Axentiva

INTEGRATING DEEP LEARNING AND STATISTICAL METHODS THROUGH STATISTIC-INFORMED NEURAL NETWORKS IN SURVIVAL ANALYSIS

XLIII Jornadas de Economía de Salud (AES)
June 2024

Yovani Torres ; **Óscar Martínez-Pérez** ; Carlos Crespo .

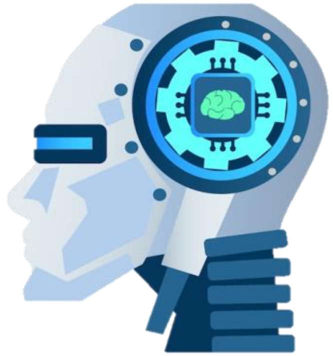
axentiva.com

INTRODUCTION

There is a **growing interest** in the medical community in applying **Deep Learning (DL)** to fit models and to predict **clinical outcomes.**[1]

What is DL?

An advanced machine learning technique with **multiple layers of neural networks** that uses data to model and solve complex problems, **imitating how humans learn**, and gradually **improving its accuracy.**



What is DL for?

It allows for achieving **better results** by identifying complex patterns in the data and relaxing assumptions, thereby **improving traditional models**.

The **objective** of this study was to **compare the usefulness** of measuring treatment effectiveness using **survival analysis** with a classic/traditional **parametric Cox model** vs. an **innovative DL-based model.**

1. Kantidakis, G., Hazewinkel, A. D., & Fiocco, M. (2022). Neural networks for survival prediction in medicine using prognostic factors: a review and critical appraisal. Computational and Mathematical Methods in Medicine, 2022

METHODS

Statistic Informed Neural Network (SINN)

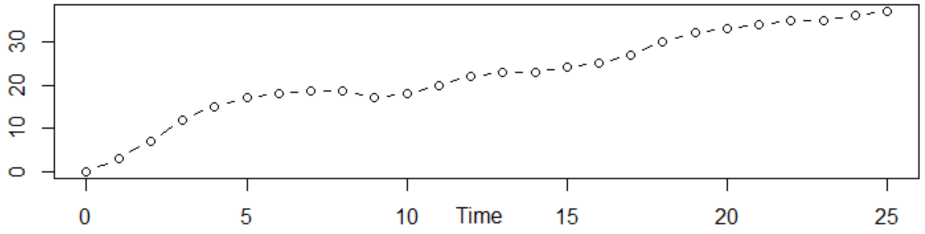
A **cohesive framework** combining traditional statistics and modern machine learning. SINNs are a variant of neural networks that integrates statistical principles to enhance data interpretation.

An example of how neural networks can be forced to follow a physics equation...

Displacement is defined as: $S_{(t)} = \boldsymbol{S_0 + at}$

Initial position (S_0)

Acceleration parameters (a)



We need to **model the behaviour of $\boldsymbol{S_{(t)}}$** given the estimation of S_0 and a, modeled based on the physics information (formula) of $S_{(t)}$'s behaviour that must be always satisfied.

This concept can be incorporated into the survival model with **statistic information** instead of physics information. Specifically, **our SINN integrated the Weibull survival function as the baseline hazard in its formulation**

Starting from the Weibull cumulative distribution function, we obtain our Weibull survival function: $F_{(t)} = 1 - e^{-\left(\frac{t}{a}\right)^b}$ ➡ $S_{(t)} = 1 - F_{(t)} = e^{-\left(\frac{t}{a}\right)^b}$

This is used as the basis for calculating the ordinary differential equation (ODE) to integrate into the model:

$$\frac{\partial S_{(t)}}{\partial t} = -\frac{b}{a}\left(\frac{t}{a}\right)^{b-1} e^{-\left(\frac{t}{a}\right)^b}$$

2. Zhu, Y., Tang, Y. H., & Kim, C. (2023). Learning stochastic dynamics with statistics-informed neural network. Journal of Computational Physics, 474, 111819.

METHODS

Why a Weibull distribution?

It models a **broad range of random variables** such as the time to failure or time between events. The Weibull distribution extends the exponential distribution to **allow constant, increasing, or decreasing hazard rates**, thus relaxing the classical proportional hazard assumption.

Data loss function built using the Weibull survival function ($S_{(t)}$).

Measures the discrepancy between the neural network predictions and the actual observations**.**

+

Physics loss function built using the ODE as the baseline hazard

This function is aigned with the Cox model's structure ensuring that the neural network's predictions adhere to the statistical information given.

=

Total loss function (weighted sum)

$$Loss_{total} = \lambda_1 \cdot Loss_{data} + \lambda_2 \cdot Loss_{physics}$$

This function is minimized by the neural network based on the estimation of all parameters.

The hazard function of the **classical Cox model** is specified as follows:

$$h_{(t|X)} = h_{0(t)} e^{(X^T \beta)}$$

where $h_{(t|X)}$ is the hazard at time *t* given predictor values X, $h_{0(t)}$ is the baseline hazard and $\beta = (\beta_1, \dots, \beta_p)$ is the parameter vector.

ODE: Ordinary differential equation

METHODS

DATA USED

- **'Lung' dataset**
  From the **R 'survival' package** as a benchmark for our **analysis**
- All variables were included as **covariates**

EVALUATION OF THE MODELS

- **Concordance Index (CcI)**
  Measures from 0 to 1 whether the model prediction is in concordance with the actual data
- **Akaike Information Criteria (AIC)**
  Evaluates both the goodness of fit and the complexity of the model

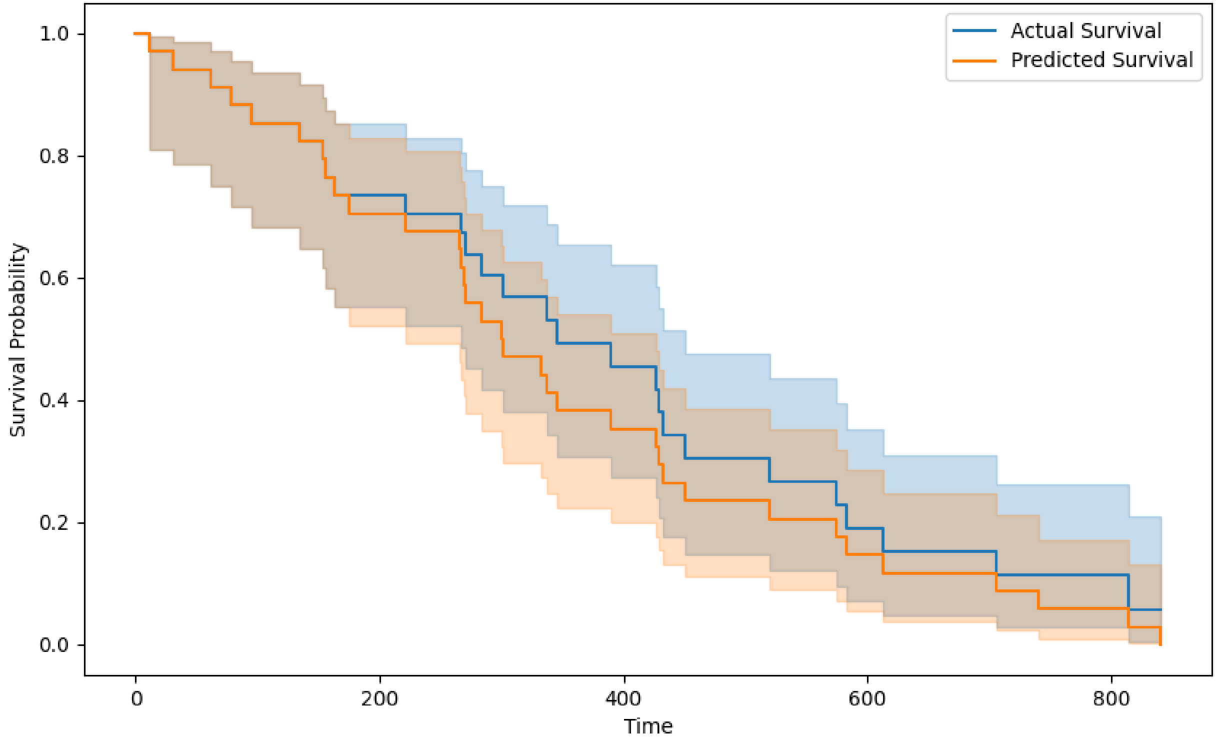
SOFTWARE USED

- **Classical model**
  RStudio®
- **SINN model**
  python™

RESULTS

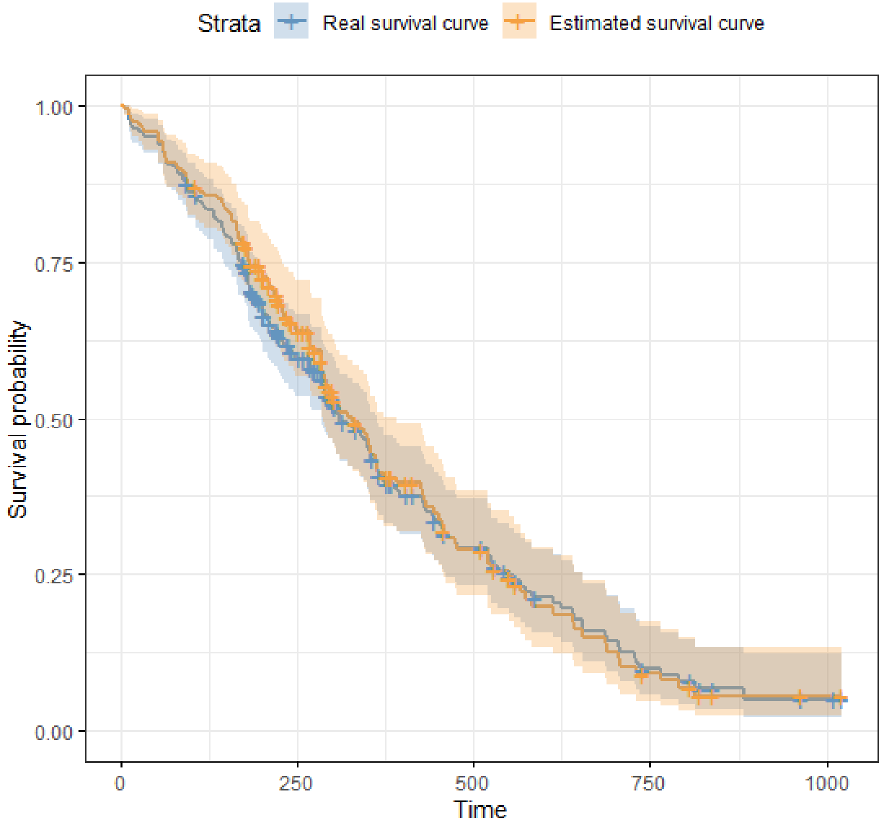
Statistic Informed Neural Network (SINN)

The **survival curve** of the regression with **SINN model** compared with **the curve of the survival model** can be seen in the figure below.



Parametric Cox model

The **survival curve** of the regression with **parametric Cox model** compared with **the curve of the survival model** can be seen in the figure below.



RESULTS

The **Concordance Index** of the **SINN model** after **10,000 simulations** was **0.9723**, indicating a **high level of discrimination capacity**. On the other hand, the **Concordance Index** of the **parametric model** is **0.6512**, a relatively **low prediction capacity**.

SINN's higher **AIC** (**1994.0; Cox: 1011.504**) suggests the **parametric model** achieves a **superior balance** between **data fitting** and **model simplicity** since **AIC penalises the model's complexity**.

CONCLUSIONS

The **innovative SINN model** proves to be a **versatile tool** for **measuring treatment effectiveness.**

It is able to expand the scope **beyond conventional methodologies.**

It also allows for a **highly differential survival prediction precision** compared to the parametric model.

Axentiva

Thank you

CONTACT DETAILS

Carlos Crespo Palomo, PhD
CEO
ccrespo@axentiva.com

Neural Network:

- It is a composition of functions: $NN_{z(x)} = \left[\left(\sigma_n(L_n)\right)_{n=1}^{k}\right]$ where $\sigma = \frac{x}{1+e^{-x}}$ (sigmoid or other) is the activation function, and $L_n = Ax + b$ a linear expression being x the data values $[x_0, \dots, x_n]$, A a matrix of parameters, and b a vector of parameters.
- The structure of this specific neural network is the following:
    - Input layer where data (x) are introduced:

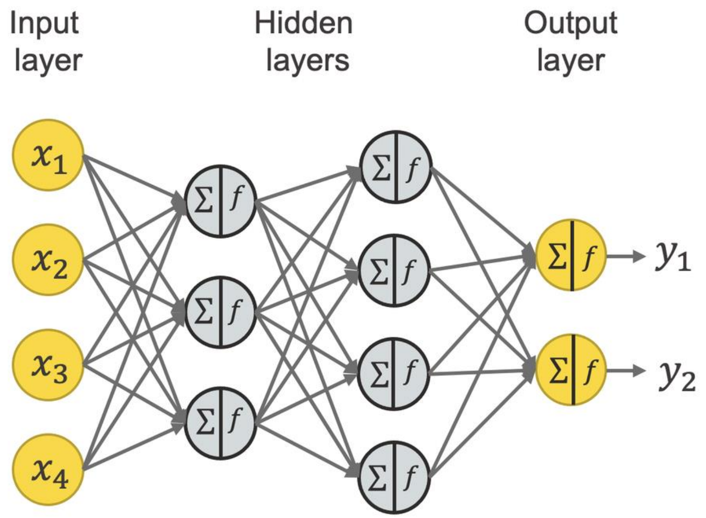
$$fcs = \sigma(Ax + b) \implies \hat{x}, A, b$$

    - Hidden layer where results produced in the previous layer $(\hat{x})$ are introduced:

$$fhs = \sigma(C\hat{x} + d) \implies \tilde{x}, C, d$$

    - Output layer where results produced in the previous layer $(\tilde{x})$ are introduced and gives us the output we are looking for y (the probability of survival in every time):

$$fs = \sigma(E\tilde{x} + g) \implies y$$



Using the *Weibull survival* function ($S_{(t)}$), we can build a **data loss function** that measures the discrepancy between the neural network predictions and the actual observations**:**

$$Loss_{data} = \frac{1}{N}\sum_{i=1}^{N}\left(NN_{x(t_i,\Phi)} - S_{t_i}\right)^2$$

where $NN_{x(t_i,\Phi)}$ denotes the neural network, Φ represents the neural network parameters, and $S_{(t)} = e^{-\left(\frac{t}{a}\right)^b}$.

Using the ODE as the baseline hazard, we can build a **physics loss function** aligned with the **Cox model's structure** ensuring that the neural network's predictions adhere to the statistical information given:

$$Loss_{physics} = \frac{1}{M}\sum_{i=1}^{M}\left(\frac{\partial}{\partial t}NN_{x(t_i,\Phi)} + \frac{b}{a}\left(\frac{t}{a}\right)^{b-1} e^{-\left(\frac{t}{a}\right)^b}\right)^2$$

The **total loss function** is a weighted sum of the data loss and the physics loss:

$$Loss_{total} = \lambda_1 \cdot Loss_{data} + \lambda_2 \cdot Loss_{physics}$$

This function is minimized by the neural network based on the estimation of all parameters.