UNCCER: Unified Network for Cancer Classification and Efficient Representation using Microarray Data

Haseeb Younis*†[a‡], Horiya Imane Brahmi†[b], Jonathan Byrne†[c], Rosane Minghim*[d],
* Department of Computer Science, University College Cork, Ireland
† Machine Learning Group, Intel Ireland Limited, Ireland
[a] haseeb.younis@cs.ucc.ie, haseeb.younis@intel.com, [b] imane.horiya.brahmi@intel.com,
[c] jonathan.byrne@intel.com, [d] r.minghim@cs.ucc.ie

***Abstract*—In recent years, there has been a significant advance in the use of machine learning (ML) techniques to extract gene expression data from microarray databases, particularly in cancer-related research. There no unified method for classifying cancer microarray data, even after ML adoption. Due to the high dimensionality of microarray data, it is difficult to extract the relevant features and provide insights that can be helpful in identifying cancer types and stages. In this paper, we propose a Unified Network for Cancer Classification and Efficient Representation (UNCCER) using Deep Learning (DL) on cancer microarray data. To implement this methodology, we employed a microarray database (CuMiDa) that has 78 carefully curated datasets for different types of cancers. Our single model has the capability to learn the patterns, cluster instances into their corresponding classes, and classify the cancer. We also used the Uniform Manifold Approximation and Projection (UMAP) to visualise, in low dimension, the instance separation both on original data and transformed data by our methodology. Using the proposed methodology, we achieved average 94% average accuracy, precision, recall, F1 Score, and 91% G-Mean.**

***Index Terms*—Cancer Microarray, Cancer Classification, Cancer Data Visualization, Deep Learning**

I. Introduction

Cancer is a major health hazard that caused 10 million deaths worldwide in 2020, according to the World Health Organization (WHO) [1]. It has been described as an association of cells that emerge from certain parts of the human body, which typically rapidly move to distant metastatic places [2].

New gene-based biomarkers are being used in the fast-developing field of molecular diagnostics to provide insights into disease causes. Until recently, the diagnosis and prognosis evaluation of tissues and tumours was primarily based on indirect markers that allowed for only broad categorizations into histologic or morphologic categories and ignored changes in the expression of specific genes. With the use of microarrays, global expression analysis currently provides an unprecedented opportunity to discover molecular markers of the state of activity of sick cells and patient samples, enabling the simultaneous interrogation of thousands of DNA molecules using a high-throughput method. Microarray analysis results in better early diagnosis and novel therapeutic methods for cancer. It also offers priceless information on disease pathophysiology, development, resistance to therapy, and reactions to cellular microenvironments [3], [4].

Using DNA microarray technology, researchers can observe an organism's level of gene expression. By comparing the expression levels of the individual genes in abnormal and healthy tissues, relevant genes to the disease can be observed. With the advent of microarray-related tools, researchers can compute the differentially expressed amounts of each gene in malignant and non-cancerous samples using statistical and mathematical techniques. These tools make it possible to identify the top genes that are differentially expressed and may be connected to a specific disease. More crucially, based on recorded empirical work previously performed with these genes, pathways and gene ontology enrichment analyses can be utilized to uncover possible biological processes involved in the disease.

In technical terms, microarrays measure the fluorescence intensity of fluorescently-labelled cDNA molecules, and the fluorescence intensity is a measure of the corresponding levels of gene expression. Slides, known as DNA Microarrays, which are pre-spotted with hundreds of probes complementary in sequence to the fluorescently-marked cDNA molecules are added to the array (commonly referred to as DNA/Gene chips). These slides are necessary for microarrays to monitor the expression of thousands of genes simultaneously. The ability to ascribe distinct patterns of gene expression to individual genes is made possible by the known positions of the probes on the chip [5].

Several traditional gene expression methods are used to extract a list of differentially expressed genes (DEG) from expression datasets [6]. However, these lists may contain hundreds or thousands of DEG, all of which are physiologically significant but from which no discernible pattern can be found. In this regard, ML methods may offer fast and precise expression pattern recognition. Using linked gene expression and mutation data, ML techniques are potent tools frequently used to create cancer prediction models. These models use

This publication has emanated from research supported in part by a grant from Science Foundation Ireland under Grant number 18/CRT/6222.
‡ Primary and Corresponding Author

gene expression data for tumour diagnosis and stratification, as well as prognosis and survival prediction for various kinds of cancer (lung, breast, ovarian, kidney, oral, liver, prostate, central nervous system, and gallbladder).

Khalsan et al. [7] outline a summary of biomarker research related to these cancer types. Numerous facets of ML-related cancer research are covered in the survey, such as RNA-Seq, microarray, and cancer classification as well as cancer prediction and discovery of biomarker genes. The work mentions technical problems with the available cancer prediction models and the accompanying measurement instruments for differentiating between non-cancerous and malignant tissues in terms of gene expression activity levels. Furthermore, detecting potential biomarker gene expression patterns can help forecast cancer risk in the future and guide the delivery of individualised care [7].

Several studies have used ML techniques to classify cancer using the Microarray datasets. Grisci et al. [8] proposed an algorithm, FS-NEAT, which used a method called Neuroevolution. This method is a branch of ML that blends neural networks and evolutionary computation. It works by concurrently classifying microarray data and choosing a subset of more relevant genes. To accommodate the high dimensional data, new structural operators were added to the core algorithm. Furthermore, a strict filtering and preparation procedure was used to choose high-quality microarray datasets—13 datasets total—from three distinct cancer types for the suggested technique. The results demonstrate that Neuroevolution was successful in classifying microarray samples and in identifying gene subsets that convey biologically significant information and can be adapted for use in other algorithms. 177 genes were found using this method; 82 were confirmed to be already linked to the specific cancer kinds, and 44 were linked to other cancer types, making them prospective candidates for further investigation as cancer biomarkers. Additionally, five long non-coding RNAs were found, of which four still lack known roles. The extracellular matrix, exosomes, and cell proliferation are inherently linked to the discovered expression patterns. They provided a Microarray database CuMiDa for different kinds of cancers and used multiple ML algorithms that include ZEROR, Support Vector Machine (SVM), Multi-Layer Perceptron (MLP), Decision Tree (DT), Naive Bayes (NB), Random Forest (RF), Hierarchical Clustering (HC), K-nearest Neighbour (KNN), and K-means [9]. The limitation of this work is that they could not achieve good accuracy and could not find a specific algorithm that work best on all the datasets.

Major issues with microarray data are the dimensionality of data and a low number of samples, leading to the "curse of dimensionality". When such a situation arises, models cannot learn the patterns and perform well. To overcome this problem, Bohere et al. [10] performed a wide range of experiments using different ML approaches as mentioned above. Similarly, Barbieri et al. [11] performed extensive experiments using multiple algorithms and ensemble approaches but could not agree on an algorithm that fits all the relevant datasets. These studies enhanced the performance of classification. However, their methodology was computationally expensive; and the results needed to improve. Additionally, separate models were used for different types of cancer, making it hard for future users to know what to use in their case. To enhance the results for the CuMiDa, these studies performed several ML approaches such as features extraction and selection [12], dimensionality reduction and statistical analysis [13], [14].

As Microarray datasets have high dimensionality, we can take advantage of DL to find the pattern that can lead us to identify the disease with more confidence and reliability [13], [15]. In this paper, we propose a methodology that uses a DL-based algorithm that automatically identifies patterns from Microarray datasets and classifies cancer. The proposed algorithm UNCCER is lightweight and independent of of uncertain preprocessing steps, such as feature extraction and selection. This single model fits all the cancerous Microarray datasets provided in CuMiDa. Using UNCCER, we achieve more than 94% average accuracy, precision, recall, and F1 score, as well as 91% of G-Mean. We employ a data visualization method to understand sample segregation by projecting the both the original data and the embeddings provided by our model, showing the robustness of our methodology and offering a tool to understand feature-based sample discrimination. The technical details of our methodology are given in Section II, and the analysis is given in Section III.

II. Methodology

To implement the UNCCER, we used the CuMiDa, a collection of datasets provided by Grisci et al. [8]. It includes 78 datasets of different cancer types that have unbalanced classes. To balance the dataset, we employed the Synthetic Minority Over-Sampling Technique (SMOTE) [16] on the training dataset. After that, we trained a DL model and visualized the data distribution using Uniformed Manifold Approximation and Projection (UMAP) [17]. Details of our methodology are outlined below.

A. Dataset

To obtain the dataset, CuMiDa curators employed GEO [18] to download multiple microarray datasets for various cancer subtypes, including colorectal, gastric, pancreatic, liver, bladder, lung, throat, renal, brain, prostate, ovary, leukaemia and breast cancers. The datasets were selected based on criteria such as the non-usage of chemotherapies, gene therapies, interfering molecules, knockdown cultures, induced mutations, clear protocol descriptions, xenograft technique, raw data availability, and a six-sample minimum per condition. After a manual curated process, 78 microarray datasets were handpicked, including single and dual-channel samples. After collection, the datasets were normalized and verified using the R package arrayQualityMetrics [19] in order to establish the quality of the samples. Samples were eliminated if they showed poor quality in at least half of the parameters that arrayQualityMetrics measured. After that, each final normalized matrix was carefully selected to exclude undesired probes

979-8-3315-0869-2/24/$31.00 ©2024 IEEE

with no connection to the nucleic acid sequences. Out of the 78 datasets, 73 were kept due to missing class information.

B. Data Preprocessing

We removed the data that was missing class information and used SMOTE to generate synthetic samples for the minority class to balance the dataset. The mathematical representation of our data is given in the equations 1 and 2.

Mathematical Representation:

In the first step, determine the number of samples in the minority class, N_{min}. Then, set the number of neighbors as:

$$k = \min(5, N_{\text{min}} - 1) \tag{1}$$

After that, generate Synthetic Samples for each minority class, sample x_i, as follows:

- Find k nearest neighbors.
- Randomly select a neighbor, $x_{i,k}$.
- Create a synthetic sample:

$$x_{\text{new}} = x_i + \lambda \cdot (x_{i,k} - x_i) \tag{2}$$

where $\lambda \sim \mathcal{U}(0, 1)$

Where:

- N_{min}: Number of samples in the minority class.
- k: Number of nearest neighbors.
- x_i: Minority class sample.
- $x_{i,k}$: k-th nearest neighbor of x_i.
- x_{new}: Synthetic sample.
- λ: Random number drawn from a uniform distribution, $\mathcal{U}(0, 1)$

The synthetic data was generated on the training set; the validation set was not included in the data generation to avoid bias.

C. Model Architecture

The model architecture consists of the following layers:

1) **Input Layer**:
   Input Shape (number_of_features, 1)
2) **Convolutional Layer (Conv1D):**
   Filters: 64
   Kernel Size: 3
   Activation: ReLU

   The convolutional operation can be represented as:

$$y_{i,k} = \text{ReLU}\left(\sum_{j=0}^{2} x_{i+j,c} \cdot w_{j,k} + b_k\right) \tag{3}$$

   Where:
   - $x_{i+j,c}$ is the input at position $i + j$ for channel c.
   - $w_{j,k}$ are the weights of the k-th filter.
   - b_k is the bias for the k-th filter.
   - $y_{i,k}$ is the output of the convolution at position i for the k-th filter.
3) **Max Pooling Layer (MaxPooling1D)**:
   Pool Size: 2
   The max pooling operation can be represented as:

$$y_{i,k} = \max(x_{2i,k}, x_{2i+1,k}) \tag{4}$$

   Where:
   - $x_{2i,k}$ and $x_{2i+1,k}$ are the input values in the pool.
   - $y_{i,k}$ is the output of the pooling operation.
4) **Dropout Layer (Dropout)**:
   Rate: 0.5
   The dropout operation can be represented as:

$$y_{i,k} = \begin{cases} 0 & \text{with probability } 0.5 \\ \frac{x_{i,k}}{0.5} & \text{with probability } 0.5 \end{cases} \tag{5}$$

   Where:
   - $x_{i,k}$ is the input.
   - $y_{i,k}$ is the output after dropout.
5) **Flatten Layer (Flatten)**:
   This layer reshapes the input to a 1D array. If the input shape is (n, m), the output shape will be $(n \cdot m)$.
6) **Dense Layer (Fully Connected Layer)**:
   Units: 128
   Activation: ReLU
   The dense layer operation can be represented as:

$$y_i = \text{ReLU}\left(\sum_{j=0}^{N-1} x_j \cdot w_{j,i} + b_i\right) \tag{6}$$

   Where:
   - x_j is the input from the previous layer.
   - $w_{j,i}$ are the weights.
   - b_i is the bias.
   - y_i is the output.
7) **Output Dense Layer (Fully Connected Layer)**:
   Units: 1
   Activation: Sigmoid
   The output layer in case of binary classification can be represented as:

$$y = \sigma\left(\sum_{j=0}^{127} x_j \cdot w_j + b\right) \tag{7}$$

   The output layer in case of multiclass classification can be represented as:
   Units: *num_classes*, Activation: Softmax

$$y_i = \frac{\exp(z_i)}{\sum_{j=1}^{num\_classes} \exp(z_j)}, \quad \text{for } i = 1, \ldots, num\_classes \tag{8}$$

   Where:

$$z_i = \sum_{j=0}^{127} x_j \cdot w_{j,i} + b_i \tag{9}$$

   and
   - $\sigma(z) = \frac{1}{1+e^{-z}}$ is the sigmoid activation function.

979-8-3315-0869-2/24/$31.00 ©2024 IEEE

- x_j is the input from the previous layer.
- w_j are the weights.
- b is the bias.
- y is the output, representing the probability of the positive class.

Full Model Summary in Mathematical Notation:

a) Convolution: $\mathbf{Y} = \text{ReLU}(\mathbf{X} * \mathbf{W} + \mathbf{b})$
b) Max Pooling: $\mathbf{Y}_{i,k} = \max(\mathbf{X}_{2i,k}, \mathbf{X}_{2i+1,k})$
c) Dropout: $\mathbf{Y}_{i,k} = \begin{cases} 0 & \text{with probability } 0.5 \\ \frac{\mathbf{X}_{i,k}}{0.5} & \text{with probability } 0.5 \end{cases}$
d) Flatten: $\mathbf{Y} = \text{flatten}(\mathbf{X})$
e) Dense (ReLU): $\mathbf{Y} = \text{ReLU}(\mathbf{X}\mathbf{W} + \mathbf{b})$
f) Output (Sigmoid): $y = \sigma(\mathbf{X}\mathbf{W} + b)$ in case of binary classification
g) Output (Softmax): $y_i = \frac{\exp(z_i)}{\sum_{j=1}^{num\_classes} \exp(z_j)}$ in case of multiclass classification

Hyperparameters and Early Stopping Criteria:
The Adam optimizer with 0.001 learning rate, cross-entropy as loss, accuracy as evaluation metrics, and batch size of 32 were used as the model hyperparameters. A customized early stopping criteria was also designed that will stop the model training once it has reached 100% validation accuracy with the patience of 10. In the case it does not reach 100% accuracy, it will return the best weights at the validation point without storing it on a drive, which is cost-effective.

D. Visual Analysis of the Datasets

We used UMAP to visualize the distribution of data in two dimensions to understand the segregation of the different labels, showing the robustness of our methodology. The mathematical representation of this UMAP is given below in equations 10 to 17.

Mathematical Representation

1. Fuzzy Topological Representation

First of all, compute the pairwise distance between all the samples using the distance matrix:

$$d(x_i, x_j) \tag{10}$$

Then, compute the local connectivity between each sample using Local Fuzzy Simplicial Set using the following equations:

$$\rho_i = \min\{d(x_i, x_j) \mid |\{x_j \mid d(x_i, x_j) > 0\}| \geq k\} \tag{11}$$

$$\sigma_i = \log k / \left(\sum_{j \neq i} \exp(-(d(x_i, x_j) - \rho_i)) \right) \tag{12}$$

$$\mu_{ij} = \exp\left(\frac{-(d(x_i, x_j) - \rho_i)}{\sigma_i} \right) \tag{13}$$

$$\mu_{ij} = \mu_{ij} + \mu_{ji} - \mu_{ij} \cdot \mu_{ji} \tag{14}$$

2. Low-Dimensional Representation

After that, define cross-entropy between the high dimensional and low dimensional fuzzy set, calculate the cost and minimize it using gradient descent as follows.

$$C = \sum_{i \neq j} \left(\mu_{ij} \log \frac{\mu_{ij}}{\nu_{ij}} + (1 - \mu_{ij}) \log \frac{1 - \mu_{ij}}{1 - \nu_{ij}} \right) \tag{15}$$

$$\nu_{ij} = \left(1 + a \cdot d(y_i, y_j)^{2b}\right)^{-1} \tag{16}$$

$$\mathbf{Y} = \arg\min_{\mathbf{Y}} C \tag{17}$$

Where:

- $\mathbf{X} \in \mathbb{R}^{n \times d}$: original dataset.
- $d(x_i, x_j)$: pairwise distance.
- ρ_i: local connectivity distance.
- σ_i: local distance scale.
- μ_{ij}: fuzzy set membership strength.
- ν_{ij}: low-dimensional fuzzy set membership strength.
- $\mathbf{Y} \in \mathbb{R}^{n \times 2}$: low-dimensional representation.
- C: cost function.
- a, b: hyperparameters.

E. Evaluation Measures

We used accuracy, precision, recall, and F1 score as the standard evaluation measures for the classification task. However, the geometric mean (G-Mean) was used to measure the performance of our methodology as we have imbalanced datasets and it was proposed to measure the performance in case of imbalanced datasets [20].

III. Results and Discussion

This section deals with the comparative analysis of results, our findings, and the validity of our methodology.

A. Dataset Analysis

The CuMiDa dataset originally 78 microarray datasets of different cancer types, categorised as 2 bladder, 2 brain, 13 breast, 10 colorectal, 3 gastric, 9 leukaemia, 10 liver, 6 lung, 4 ovary, 1 pancreatic, 10 prostate, 4 renal, and 4 throat. These datasets range from 2 classes, such as cancer vs normal, to 7 classes, including different stages and types of cancer, and were unbalanced. We removed Gastric_GSE22804, Liver_GSE62043, Prostate_GSE8511, Prostate_GSE38241, and Prostate_GSE60329 from the experiments due to non-availability of class information, thus reaching 73 datasets. These datasets have a small number of instances and a high number of features that potentially lead to model underfitting. To overcome this problem, we used the SMOTE method to generate synthetic instances; this method was only applied to the training data after doing an 80:20 train-test split with the stratify method. This method ensures the right distribution of classes in the train and test set. The problem with generating the synthetic data with such a small dataset is the failure to generate data for a minority class that has a smaller number of instances. We resolved this problem by setting the k-neighbours less than or equal to the number of instances of the minority class.

B. Results and Comparison

Our methodology performed very well on all the datasets, showing that our model is capable of handling binary as well as multiclass datasets. The results for Breast_GSE2280 with 2 classes and Leukemia_GSE28497 with 7 seven classes are shown in Figures 1 and 2, and point distribution for these are shown in Figures 3 and 4, respectively.

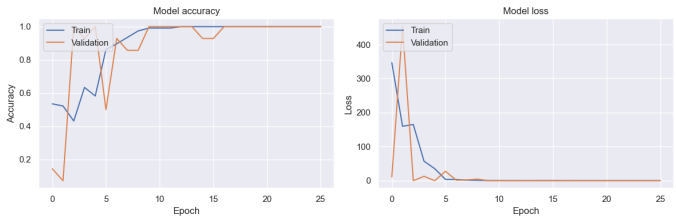

Fig. 1. Results for Breast_GSE22820 with 2 classes

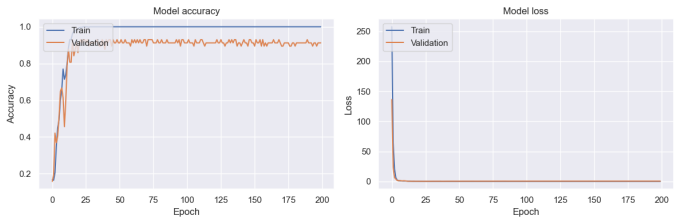

Fig. 2. Results for Breast_GSE28497 with 7 classes

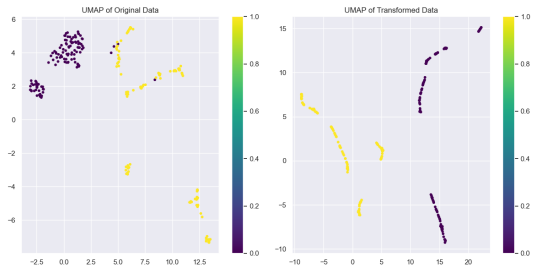

Fig. 3. Data Point Distribution of Original Dataset and Separated by our Model for Breast_GSE22820

The figures above show the results of our methodology as well as how well it can separate the data to classify, even with different numbers of categories. We also compared our results with [8], and our methodology outperformed their methodology, as shown in Figure 5.

C. Robustness of our Methodology

The proposed methodology achieved very promising results over almost all of the datasets shown in Image 5. Other comparative studies used extensive preprocessing methods and multiple models to classify cancer types and stages; however, our methodology used a single model to fit all of those datasets. UNCCER is lightweight and computationally less expensive than previous methods. The model was trained

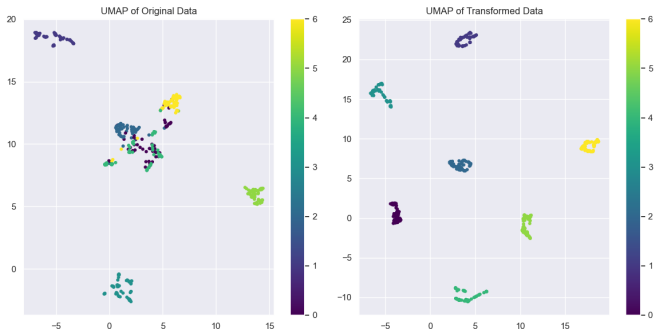

Fig. 4. Data Point Distribution of Original Dataset and Separated by our Model for Breast_GSE28497

using 200 epochs with early stopping criteria, so it stopped at different numbers of epochs for different datasets. It stops as it learns the best pattern to classify the target cancer type and avoids model overfitting. Employing UMAP to show the ability of the model to segregate classes can be explicitly used to perform unsupervised weak labelling of microarray cancer datasets. The learning ability of this architecture shows that it can find a feature space for similar datasets without extensive preprocessing and be less sensitive to the reduced number of instances. It also can filter a large number of features by relevance.

IV. Conclusion

ML and biology experts are working hard to find algorithms and processes to shortlist descriptors of interest and to classify cancer. In this paper we have proposed UNCCER, a Deep learning and visualization metholodogy which can classify cancer with promising results using a single DL Model as well as offer observations in sample segregation by features. The model has the ability to work on binary or multiple classes and to separate clusters corresponding to classes. The architecture's learning capacity demonstrates that it can locate the feature space for other datasets that are connected to it without extensive preprocessing or being constrained by the number of instances. Additionally, it has the capacity to handle a large amount of features to and filter the important ones. In future, this model could be used to train and predict cancer types and stages using the microarray data, and serves as a pre-trained model to learn different patterns for different cancer modalities.

Code and Datasets Availability

The datasets used in this research are available at CuMiDa's original site and code is made available in our UNCCER repository.

Acknowledgment

This publication has emanated from research supported in part by a grant from Science Foundation Ireland under Grant number 18/CRT/6222. For the purpose of Open Access, the author has applied a CC BY public copyright license to

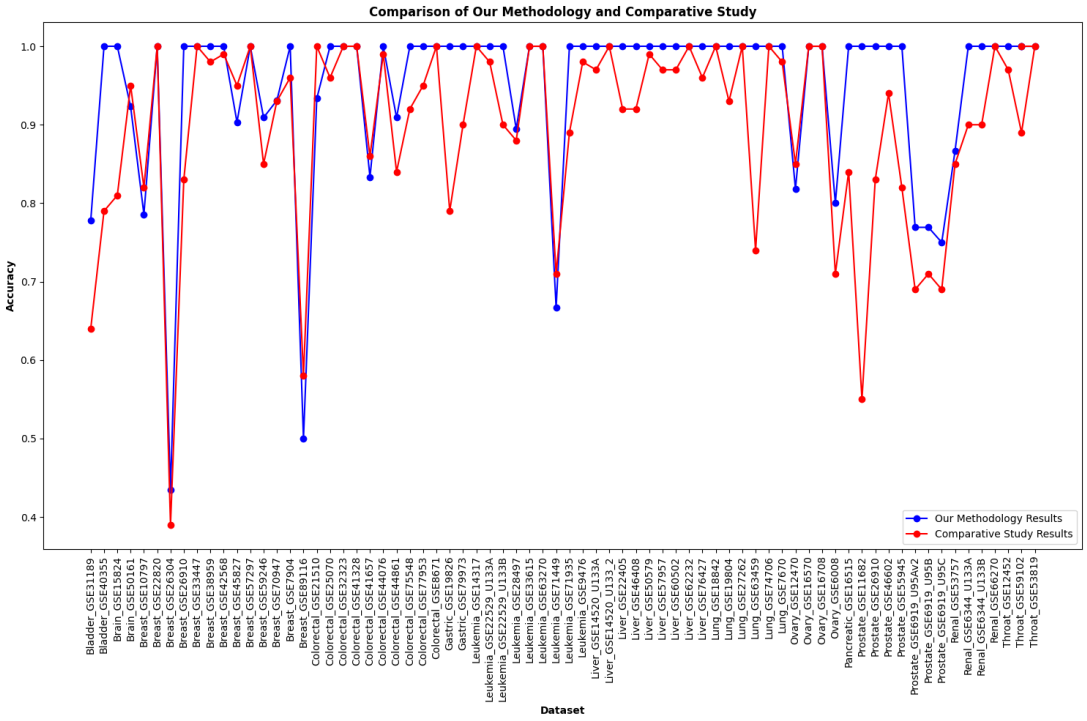

Fig. 5. Comparison of our Methodology with Base Study. Out of the 73, our model outperformed in 47, equalled in 19 and underperformed in 8 Datasets.

any Author Accepted Manuscript version arising from this submission.

REFERENCES

[1] W. H. O. (WHO), "Cancer," https://www.who.int/news-room/fact-sheets/detail/cancer, 2024, accessed: 2024-07-14.

[2] S. Shandilya and C. Chandankhede, "Survey on recent cancer classification systems for cancer diagnosis," in *2017 International Conference on Wireless Communications, Signal Processing and Networking (WiSPNET)*. IEEE, 2017, pp. 2590–2594.

[3] P. F. Macgregor and J. A. Squire, "Application of microarrays to the analysis of gene expression in cancer," *Clinical chemistry*, vol. 48, no. 8, pp. 1170–1177, 2002.

[4] E. Alhenawi, R. Al-Sayyed, A. Hudaib, and S. Mirjalili, "Feature selection methods on gene expression microarray data for cancer classification: A systematic review," *Computers in biology and medicine*, vol. 140, p. 105051, 2022.

[5] R. Govindarajan, J. Duraiyan, K. Kaliyappan, and M. Palanisamy, "Microarray and its applications," *Journal of Pharmacy and Bioallied Sciences*, vol. 4, no. Suppl 2, pp. S310–S312, 2012.

[6] L. Liu, S. Wang, C. Cen, S. Peng, Y. Chen, X. Li, N. Diao, Q. Li, L. Ma, and P. Han, "Identification of differentially expressed genes in pancreatic ductal adenocarcinoma and normal pancreatic tissues based on microarray datasets," *Molecular Medicine Reports*, vol. 20, no. 2, pp. 1901–1914, 2019.

[7] M. Khalsan, L. R. Machado, E. S. Al-Shamery, S. Ajit, K. Anthony, M. Mu, and M. O. Agyeman, "A survey of machine learning approaches applied to gene expression analysis for cancer prediction," *IEEE Access*, vol. 10, pp. 27 522–27 534, 2022.

[8] B. I. Grisci, B. C. Feltes, and M. Dorn, "Neuroevolution as a tool for microarray gene expression pattern identification in cancer research," *Journal of biomedical informatics*, vol. 89, pp. 122–133, 2019.

[9] F. Nelli, "Machine learning with scikit-learn," in *Python data analytics: with pandas, numPy, and matplotlib*. Springer, 2023, pp. 259–287.

[10] J. d. S. Bohrer and M. Dorn, "Enhancing classification with hybrid feature selection: A multi-objective genetic algorithm for high-dimensional data," *Expert Systems with Applications*, p. 124518, 2024.

[11] M. C. Barbieri, B. I. Grisci, and M. Dorn, "Analysis and comparison of feature selection methods towards performance and stability," *Expert Systems with Applications*, p. 123667, 2024.

[12] M. Ilyas, K. M. Aamir, A. Jaleel, and M. Deriche, "A novel leukemia gene features extraction and selection technique for robust type prediction using machine learning," *Arabian Journal for Science and Engineering*, pp. 1–19, 2024.

[13] B. I. Grisci, M. J. Krause, and M. Dorn, "Relevance aggregation for neural networks interpretability and knowledge discovery on tabular data," *Information sciences*, vol. 559, pp. 111–129, 2021.

[14] S. Osama, H. Shaban, and A. A. Ali, "Gene reduction and machine learning algorithms for cancer classification based on microarray gene expression data: A comprehensive review," *Expert Systems with Applications*, vol. 213, p. 118946, 2023.

[15] A. Khan and B. Lee, "Deepgene transformer: Transformer for the gene expression-based classification of cancer subtypes," *Expert Systems with Applications*, vol. 226, p. 120047, 2023.

[16] R. Pears, J. Finlay, and A. M. Connor, "Synthetic minority oversampling technique (smote) for predicting software build outcomes," *arXiv preprint arXiv:1407.2330*, 2014.

[17] B. Ghojogh, M. Crowley, F. Karray, and A. Ghodsi, "Uniform manifold approximation and projection (umap)," in *Elements of Dimensionality Reduction and Manifold Learning*. Springer, 2023, pp. 479–497.

[18] T. Barrett, T. O. Suzek, D. B. Troup, S. E. Wilhite, W.-C. Ngau, P. Ledoux, D. Rudnev, A. E. Lash, W. Fujibuchi, and R. Edgar, "Ncbi geo: mining millions of expression profiles—database and tools," *Nucleic acids research*, vol. 33, no. suppl_1, pp. D562–D566, 2005.

[19] A. Kauffmann, R. Gentleman, and W. Huber, "arrayqualitymetrics—a bioconductor package for quality assessment of microarray data," *Bioinformatics*, vol. 25, no. 3, pp. 415–416, 2009.

[20] Y. Sun, M. S. Kamel, A. K. Wong, and Y. Wang, "Cost-sensitive boosting for classification of imbalanced data," *Pattern recognition*, vol. 40, no. 12, pp. 3358–3378, 2007.