A Closer Look at Deep Learning Survival Prediction for High-Throughout Data

Wankang Zhai [#]
Houston International Institute
Dalian Maritime University
Dalian, Liaoning 116026, China
wzhai2@cougarnet.uh.edu

Yuhan Wang*[#]
Houston International Institute
Dalian Maritime University
Dalian, Liaoning 116026, China
557537577mm@dlmu.edu.cn

Feng Tang
Houston International Institute
Dalian Maritime University
Dalian, Liaoning 116026, China
2804087776@qq.com

Boyang Chen
Houston International Institute
Dalian Maritime University
Dalian, Liaoning 116026, China
1069029621@qq.com

[#]These authors contributed equally to this work.
*Corresponding Author

***Abstract*—Research on survival prediction with deep learning has recently emerged. We found that deep learning has great advantages in the process of survival prediction. In this article, we compare traditional deep learning with two baseline models. They are Cox-nnet and DeepSurv. To summarize: When mining TCGA (The Cancer Genome Atlas) high-throughput genetic data sets for analysis: the results obtained by the simple and easy-to-use model are more credible, and its concordance index (evaluation of survival prediction model indicators) is also higher. We compare the loss curves of the two models and give our analysis.**

Keywords-Deep Learning, High-Throughout Data, Survival Prediction, Cox Model

I. INTRODUCTION

Cancer, the eternal challenge, the continuation of Prometheus. Throughout the history of mankind, cancer has always been a mysterious and frightening existence. This problem has puzzled scientists for generations for thousands of years. Even today, despite the rapid development of medical technology, cancer is still a very challenging opponent that leaves humans feeling helpless.

Nowadays much work has been done to handle survival prediction. Starting from the earliest Deepsurv [1], a hidden layer is used combined with an advanced activation function to prevent over-fitting technology to predict the patient's risk score and provide corresponding diagnosis and treatment recommendations. Cox-nnet [2] solves the high-throughput problem and uses high-throughput data to predict patient survival data. Two works laid the foundation for the use of deep learning. AE-cox [3] The author used Auto-Encoder technology to first reduce its dimension and then use cox processing.

For high-throughput data, one of the most difficult problems is overfitting. There is a lot of work done to prevent overfitting from happening. Dropout [4] is a widely employed regularization technique. Its function involves randomly setting the outputs of a subset of neurons to zero during each training iteration. This measure effectively curbs the network's tendency to overly depend on specific neurons, thereby enhancing the network's robustness and its ability to generalize. L2 regularization involves adding the sum of squared weight parameters to the loss function. By restraining excessive weights, L2 regularization enhances the model's generalization ability, mitigating the risk of overfitting.

In this article, we discovered the impact on overfitting at the scale of the number of hidden layers. As well as the thorny problems encountered by the model when processing high-throughput data, and our analysis is given.

II. THE BASELINE SURVIVAL PREDICTION MODEL

A. General Structure of the Network

For this model, it implements survival prediction and diagnosis recommendation. This article uses two blocks, antecedent and consequent, to complete survival prediction and diagnosis recommendation. Among them, as a prerequisite, we use the DNN feedforward network to fully connect multiple hidden layers, and add a Drop layer to randomly discard the output of a part of the neurons during the training process to reduce overfitting. This model also integrates advanced activation functions and hyperparameter optimization methods to improve the accuracy of the algorithm.

For the consequent, the model inputs the log risk score predicted by the antecedent into a Cox-PH model, resulting in a time-varying risk equation. By combining the antecedent and the consequent, the final prediction result can be obtained. When the feature This article will introduce the structure of the network's pre-ware and post-ware in order.

B. Introducntion to The TCGA Dataset

The data set used in this article is the TCGA data set. TCGA is a large-scale collaborative project supported by the National Cancer Institute and the National Human Genome Research Institute. The TCGA data set contains a variety of cancer types, genomics, epigenomics and other types of data, and has a wide range of patient samples. The reason for choosing this data set is to verify the prediction degree of the model for high-dimensional nonlinear data, and its final prediction results are better than those of previous models proposed in academia.

Each column of the TCGA data set used in this article (except the last two columns) is the expression level of genes that may cause cancer in each patient, and each row is the data on the expression level of all oncogenes in a cancer patient. The penultimate column of the data set is the survival status of the patient during the monitoring time (recorded as E). If E=1, it means this data was right censored [6], we only know that the survival time is greater than the observation time. If E=0, the observation time is the survival time. Survive eventually within the surveillance time. The penultimate column is the survival period of the patient. If the patient eventually dies because of this reason, the number represents the number of days from the beginning of monitoring to the end of monitoring; if the patient is right censored data, the only thing we know is he/she survived longer than this specific time. The format of the example TCGA data we handled set is shown in Figure 1.

Tags	RAB4B	C12orf5	RNF44	...	CCDC7	SPRY3	CRLF2	time	status
TCGA.BF.A1PU	2.723684	3.629537	3.466128	...	0.0570756	0.0585907	0.1508326	387	0
TCGA.BF.A1PV	1.761349	3.555166	3.587091	...	0.1143752	0.1061106	0.0215556	14	0
TCGA.BF.A1PX	2.737028	3.584654	3.442409	...	0.1048418	0.2390276	0.0502759	282	1
TCGA.BF.A1PZ	2.527428	3.301321	3.1122889	...	0.0826875	0.0406105	0.0247608	853	0
TCGA.BF.A1Q0	2.255058	2.008007	3.6712988	...	0.405367	0.0536363	0.0338145	831	0
TCGA.BF.A3DJ	2.325468	3.11704	2.2726344	...	0.3550016	0.1498653	0.0486106	464	0
TCGA.BF.A3DL	2.493967	2.673465	3.446451	...	0.0298834	0.0624819	0.0062153	769	0

Figure 1 The format of TCGA dataset

After data cleaning of the TCGA [4] data set, the format of the data set mainly consists of three variables: the patient's gene expression(features), the patient's survival time, and the patient's survival status. In this network, the above three variables will be used as inputs, and the final output is the logarithmic hazard function $h(x)$, which is obtained by taking the logarithm of the hazard function. The meaning of the hazard function is the probability of the patient dying at time t, and its calculation is the derivative of the survival function with respect to time t. The hazard function is given as (1).

$$\lambda(t) = \lim_{\delta \to 0} \frac{\Pr(t \le T < t + \delta \mid T \ge t)}{\delta} \tag{1}$$

It can be seen that the network is a regression model that feeds back the regression network of individual survival rates by inputting different high-dimensional nonlinear features.

III. The Structure of The Antecedent Network

We use the DNN deep neural network to build the antecedent network. Our model uses a multi-layer perceptron. The predicted output of the network is a value that represents the patient's health risk. Its loss function is defined as (2).

$$l(\theta) := -\frac{1}{N_{E=1}} \sum_{i:Ei=1} (\hat{h}_\theta(x_i) - \log \sum_{j \in \Re(Ti)} e^{\hat{h}\theta(xj)}) + \lambda \| \theta \|_2^2 \tag{2}$$

The loss function is obtained by taking the logarithm of the Cox partial likelihood equation. The Cox partial likelihood [5] equation is a method used to estimate model parameters in the Cox proportional hazards model. Its definition formula is (3).

$$L_c(\beta) = \prod_{i:Ei=1} \frac{\hat{r}_\beta(x_i)}{\sum_{j \in \Re(Ti)} \hat{r}_\beta(x_j)} = \prod_{i:Ei=1} \frac{\exp(\hat{h}_\beta(x_i))}{\sum_{j \in \Re(Ti)} \exp(\hat{h}_\beta(x_j))} \tag{3}$$

The partial likelihood function is the reverse application of probability and needs to select the likelihood model with the highest probability. Therefore, this article uses the expected value to implement the calculation of the loss function, and adds L2 regularization at the end to prevent overfitting and fit it into the final loss function. For (3), it can be further understood as: the probability of the i-th person dying at time ti divided by the accumulation of the probabilities of all people participating in the experiment who may die at time t_i, so it is the probability of a certain person dying at time t, The accumulation of death for each person at time t is the maximum partial likelihood equation. At this point t has been decided by $\lambda_0(t)$ and has nothing to do with x.

At the same time, this paper adopts advanced optimization methods. There are multiple optimization methods such as weight decay regularization, ReLU activation, batch normalization, SELU, dropout, SGD, Adam, gradient clipping, decay_rate and Learning_rate grid adjustment strategies.

A. DNN(Deep Neural Network)

DNN is a generalization of MLP. It improves the learning ability and representation ability of the model by increasing the depth (that is, increasing the number of hidden layers). Therefore, multiple hidden layers can better reflect the characteristics of high-dimensional data. A deep neural network is a neural network composed of multiple hidden layers. Each hidden layer contains multiple neurons that are connected to neurons in the previous and subsequent layers. The depth of a deep neural network refers to the number of hidden layers it has. The benefit of using deep networks to model high-dimensional non-linear data is that it is able to learn more complex features and relationships because each hidden layer can learn different levels of abstract features.

B. Dropout Layer

Dropout is a regularization technique applied in neural networks, especially suitable for processing high-dimensional nonlinear data. Prevent overfitting: In high-dimensional nonlinear data, the model is prone to learning noise or specific local patterns in the data, and these patterns are not very useful for generalizing to new data. Dropout layers can help prevent the neural network from relying too much on certain neurons, thereby reducing the risk of overfitting. At the same time, it helps to increase robustness, so that even if some neurons are not activated during testing, the network can still make predictions effectively. Dropout reduces the dependencies

between neurons by randomly turning off some neurons at each training step. This helps prevent the network from forming complex co-adaptations, allowing each neuron to independently learn an effective representation of the input without overly relying on other neurons.

C. Grid Method to Adjust Hyperparameters

This article makes the results convincing by adjusting the grid search engine, adopting different hyperparameter configurations, and using the internal and external five-fold cross-validation method. The grid method verification has the following advantages: the grid search method achieves a comprehensive search of the entire hyperparameter space by permuting and combining predefined hyperparameter combinations. This ensures that every possible combination of hyperparameters is tried within a given range to find a relatively optimal combination. Interpretability: Since the grid search method performs an explicit iteration of all possible hyperparameter combinations, the results The explanation is better. Helps gain a deeper understanding of model behavior.

IV. The Structure Of Cox Propagation Model

The consequent network used in this article is the Cox-PH regression prediction network. The generated risk-log function [7] is input into the Cox regression model for linear change, and finally converted into a risk function with x characteristics (4).

$$\lambda(t \mid x) = \lambda_0(t) \cdot e^{h(x)} \tag{4}$$

Among them, the right side of the equal sign is the e index RR of the log-risk function obtained by our nonlinear neural network. When the value of RR is greater than 1, it means that the risk is high at this time; when the value of RR is equal to 1, it means that the variable has no impact on survival; when the value of RR is less than 1, it means that the risk is small at this time.

There are some restrictions that need to be met before the Cox hazard regression model can be applied. First, the ratio of the hazard functions of any two individuals, that is, the hazard ratio (HR), maintains a constant ratio regardless of time t; second, the effect of the covariates in the model does not change with time. Because the algorithm in this article focuses on the nonlinear relationship of x, this model is also very reasonable as a consequence.

V. The Final Network And Evaluation Index

A. The Combination of Two Networks

After combining the antecedent and the consequent, we have a complete regression network. At this time, only the features need to be adjusted to complete the conversion of survival prediction and diagnosis recommendation. When the x feature is a survival feature, that is, a patient's attribute, this article infers the patient's survival rate based on the attributes and establishes a survival prediction network. When the x feature is a diagnostic means, the network is a diagnostic recommendation prediction. Here, the recommendation function (5) is proposed.

$$rec_{ij}(x) = \log\left(\frac{\lambda(t;x \mid \tau = i)}{\lambda(t;x \mid \tau = j)}\right) = \log\left(\frac{\lambda_0(t) \cdot e^{hi(x)}}{\lambda_0(t) \cdot e^{hj(x)}}\right) = h_i(x) - h_j(x) \tag{5}$$

The rec function represents the horizontal comparison of risk predictions between diagnostic means i and diagnostic means j. When the risk function of the two diagnostic methods is measured, it can be found that at time t, the recommendation function is greater than 0, which means that the diagnosis and treatment method i is better than the method j. This article uses the same network to get rid of the shackles of expert's knowledge and re-measure the advantages and disadvantages of diagnosis and treatment recommendations from a quantitative perspective.

Table 1 Diagnosis and treatment recommmendation results

	G1	G2	G3	G4	T1	T2	T3	Rate
x_b	0.5723	0.3452	0.1267	0.8726	0	0	0	40.25%
x_b	0.5723	0.3452	0.1267	0.8726	1	0	0	42.23%
x_b	0.5723	0.3452	0.1267	0.8726	0	1	0	41.28%
x_b	0.5723	0.3452	0.1267	0.8726	0	0	1	63.12%

B.

Table 1 illustrates the treatment recommendation result of a given patient. x_b represents a patient who suffers from the cancer. $G1$ to $G4$ are the gene representation of the given patient. $T1$ to $T3$ are the methods that the patient may take during the treatment. In the Table, for the given example, $T1$ represents radiotherapy, $T2$ represents hormone therapy and $T3$ represents chemotherapy. The last column, Rate, represents the survival rate of the given patient. From this example, it can be concluded that for patient x_b, the survival rate with chemotherapy($T3$) is higher, as high as **63.12%**. It can be seen from this that this network can implement diagnosis and treatment algorithm recommendations for cancer patients, and has great application prospects in the medical field.

C. The Network Evaluative Indicators C-index

Headings, or heads, are organizational devices that guide the reader through your paper. There are two types: component heads and text heads.

C-index is an index used to evaluate the prediction accuracy of the model and is often used in survival analysis. This article will introduce the meaning of the evaluation index C-index of the algorithm network. The calculation of the C-index involves the relative ranking of samples by the model. In survival analysis, for each pair of samples, if the model correctly predicts that the sample with a shorter survival time does have a shorter survival time and the sample with a longer survival time does have a longer survival time, the pair of samples is considered "consistent" "of. The C-index is the proportion of all "consistent" pairs of samples to all comparable pairs of samples.

The specific calculation method of C-index is as follows: For a given pair of samples i and j, if the survival time of sample i is less than the survival time of sample j, and the relative ranking of predicted risks by the model for this pair of samples is correct, then The pair of samples is considered consistent. If samples i and j are observed at the same time, the pair of samples is not included in the comparison. Calculate the proportion of pairs of samples that are consistent, the C-index. The C-index ranges from 0 to 1, where 0 means there is no consistency in the model's predictions and 1 means the model's predictions are completely consistent. Generally speaking, the closer the C-index is to 1, the better the performance of the model.

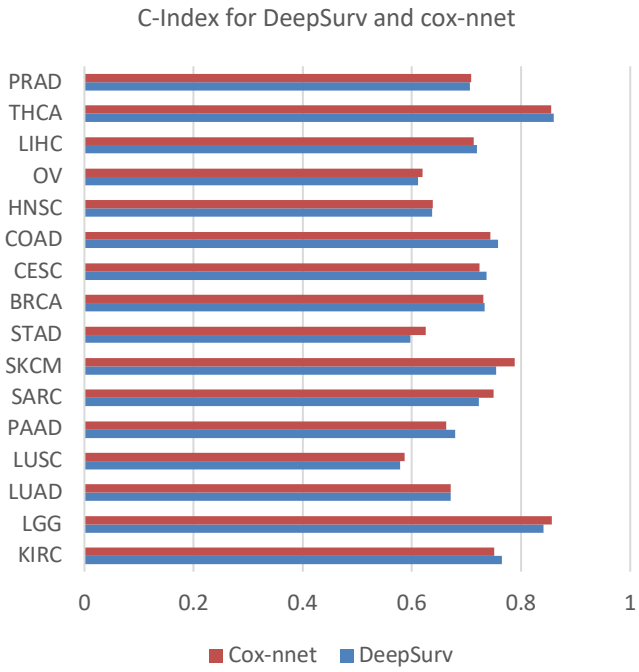

Figure 2 C-index in each model

From Figure 2, we can see that compared with other models, the C-index value of the two models. As illustrated in the table, we can discern the prediction efficacy of DeepSurv and Cox-nnet across various cancer types. It is apparent that in the majority of datasets, the overall performance of DeepSurv appears to be inferior to that of Cox-nnet. However, within high-quality datasets, characterized by a substantial number of training samples, DeepSurv demonstrates superior performance. For instance, in datasets like PAAD (496 samples) and KIRC (411 samples), where ample data is available, the utilization of more intricate networks yields significant performance enhancements. Conversely, in datasets with fewer samples, such as SKCM (99 samples) and SARC (258 samples), Cox-nnet outperforms DeepSurv.

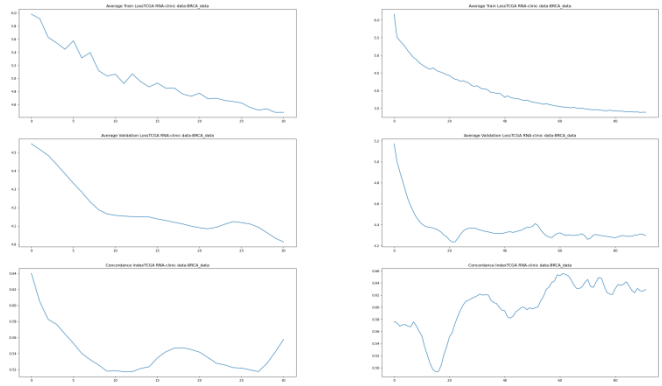

Figure 3 Curves for TCGA mi-RNA BRCA data

Three pictures on the left are the TrainLoss, TestLoss, and C-index of DeepSurv during the training process. And right Pircture is the same for Cox-nnet. In our experiment, we employed external five-fold cross-validation to fine-tune hyperparameters, subsequently utilizing the optimized settings for model training and prediction.

From Figure 3, it's apparent that we conducted 100 rounds of training for both models. While both exhibited fitting trends, DeepSurv did not display performance improvement with increased model fitting, whereas Cox-nnet consistently demonstrated enhanced performance while maintaining stability. Despite fluctuations within a certain range, Cox-nnet ultimately outperformed DeepSurv.

We conducted a detailed analysis to elucidate these findings:

A. Insufficient Training Samples: The limited number of training samples relative to the features may have led to suboptimal performance. In such scenarios, simpler models tend to outperform complex ones, as the latter may overfit the noise in the training data and struggle to generalize to new data.

B. Overfitting: The deeper architecture of DeepSurv may have exacerbated the overfitting problem. While the model may perform well on the training set, its performance on unseen data could be compromised. Mitigating this issue involves reducing the number of hidden layers to promote better generalization to new data.

C. Complexity of Model: The abundance of hidden layers in DeepSurv significantly increases the network's structural complexity, making it more susceptible to getting trapped in local minima during the learning process. Simplifying the architecture could potentially improve performance and reduce overfitting.

VI. CONCLUSION

In this article, we compare two common baseline models within the Cox proportional hazards model framework. Our analysis of the prediction results leads us to conclude that Cox-nnet exhibits advantages in processing high-throughput data. Furthermore, we delve into the verification process of its training

rounds and uncover that DeepSurv encounters challenges in processing such data.

We illustrate the reasons behind this phenomenon. Simultaneously, we unearth the promising prospects for the application of survival analysis. For instance, a deep learning prediction model can be leveraged to forecast patient survival, with added diagnostic and treatment recommendations embedded within the deep learning prediction system. Specifically, by integrating Cox-nnet with a diagnosis and treatment recommendation system, we can furnish patients with rational diagnosis and treatment suggestions. This is achieved by analyzing the patient's gene expression and medication status to enhance both their three-year and five-year survival rates.

We observe that as few-shot learning matures, a broader array of algorithms, such as meta-learning and transfer learning, can be applied. Coupled with pre-training and fine-tuning on the TCGA multi-omics database, this approach further mitigates the issue of overfitting.

REFERENCES

[1] Katzman, J. L., Shaham, U., Cloninger, A., Bates, J., Jiang, T., & Kluger, Y. (2018). DeepSurv: personalized treatment recommender system using a Cox proportional hazards deep neural network. BMC medical research methodology, 18, 1-12.

[2] Ching, T., Zhu, X., & Garmire, L. X. (2018). Cox-nnet: an artificial neural network method for prognosis prediction of high-throughput omics data. PLoS computational biology, 14(4), e1006076.

[3] Huang, Z., Johnson, T. S., Han, Z., Helm, B., Cao, S., Zhang, C., ... & Huang, K. (2020). Deep learning-based cancer survival prognosis from RNA-seq data: approaches and evaluations. BMC medical genomics, 13, 1-12.

[4] Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., & Salakhutdinov, R. (2014). Dropout: a simple way to prevent neural networks from overfitting. The journal of machine learning research, 15(1), 1929-1958.

[5] Cox, D. R. (1975). Partial likelihood. Biometrika, 62(2), 269-276.

[6] Koul, H., Susarla, V., & Van Ryzin, J. (1981). Regression analysis with randomly right-censored data. The Annals of statistics, 1276-1288.

[7] Therneau T, Grambsch PM. Modeling Survival Data : Extending the Cox Model. New York: Springer; 2000.