization algorithm. It utilizes a population of search agents to determine the global optimum for optimization problems. Similarly to other population-based algorithms, the search process starts with creating a set of random solutions for a given problem. It then improves this set until the satisfaction, which is the Bubble-net hunting behavior of an end criterion. The main difference between WOA and other algorithms is the rules that improve the candidate solutions in each step of optimization.

WOA mimics the hunting behavior of humpback whales in finding and attacking prey called bubble-net feeding behavior. Humpback whales prefer to hunt small fishes close to the surface. The leader whale finds the prey and dives down creating a spiral-shaped bubble around the prey, and then swimming up to the surface while following the bubbles. An experienced whale who supports the leader will make a call to synchronize. The others form a formation behind the leader and take the same position for each lunge. This intelligent foraging method is the main inspiration of the WOA.

In the WOA algorithm, it uses a mathematical model, where it first randomly selects prey and then performs optimization following the spiral-shaped path. Another simulated behavior of humpback whales in WOA is the encircling mechanism. Humpback whales circle around preys to start hunting them using the bubble-net method. The WOA starts optimizing a given problem by creating a set of random solutions. In each step of optimization, search agents update their positions based on a randomly selected search agent or the best search agent obtained so far. To guarantee exploration and convergence, the best solution is the pivot point to update the position of other search agents.

2.4.3 Ant lion optimization

Ant lion optimization (ALO)[20] algorithm mimics the interaction between antlions and ants in the trap. An antlion larvae digs a cone-shaped pit in the sand by moving along a circular path and throwing out sands with its massive jaw. After digging the trap, the larvae hides underneath the bottom of the cone and waits for insects to be trapped in the pit. The edge of the cone is sharp enough for insects to fall to the bottom of the trap easily. Once the antlion realizes that the prey is in the trap, it tries to catch it. However, insects usually are not caught immediately and try to escape from the trap. In this case, antlions intelligently throw sands towards to edge of the pit to slide the prey into the bottom of the pit. When prey is caught in the jaw, it is pulled under the soil and consumed. After consuming the prey, antlions throw the leftovers outside the pit and amend the pit for the next hunt.

To model such interactions, ants are required to move over the search space, and antlions are allowed to hunt them and become fitter using traps. Ants move stochastically in nature when searching for food, a random walk is chosen for modeling ants' movement. The ALO algorithm takes three inputs, as *ALO (A, B, C)*. A is a function that generates the random initial solutions for ants and antlions. Function B manipulates the initial population provided by function A, and C returns true when the end criterion is satisfied.

In the ALO algorithm, the antlion and ant matrices are initialized randomly using the function A. In every iteration, function B updates the position of each ant with respect to an antlion selected by the roulette wheel operator and the elite. The boundary of position updating is first defined proportional to the current number of iterations. The updating position is then accomplished by two random walks around the selected antlion and elite. When all the ants randomly walk, they are evaluated by the fitness function. If any of the ants become fitter than any other antlions, their positions are considered as the new positions for the antlions in the next iteration. The best antlion is compared to the best antlion found during optimization (elite) and substituted if it is necessary. These steps are iterative until function C returns false.

2.4.4 Firefly optimization

Firefly algorithm (FA)[21] mimics imitating the exchange of information, attracting each other, and alerting danger between fireflies. In this algorithm, the position of each firefly represents a feasible solution to the problem that needs to be solved, and the brightness of the firefly represents the fitness of the firefly position. The brighter the firefly is, the better position of the individual firefly in the solution space. Among the fireflies, high-brightness fireflies will attract low-brightness fireflies. In the solution space, each firefly will fly around like a bright firefly, and they are looking for a better location. The brighter the firefly is, the more attractive it is to other fireflies. At the same time, the light-transmitting medium between fireflies will absorb light, and reduce the brightness of a light. Also, it will affect the transmission of light, so the attraction between fireflies will be inversely proportional to the spatial distance. It means that the attraction between two fireflies decreases when the distance between the two fireflies increases.

In the FA algorithm, the number of fireflies and several variables need to be determined including a parameter controlling the step size, the initial attractive value, and the absorption coefficient of light by the medium. As for the second step, the fitness value of each firefly is calculated according to the position of the firefly. The better the fitness value is, the brighter the firefly is. Then, each firefly will fly to the fireflies with a higher brightness than its brightness, and the brightest firefly in the group will not update its position. This step of the process will be completed after ranking. If the algorithm reaches the maximum number of iterations, the searched optimal firefly position will be output as the solution. Otherwise, the FA algorithm will go back to the second step to repeat.

2.4.5 Moth-flame optimization

Moth-flame algorithm (MFO) [22] is inspired by the natural behavior of a moth flying toward a flame. This algorithm simulates the death behavior of moths, which is a spiral flying path for each moth. The fact about moths is that they would fly with lights by a specific path. By observing the path, this algorithm gives an objective optimizer.

In the MFO algorithm, it is assumed that the agents are moths and the problem's variables are the position of moths in the space. At first, the position of moths is rendered randomly. And at each iteration, each moth would be randomly changed to another position on the spiral curve, and keep the best results for the aggregate moths(original and new positions for all the moths). The basic rule for changing the position of moths is that the closer the moth gets to the previous corresponding best answer, the less position change will be. With this rule, moths are able to possibly get closer to the global best solution but have chances to jump out the local optimal solution. Besides, the MFO algorithm sets two search borders as search space, any moths that fly out of borders would be set on the border so to control the search space.

The MFO algorithm takes six parameters for its implementation, which are objective function, lower bound, upper bound, N for agents number, and max iterations for the maximum allowed iterations. For each iteration, the position of every moth would be updated, and the MFO algorithm uses the objective function to decide whether they are better after changing and to pick up the best moths.

2.4.6 Grey wolf optimization

Grey Wolf Optimization (GWO) is a new meta-heuristic optimization technique proposed by Mirjalili et al. in 2014 [23]. Meta-heuristic becomes a popular optimization technique recently because it is simple, derivation-free, flexible to be used in many applications, and has the ability to avoid local optima. Generally, the search process in meta-heuristic consists of exploration and exploitation. Exploration randomly and globally searches the promising area, and exploitation locally searches around the promising area discovered during the exploration phase.

Inspired by the prey activity of gray wolves, GWO has a similar mechanism including social hierarchy, encircling prey, hunting, and attacking prey. To mimic social hierarchy in gray wolves, the best, the second, and the third fittest solution is called alpha, beta, and delta respectively which guides optimization/hunting. Other solutions are assumed to be omega which follows with the alpha, beta, and delta. During encircling prey, gray wolves can update their position according to the prey and gradually approach the prey. In hunting, the best three solutions are saved as alpha, beta, and delta. Other search agents updates position based on the position of alpha, beta, and delta. In the search process of attacking prey (exploitation) and searching for prey (exploration), a coefficient vector A allows the search agents to attack/diverge from the prey. When $|A| < 1$, the search agent tends to attack the prey; when $|A| > 1$, the search agent tends to diverge from the prey.

2.5 Classification

The classification module was constructed with a fully-connected layer having the activation function of ReLU, followed by the softmax function layer for the final breast cancer subtype classification. The model was optimized based on the cross-entropy loss using the Adam optimizer [24]. Batch normalization was applied with a batch size of 64. The learning rate and the training epoch were set as 1e-3 and 100, respectively. Our proposed model was built by Keras library (V 2.7.0), and all the experiments were performed on the CPU server of Intel(R) Xeon(R) Gold 5218 CPU @ 2.30GHz with the main memory of 394GB.

3 Experimental setting

3.1 Hyperparameter tuning

We performed the experiments to investigate the different structures of the classification module and optimized the hyperparameters including the depth of the hidden layer and the number of hidden nodes based on the weighted F1-score. From the results shown in Table 2, most of the structures showed similar performance, where the 3 hidden layer-based structure with 500-250-125 nodes obtained the best score. However, due to the computational limitation of our experimental setting, we chose a neural network architecture of 1 hidden layer with 125 nodes, which showed comparable performance with much less running time. This hyperparameter tuning experiment was performed based on the default Glorot initialization and based on the selected neural network structure, we added the optimization algorithm for BP, where all the experiments were performed based on this architecture.

Table 2: Average F1-score under different structures of the neural network based on 10-fold cross validation

	3 layers			2 layers			1 layer		
Number of nodes	750-500-250	500-250-125	250-125-64	750-500	500-250	250-125	500	250	125
F1-score	0.865	0.875	0.864	0.868	0.869	0.873	0.873	0.872	0.873

3.2 Target function setting of the BP-optimization

The target function of the optimization algorithms was to minimize the cross-entropy value, and before training the classification module based on the backpropagation, the optimization algorithm was iterated 50 times. After that, based on these initialized weights and biases, the model was trained for 100 epochs with an Adam optimizer.

4 Results

4.1 Performance evaluation based on different optimization

First, we conducted experiments to compare the weighted F1-score and accuracy of 6 optimization schemes, embedded within the above-mentioned optimized neural network model. We also incorporated random initialization and Glorot [25] for comparison. Glorot (also known as Xavier), which is the default initialization method for Keras, is to initialize each weight with a small Gaussian value with mean = 0.0 and variance based on the fan-in and fan-out of the weight. For example, each weight that connects an input node to a hidden node has fan-in of the number of input nodes and fan-out of the number of hidden nodes. From the results (Figure 1, Glorot and ALO showed the best average F1-score of 0.867 and 0.853, respectively, followed by GWO and Random initialization. Initial weights generated by MFO, WOA, and SCA can achieve near 0.8. FA showed the worst performance. From the results, we could also observe that ALO, Glorot, and GWO exhibited the lowest variances, followed by MFO and WOA. SCA and FA showed the highest instabilities.

In the aspect of the accuracy, the results are slightly different from F1-score, that GWO, MFO, and WOA demonstrate the best performance and the lowest variances. We assumed that GWO, MFO, and WOA have the most fine-grained and certain optimization steps to follow than others, making them extremely stable. The second tier consists of Glorot, ALO, and Random initialization. Although SCA achieved better performance compared to FA, it also experienced the highest variance among all optimization schemes.

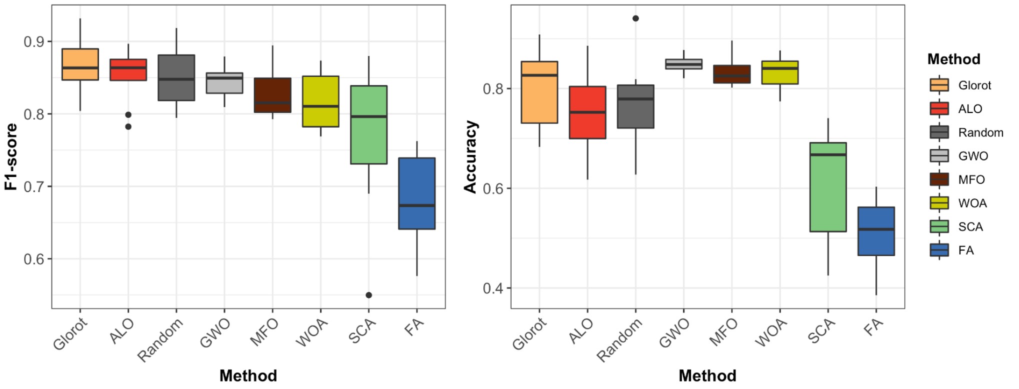

Figure 1: Performance comparison of BRCA subtype classification models with the different optimization algorithms based on 10-fold cross validation

4.2 Transition of loss

We monitored the cross entropy loss of the 6 initialization schemes with respect to running epochs in figure 2. We could observe that SCA and FA have a higher loss at first, and SCA converged smoothly. This could be due to the log-like mathematics implemented in SCA, while FA showed more fluctuation, representing more randomness it preserves. For GWO, MFO, WOA, and ALO, all of which demonstrated a low loss in the first place, and converged to near-zero (become flat) quickly. Among all, ALO provided very competitive results and outperformed other algorithms. Therefore, it may be concluded from the comparative results that the ALO is more suitable as an optimizer for breast cancer subtype classifications, that only requires only a few epochs to obtain optimal initial weights.

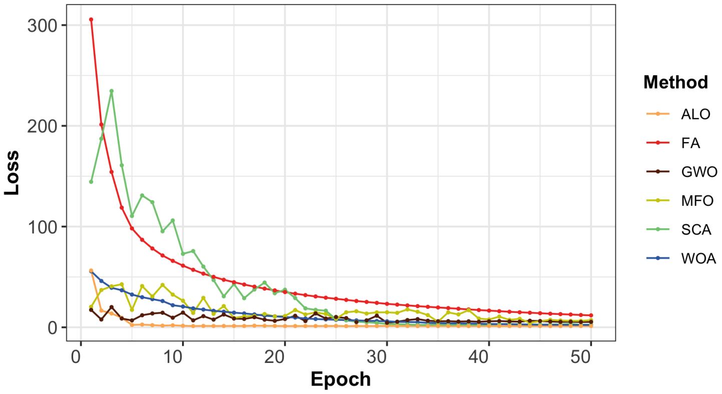

Figure 2: The change curve of network loss with initialization epoch in different optimization algorithms

4.3 Running time comparison

In order to compare the efficiency of the optimization algorithms, we measured the running time to search for the optimal value, which is one of the key issues. For each fold cross-validation, the number of iterations for each optimization algorithm was 50 and the number of searching particles was set to 10. From Table 3, the sine cosine algorithm showed the lowest running time, with only 140.6 seconds on average. All of the algorithms could finish running in 1000 seconds.

Table 3: Average running time of each optimization algorithm for parameter initialization conducting 10-fold cross-validation

	WOA	SCA	ALO	FA	MFO	GWO
Avg running time (s)	316.5	140.6	567.8	329	801.8	861.7

4.4 Performance comparison with traditional machine learning algorithms

We compared the models showing the top four performances with some traditional machine learning algorithms, including logistic regression (LR), support vector machine (SVM), random forest (RF), naïve Bayes (NB), and decision tree (DT). From the results (Table4), we could see that neural network-based models could outperform the classical machine learning methods for breast cancer subtype classification. The GWO-based model achieved the highest accuracy, and the Glorot-based model showed the best F1-score.

Table 4: Average F1-score and accuracy results under different classifiers for breast cancer subtypes performing 10-fold cross-validation

	Glorot	GWO	ALO	Random	LR	SVM	RF	NB	DT
Accuracy	0.804	**0.849**	0.750	0.769	0.836	0.840	0.854	0.721	0.772
F1-score	**0.867**	0.844	0.853	0.851	0.859	0.820	0.839	0.732	0.772

5 Conclusion

In this paper, we presented the optimized BP neural network models for breast cancer subtype prediction using gene expression profiles. Six optimization algorithms were implemented for weight initialization and we compared their performances to investigate their potential for prediction improvement. Compared to the other tasks such as image classification problems, such kind of heuristic optimization did not show much improvement in the breast cancer subtype classification. As a future work, we will focus on developing an algorithm to provide proper initialization for neural networks utilizing biological datasets such as gene expression profiles and perform breast cancer subtype classification as a case study.

References

[1] Nasrindokht Azamjah, Yasaman Soltan-Zadeh, and Farid Zayeri. Global trend of breast cancer mortality rate: a 25-year study. *Asian Pacific journal of cancer prevention: APJCP*, 20(7):2015, 2019.

[2] Mariona Baliu-Piqué, Atanasio Pandiella, and Alberto Ocana. Breast cancer heterogeneity and response to novel therapeutics. *Cancers*, 12(11):3271, 2020.

[3] Therese Sørlie, Robert Tibshirani, Joel Parker, Trevor Hastie, James Stephen Marron, Andrew Nobel, Shibing Deng, Hilde Johnsen, Robert Pesich, Stephanie Geisler, et al. Repeated observation of breast tumor subtypes in independent gene expression data sets. *Proceedings of the national academy of sciences*, 100(14):8418–8423, 2003.

[4] Joel S Parker, Michael Mullins, Maggie CU Cheang, Samuel Leung, David Voduc, Tammi Vickery, Sherri Davies, Christiane Fauron, Xiaping He, Zhiyuan Hu, et al. Supervised risk predictor of breast cancer based on intrinsic subtypes. *Journal of clinical oncology*, 27(8):1160, 2009.

[5] Ozlem Yersal and Sabri Barutca. Biological subtypes of breast cancer: Prognostic and therapeutic implications. *World journal of clinical oncology*, 5(3):412, 2014.

[6] Zhezhou Yu, Zhuo Wang, Xiangchun Yu, and Zhe Zhang. Rna-seq-based breast cancer subtypes classification using machine learning approaches. *Computational intelligence and neuroscience*, 2020, 2020.

[7] Feng Gao, Wei Wang, Miaomiao Tan, Lina Zhu, Yuchen Zhang, Evelyn Fessler, Louis Vermeulen, and Xin Wang. Deepcc: a novel deep learning-based framework for cancer molecular subtype classification. *Oncogenesis*, 8(9):1–12, 2019.

[8] Adham Beykikhoshk, Thomas P Quinn, Samuel C Lee, Truyen Tran, and Svetha Venkatesh. Deeptriage: interpretable and individualised biomarker scores using attention mechanism for the classification of breast cancer sub-types. *BMC medical genomics*, 13(3):1–10, 2020.

[9] Sungmin Rhee, Seokjun Seo, and Sun Kim. Hybrid approach of relation network and localized graph convolutional filtering for breast cancer subtype classification. In *Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence, IJCAI-18*, pages 3527–3534. International Joint Conferences on Artificial Intelligence Organization, 7 2018.

[10] Sangseon Lee, Sangsoo Lim, Taeheon Lee, Inyoung Sung, and Sun Kim. Cancer subtype classification and modeling by pathway attention and propagation. *Bioinformatics*, 36(12):3818–3824, 2020.

[11] Salman Mohaghegi, Yamille del Valle, Ganesh K Venayagamoorthy, and Ronald G Harley. A comparison of pso and backpropagation for training rbf neural networks for identification of a power system with statcom. In *Proceedings 2005 IEEE Swarm Intelligence Symposium, 2005. SIS 2005.*, pages 381–384. IEEE, 2005.

[12] Vishnu Pratap Singh Kirar. Improving the performance of back-propagation training algorithm by using ann. *World Academy of Science, Engineering, and Technology, International Journal of Computer, Electrical, Automation, Control and Information Engineering*, 9(1):187–192, 2015.

[13] Haoqiu Song, Zhiwei Ye, Chunzhi Wang, and Lingyu Yan. Image classification based on bp neural network and sine cosine algorithm. In *2019 10th IEEE international conference on intelligent data acquisition and advanced computing systems: technology and applications (IDAACS)*, volume 1, pages 562–566. IEEE, 2019.

[14] Nebojsa Bacanin, Miodrag Zivkovic, Fadi Al-Turjman, K Venkatachalam, Pavel Trojovskỳ, Ivana Strumberger, and Timea Bezdan. Hybridized sine cosine algorithm with convolutional neural networks dropout regularization application. *Scientific Reports*, 12(1):1–20, 2022.

[15] Katarzyna Tomczak, Patrycja Czerwińska, and Maciej Wiznerowicz. The cancer genome atlas (tcga): an immeasurable source of knowledge. *Contemporary oncology*, 19(1A):A68, 2015.

[16] Wei Zheng, Lisa M Chung, and Hongyu Zhao. Bias detection and correction in rna-sequencing data. *BMC bioinformatics*, 12(1):1–14, 2011.

[17] Michael I Love, Wolfgang Huber, and Simon Anders. Moderated estimation of fold change and dispersion for rna-seq data with deseq2. *Genome biology*, 15(12):1–21, 2014.

[18] Seyedali Mirjalili. Sca: a sine cosine algorithm for solving optimization problems. *Knowledge-based systems*, 96:120–133, 2016.

[19] Seyedali Mirjalili and Andrew Lewis. The whale optimization algorithm. *Advances in engineering software*, 95:51–67, 2016.

[20] Seyedali Mirjalili. The ant lion optimizer. *Advances in engineering software*, 83:80–98, 2015.

[21] Xin-She Yang. Firefly algorithm, stochastic test functions and design optimisation. *International journal of bio-inspired computation*, 2(2):78–84, 2010.

[22] Seyedali Mirjalili. Moth-flame optimization algorithm: A novel nature-inspired heuristic paradigm. *Knowledge-based systems*, 89:228–249, 2015.

[23] Seyedali Mirjalili, Seyed Mohammad Mirjalili, and Andrew Lewis. Grey wolf optimizer. *Advances in engineering software*, 69:46–61, 2014.

[24] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[25] Xavier Glorot and Yoshua Bengio. Understanding the difficulty of training deep feedforward neural networks. In *Proceedings of the thirteenth international conference on artificial intelligence and statistics*, pages 249–256. JMLR Workshop and Conference Proceedings, 2010.