2021 43rd Annual International Conference of the
IEEE Engineering in Medicine & Biology Society (EMBC)
Oct 31 - Nov 4, 2021. Virtual Conference

Training with Small Medical Data: Robust Bayesian Neural Networks for Colon Cancer Overall Survival Prediction

Te-Cheng Hsu[1] and Che Lin[2]

Abstract— **Fast and accurate cancer prognosis stratification models are essential for treatment designs. Large labeled patient data can power advanced deep learning models to obtain precise predictions. However, since fully labeled patient data are hard to acquire in practical scenarios, deep models are prone to make non-robust predictions biased toward data partition and model hyper-parameter selection. Given a small training set, we applied the systems biology feature selector in our previous study to avoid over-fitting and select 18 prognostic biomarkers. Combined with three other clinical features, we trained Bayesian binary classifiers to predict the 5-year overall survival (OS) of colon cancer patients in this study. Results showed that Bayesian models could provide better and more robust predictions compared to their non-Bayesian counterparts. Specifically, in terms of the area under the receiver operating characteristic curve (AUC), macro F1-score (maF$_1$), and concordance index (CI), we found that the Bayesian bimodal neural network (late fusion) classifier (*B-Bimodal*) achieved the best results (AUC: 0.8083 ± 0.0736; maF$_1$: 0.7300 ± 0.0659; CI: 0.7238 ± 0.0440). The single modal Bayesian neural network classifier (*B-Concat*) fed with concatenated patient data (early fusion) achieved slightly worse but more robust performance in terms of AUC and CI (AUC: 0.7105 ± 0.0692; maF$_1$: 0.7156 ± 0.0690; CI: 0.6627 ± 0.0558). Such robustness is essential to training learning models with small medical data.**

I. Introduction

According to an estimate from the World Health Organization (WHO), colon cancer is one of the top-leading causes of cancer mortality worldwide. It is estimated to contribute up to 9% of new cases and deaths in 2020 [1]. Early and appropriate adjuvant chemotherapy was proved to be beneficial for prognostic conditions of colon cancer patients [2]. There is, therefore, an essential need for fast and accurate prognosis prediction models in the early stages of colon cancer. With this aim, several well-known biomarkers have been proposed and widely studied in predicting cancer prognosis [3]. However, due to the high dimensionality and small size nature of biological data, it is not easy to understand the underlying interactions directly from raw data. Our previous research utilized systems biology approaches to identify differentially enriched pathways. Prognostic biomarkers with biological insights were selected by constructing gene interaction networks (GINs) and calculating the prognostic relevance values (PRVs) from microarray data for non-small cell lung cancer (NSCLC) and breast cancer [4], [5]. We can select a set of prognostic biomarkers for colon cancer patients through the same systems biology feature selector and used deep learning models to predict their prognostic condition.

Although plenty of deep learning models have been proposed for binary classification tasks, many of them can result in over-fitting when trained with only small data. The predictions could be highly sensitive to model hyper-parameter selection and training/test sets data partition [6]. Recent works showed that predictions generated from well-tuned deep neural networks (DNNs) could vary significantly among different network initialization in patient-specific predictions [7]. Many works used ensemble learning to improve the overall prediction accuracy and capture model uncertainty [7]. However, by learning an ensemble of models, the model parameters increase significantly with the number of models used, affecting the computational complexity at the training time. Instead of learning an ensemble of point estimates, Bayesian deep learning (BDL) [8] views the parameters of the neural networks (NNs) as distributions and captures model uncertainty efficiently [7]. In BDL, we can obtain the distributions through variational inference [6]. Variational dropout can be further introduced in a full Bayesian analysis [9] to avoid over-fitting.

For integrating heterogeneous patient data, bimodal NN classifiers use distinct subnetworks to extract meaningful representation from each data modality (late fusion) and then produce the final prediction through the following merged network. Our previous research found that bimodal NN classifiers could provide better performance compared to various machine learning benchmarks [4]. Following [4], in this work, we introduce bimodal BDL to predict the 5-year OS of colon cancer patients and compare it with powerful benchmarks such as the support vector machine (SVM) and random forest (RF) classifiers. Bimodal and single concatenation NNs (early fusion), as well as their Bayesian counterparts, are proposed in this work. The main challenges of this work result from data scarcity. We want to build a powerful classifier with few data samples without over-fitting via BDL and sufficient regularization. Results showed that BDL-based models consistently provide more robust and better predictions than SVM and RF. The essential methods used in this work were summarized in Fig. 1.

[1] T.C.-Hsu is a Ph.D. student of Institute of Communication Engineering, National Tsing-Hua University (NTHU), 30013 Hsinchu, Taiwan andy810436@gmail.com

[2] C. Lin is an associate professor of Department of Electrical Engineering and Graduate Institute of Communication Engineering, National Taiwan University (NTU), Taipei 10617, Taiwan chelin@ntu.edu.tw

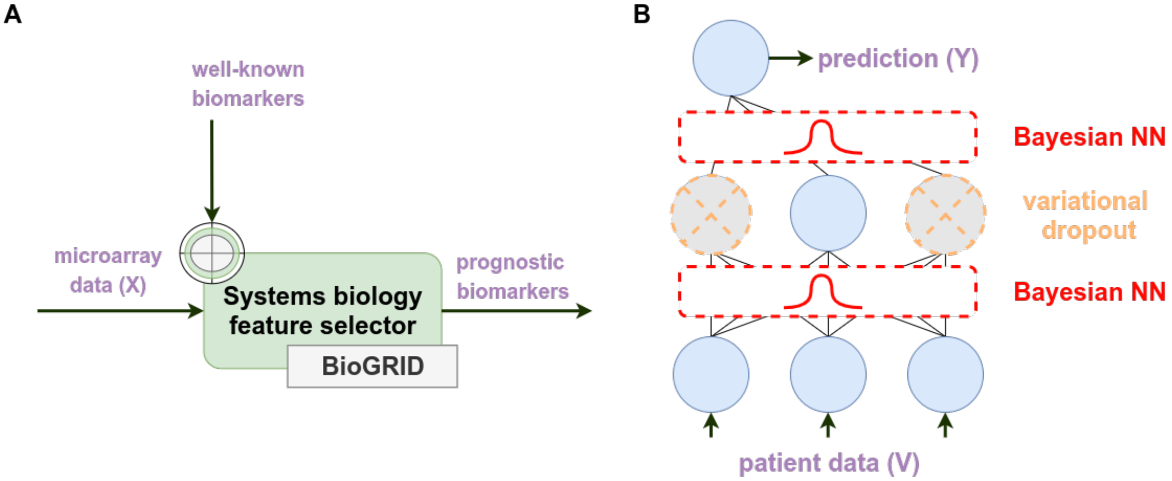


Fig. 1: Schematic illustration of essential techniques. (A) Systems biology feature selector. (B) Bayesian neural network.

II. Methodology

A. Systems biology feature selector

Since biological data are of high dimensionality and small data size, appropriate feature selection approaches should be first applied before training classifiers. We followed our systems biology feature selector in [4] to identify prognostic biomarkers as shown in Fig. 1(A). We started with BioGrid [10] as the candidate network. Multiple GINs were built on patient subsets split by high/low well-known biomarker expression levels. They were then trimmed with statistical model order selection techniques to remove false-positive connections. Prognosis relevance values (PRVs) were calculated from built networks to identify the final gene feature subset as the prognostic biomarkers for training classifiers [5], [4].

B. Bayesian Deep Learning (BDL)

We briefly introduce how we apply the BDL framework to our model (Fig. 1(B)). Suppose we have a probabilistic model p that captures the relationship between the observed and latent (hidden) variables. Let a $L-$layer neural network classifier parameterize p. The observed variables are patient data and label, and the latent variables are the parameters of the neural network. Denote the patient gene expression profiles and clinical information of N patients as $X = \{\mathbf{x}_i\}_{i=1}^N$ and $C = \{\mathbf{c}_i\}_{i=1}^N$, respectively. The prediction targets, $Y = \{\mathbf{y}_i\}_{i=1}^N$, are modeled as one-hot categorical distributions, indicating either poor ($\mathbf{y}_i = [1 \quad 0]^T$) or good ($\mathbf{y}_i = [0 \quad 1]^T$) prognosis class. The weights and biases of the neural network, $W = \{\mathbf{w}_i\}_{l=1}^L$, are modeled as Gaussian distributions. To simplify the notation, we collect X and C as $V = \{X, C\}$ in the following sections.

1) Variational Inference: In practice, it is hard to evaluate and directly represent $p(W|V,Y)$ in a tractable fashion. Following the mean-field approach [8] with a factorial distribution, q, such that

$$q(W) = q\left(\{\mathbf{w}_i\}_{l=1}^L\right) = \Pi_{i=1}^L q\left(\mathbf{w}_i\right), \tag{1}$$

we can derive the evidence lower bound (ELBO) of $p(Y|V)$ as

$$\begin{aligned}&\mathcal{L}(Y|V,W,q)\\ &\triangleq \log p(Y|V) - \mathbb{KL}\left(q(W)\|p(W|V,Y)\right) \\ &= \mathbb{E}_{q(W)}\left[\log p(Y|V,W)p(W) - \log q(W)\right].\end{aligned} \tag{2}$$

When $q(W) = p(W|V,Y)$, the KL divergence is zero so that the ELBO is equal to the true posterior. By maximizing the ELBO, we can optimize W through back-propagation.

2) Model Evaluation: At testing time, we make predictions with the sample mean of the M samples of the output one-hot categorical distribution $\mathbf{y}$:

$$\mathbf{y}_{pred} = \mathbb{E}_{p(Y|V)}\mathbf{y} \approx \frac{1}{M}\sum_{i=1}^{M}\mathbb{E}_{p(Y|V,W^i)}\mathbf{y}, \quad W^i \sim q(W), \tag{3}$$

where M ranges usually from 10 to 1000.

3) Variational Dropout: Classic dropout approaches require certain neurons to be masked out with a dropout probability, p_{drop}, at the training time. The weights are scaled by p_{drop} at the testing time [11]. On the other hand, in variational dropout [9], multiplicative Gaussian noises with mean 1 and learnable standard deviation α are introduced to the weights of NNs. It is reported that both approaches achieved similar performance in image classification tasks [12]. One major advantage of variational dropout is that it maintains identical means of the weights. No scaling at the testing time is needed, which allows us to perform full Bayesian analysis [9]. Besides L2-regularization, we further applied variational dropout to BDL-based models to avoid over-fitting in this work.

III. Dataset and Results

A. Dataset

We collected 12 GEO datasets (GSE9254, GSE11831, GSE12945, GSE13067, GSE13294, GSE13471, GSE14333, GSE17538, GSE18088, GSE18105, GSE20916, and GSE29623) of colon cancer patients from National Center for Biotechnology Information (NCBI) as a cohort. Our preliminary research collected 815 patients with full microarray data for systems biology feature selection. We further filter patients with full clinical and microarray data for bimodal learning and collected 50 patients in the training set and 152 patients in the test set. The rest 765 patients are without labels will not be utilized here. They could be exploited for semi-supervised learning models, which will be a topic for our future work. Note that we included more patients in the test set to ensure the performance is less biased by the small patient data size. We included 18 prognostic biomarkers (APP, CAND1, COPS5, CUL1, EED, EGFR, ELAVL1, GRB2, HDAC1, RPA2, EPCAM, CD44, ALCAM, PROM1, ABCB1, ABCC1, ABCG2, and ALDH1A1) and three clinical features (gender, tumor grade, and cancer stage), which are the only features available among all GEO datasets. We labeled patients with the OS event before 5 years as the poor prognosis class and patients who survived over five years as the good prognosis class.

TABLE I: The patient distribution of the cohort studied in this work.

Features		**Train**	(N = 50)	**Test**	(N = 152)
Prognosis	Good	21	(42.00%)	65	(42.76%)
	Poor	29	(58.00%)	87	(57.24%)
Gender	Female	32	(64.00%)	64	(42.11%)
	Male	18	(32.00%)	88	(57.89%)
Grade	I	2	(4.00%)	11	(7.24%)
	II	32	(64.00%)	120	(78.95%)
	III	16	(32.00%)	21	(13.82%)
Stage	I	4	(8.00%)	15	(9.87%)
	II	14	(28.00%)	35	(23.03%)
	III	16	(32.00%)	52	(34.24%)
	IV	16	(32.00%)	50	(32.89%)

B. Experimental setups

The following models were included into comparison:

- *SVM*: a support vector machine with radial basis kernel.
- *RF*: a random forest classifier.
- *Concat*: an NN classifier fed with V, a direct concatenation of X and C (early fusion).
- *Bimodal*: a bimodal NN classifier (late fusion) [5], [4].
- *B-Concat*: a Bayesian NN classifier fed with V (early fusion).
- *B-Bimodal*: a Bayesian bimodal NN classifier (late fusion).

We adopted NADAM [13] optimizer with default hyper-parameters for all neural network classifiers except for the learning rate, which was decided at the beginning of the training process. The model hyper-parameters were obtained through 4-fold cross-validation (4-CV). We used area under the receiver operating curve (AUC), unweighted F1-score (macro F1-score, maF_1) [14], concordance index (CI) [15], accuracy (ACC), Kaplan Meier analysis (KM-plot) [16], and log-rank test [17] as performance evaluation metrics. We retrieved the hard classification threshold by maximizing the Youden index [18] to calculate ACC and macro F_1. We adopted the same test set for all models to have fair comparisons. All models were implemented with Python: *SVM* and *RF* were built with scikit-learn packages [19], and BDL-based models were trained with Zhusuan [20] built with TensorFlow backend. The 95% confidence intervals for all metrics reported in Table II were obtained through the test set bootstrapping with 1000 bootstrap sets.

TABLE II: Performance summary: Half of the 95% confidence intervals were provided in the parentheses.

Models	**AUC** (%)	**maF$_1$** (%)	**CI** (%)	**ACC** (%)
SVM	66.74 (0.00)	60.69 (3.89)	61.46 (0.00)	57.24 (7.89)
RF	52.09 (1.92)	52.56 (3.89)	50.54 (1.85)	55.26 (7.89)
Concat	58.28 (9.23)	58.93 (11.18)	53.28 (5.48)	60.53 (9.22)
Bimodal	76.18 (7.64)	71.35 (8.33)	68.13 (4.39)	71.71 (8.22)
B-Concat	**71.05 (6.92)**	**71.56 (6.90)**	**66.27 (5.58)**	**71.05 (6.92)**
B-Bimodal	**80.83 (7.36)**	**73.00 (6.59)**	**72.38 (4.40)**	**78.29 (6.26)**

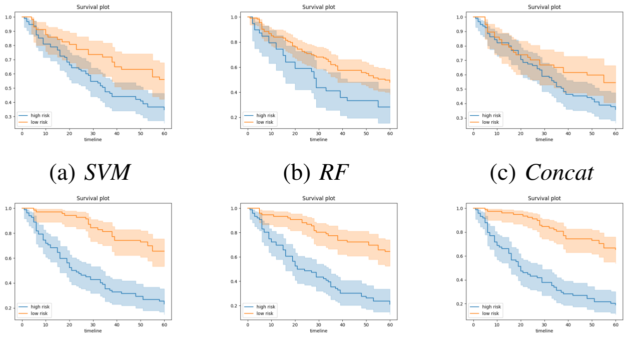
C. Results

1) NN-based models performed well with sufficient regularization: It can be observed that most NN-based models achieved better overall performance compared to *SVM* and *RF* from Table II. They were constrained to be simpler via sufficient regularization. However, *Concat* exhibited worse performance and wider confidence intervals than *SVM* in almost all metrics. This result seems to indicate that *Concat* could not combine different data modalities as well as *Bimodal*. Its wide confidence interval on every metric showed that it is sensitive to data partition.

2) Bimodal NNs excel in combining heterogeneous data types: Heterogeneous data types such as microarray and clinical data were combined well through bimodal NNs. Both *Bimodal* and *B-Bimodal* performed better than *Concat* and *B-Concat* in most metrics, respectively, with approximately the same confidence interval widths. We only observed a slight increase in the confidence interval of AUC for *B-Bimodal* compared to *B-Concat*. The results showed that bimodal NNs are better at combining information shared between heterogeneous data types.

3) Bayesian NNs achieved more robust performance: Comparing *B-Concat* and *B-Bimodal* to their non-Bayesian counterparts, we observed significant performance advantages. *B-Concat* was superior to *Concat* in all metrics by at least 0.1 and had narrower confidence intervals. With approximately the same confidence intervals, *B-Bimodal* outperformed *Bimodal* in all metrics. The results showed that we could obtain more robust predictions with similar architectures via BDL.

4) Bayesian NNs showed significant stratification in survival analysis: The results for KM analysis were summarized in Fig. 2. Except for *Concat*, all NN-based models showed significant stratification and were much better than *SVM* and *RF*. In particular, *B-Concat* and *B-Bimodal* achieved much superior stratification.



(a) *SVM* (b) *RF* (c) *Concat*

(d) *Bimodal* (e) *B-Concat* (f) *B-Bimodal*

Fig. 2: Survival analysis: the horizontal axis represents the time (in months) and the vertical axis is the survival rate.

5) Bayesian NNs were less sensitive to hyper-parameter selection: We summarized each hyper-parameter's performance with its average performance over 4-CV splits to visualize the sensitivity of model hyper-parameter selection to its stratification performance. Therefore, we could, for each model, estimate the test performance distribution for every set of hyper-parameters searched during the 4-CV training process. The PDFs for AUC were illustrated in Fig. 3. Interestingly, we found that *B-Concat* and *B-Bimodal* had much higher AUCs and were much more concentrated than their non-Bayesian counterparts. This result suggests that BDL could lower the sensitivity of hyper-parameter selection even in bimodal NNs. The hyper-parameter search space, therefore, could be reduced to save computational complexity.

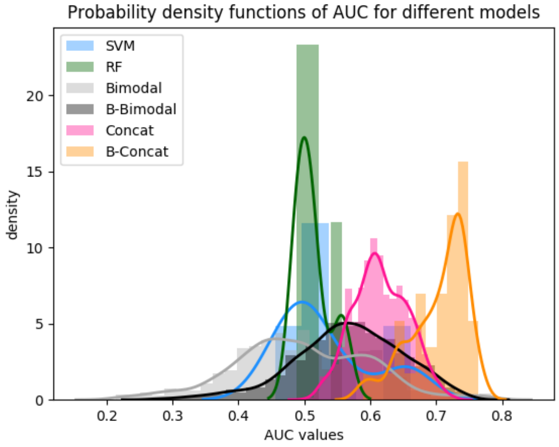


Fig. 3: Probability density functions of AUCs for various model hyper-parameter selection.

IV. Conclusion

In this work, we introduced bimodal BDL to predict the 5-year OS for colon cancer patients. With BDL, NN models performed much better and were more robust to model hyper-parameter selection and data partition. Our results showed that bimodal BDL could incorporate two data modalities well. With sufficient regularization and appropriate feature selection, BDL provided superior performance even with small patient data. We hope that these results shed light on the possibilities to use BDL in various other medical applications.

References

[1] Rebecca L Siegel, Kimberly D Miller, and Ahmedin Jemal. Cancer statistics, 2019. *CA: a cancer journal for clinicians*, 69(1):7–34, 2019.

[2] Gabrielle Jongeneel, Thomas Klausch, Felice N van Erning, Geraldine R Vink, Miriam Koopman, Cornelis JA Punt, Marjolein JE Greuter, and Veerle MH Coupé. Estimating adjuvant treatment effects in stage ii colon cancer: Comparing the synthesis of randomized clinical trial data to real-world data. *International Journal of Cancer*, 146(11):2968–2978, 2020.

[3] Aristides G Vaiopoulos, Ioannis D Kostakis, Michael Koutsilieris, and Athanasios G Papavassiliou. Colorectal cancer stem cells. *Stem cells*, 30(3):363–371, 2012.

[4] Y.-H. Lai, W.-N. Chen, T.-C. Hsu, et al. Overall survival prediction of non-small cell lung cancer by integrating microarray and clinical data with deep learning. *Scientific reports*, 10(1):1–11, 2020.

[5] L.-H. Cheng and C. Lin. Integrating ensemble systems biology feature selection and bimodal deep neural network for breast cancer prognosis prediction. *bioRxiv*, (810176), 2019.

[6] I. Goodfellow, Y. Bengio, and A. Courville. *Deep learning*. MIT press, 2016.

[7] Michael W Dusenberry, Dustin Tran, Edward Choi, Jonas Kemp, Jeremy Nixon, Ghassen Jerfel, Katherine Heller, and Andrew M Dai. Analyzing the role of model uncertainty for electronic health records. In *Proceedings of the ACM Conference on Health, Inference, and Learning*, pages 204–213, 2020.

[8] Yarin Gal and Zoubin Ghahramani. Dropout as a bayesian approximation: Representing model uncertainty in deep learning. In *international conference on machine learning*, pages 1050–1059, 2016.

[9] Durk P Kingma, Tim Salimans, and Max Welling. Variational dropout and the local reparameterization trick. In *Advances in neural information processing systems*, pages 2575–2583, 2015.

[10] C. Stark, B.-J. Breitkreutz, T. Reguly, et al. Biogrid: a general repository for interaction datasets. *Nucleic acids research*, 34(suppl_1):D535–D539, 2006.

[11] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. *arXiv preprint arXiv:1502.03167*, 2015.

[12] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: a simple way to prevent neural networks from overfitting. *The journal of machine learning research*, 15(1):1929–1958, 2014.

[13] T. Dozat. Incorporating nesterov momentum into adam. In *Workshop track ICLR 2016-2016 International Conference on Learning Representations (ICLR)*, 2016.

[14] Yiming Yang and Xin Liu. A re-examination of text categorization methods. In *Proceedings of the 22nd annual international ACM SIGIR conference on Research and development in information retrieval*, pages 42–49, 1999.

[15] F. E. Harrell. Regression modeling strategies. *BIOS*, 330:2018, 2017.

[16] T. G. Clark, M. J. Bradburn, S. B. Love, et al. Survival analysis part i: basic concepts and first analyses. *British journal of cancer*, 89(2):232, 2003.

[17] R. Peto, M. C. Pike, P. Armitage, et al. Design and analysis of randomized clinical trials requiring prolonged observation of each patient. ii. analysis and examples. *British journal of cancer*, 35(1):1, 1977.

[18] R. Fluss, D. Faraggi, and B. Reiser. Estimation of the youden index and its associated cutoff point. *Biometrical Journal: Journal of Mathematical Methods in Biosciences*, 47(4):458–472, 2005.

[19] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

[20] Jiaxin Shi, Jianfei Chen, Jun Zhu, Shengyang Sun, Yucen Luo, Yihong Gu, and Yuhao Zhou. Zhusuan: A library for bayesian deep learning. *arXiv preprint arXiv:1709.05870*, 2017.