Supplementary Information

Leveraging the variational Bayes autoencoder for survival analysis

Patricia A. Apellániz*[1], Juan Parras[1], Santiago Zazo[1]
[1]Information Processing and Telecommunications Center, Universidad Politécnica de Madrid, 28040, Madrid, Spain
*Corresponding author: patricia.alonsod@upm.es

Sensitivity Analysis

In artificial intelligence models, particularly in complex deep learning architectures, sensitivity and robustness analyses are essential to understand how variations in the input data influence the model predictions. These analyses are crucial for evaluating the stability and reliability of a model's output, providing insight into whether small changes in input features could significantly impact the results. This is especially important in survival analysis tasks, where accurate and stable predictions are critical for applications in healthcare and other fields.

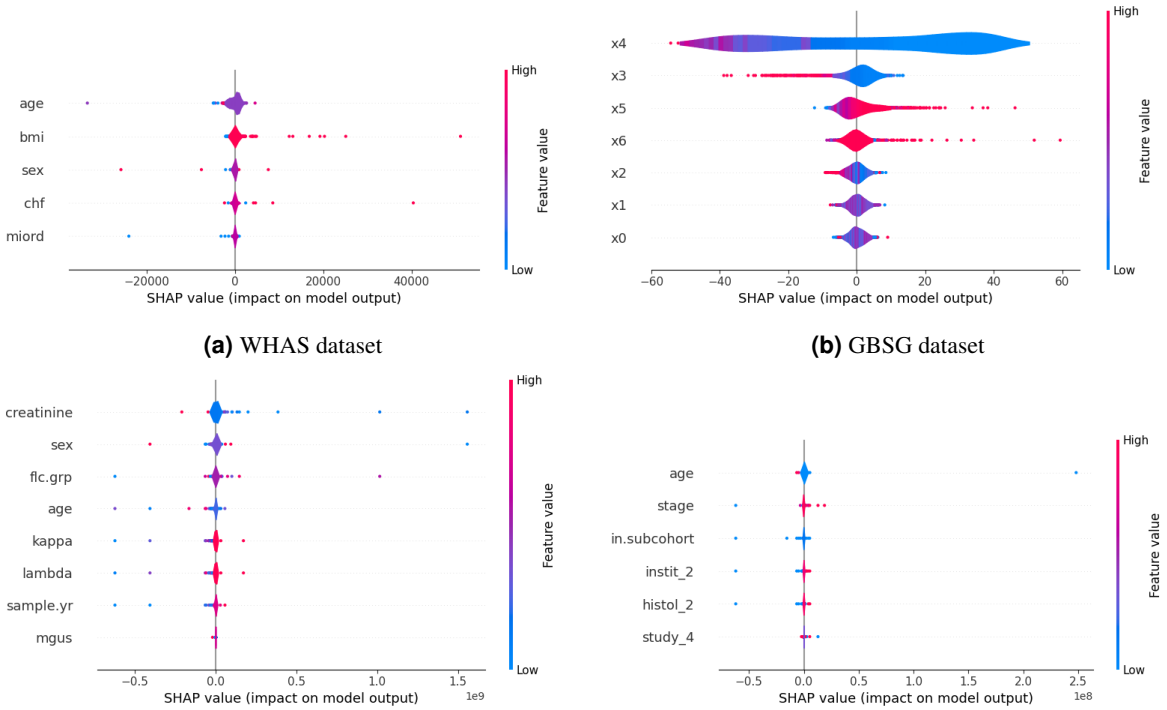
We have implemented two strategies to ensure the robustness of our proposed model, SAVAE. We run the model for each dataset using ten different random seeds and then perform a 5-fold cross-validation. This approach allows us to assess the variability in model performance across different training-test splits and random initializations, ensuring that our results are not excessively dependent on a single set of conditions. By averaging the results across these multiple runs, we provide more reliable and stable performance metrics that account for the variability inherent in the data and model training process.

In addition to these measures, we also performed a Shapley value-based sensitivity analysis to explore how the input features impact the predicted survival times. Shapley values, originating from cooperative game theory, offer a mathematically grounded method to determine the contribution of each input feature to the final prediction. By calculating the Shapley values for each feature, we can quantify the importance of each feature in influencing the model's output. This analysis adds transparency and interpretability to the model's decision-making process and helps us understand the relative influence of different covariates on survival time predictions. These combined methods provide a thorough sensitivity and robustness assessment, ensuring the model performs consistently across different conditions and offering insight into how input features affect the model's predictions. These analyses strengthen the overall validity and interpretability of our model. The detailed results of the sensitivity analysis are presented below.

In particular, our analysis focuses on determining each input feature's importance. For this purpose, we used SHAP (SHapley Additive exPlanations)[1], an approach that enables global analyses for each dataset and the model in question. Specifically, we will use Python's Shap module to compute the importance of the feature. Following the official documentation, we determined the importance of each feature for each dataset.

Figure 1 represents the importance of the features for four datasets, chosen because they have the fewest characteristics, allowing for better visualization and analysis. To assess the robustness and generalizability of our model, we computed SHAP values for the training set of each dataset. We have represented the analysis for just one fold from one training seed. Recall that we trained the model for each dataset using ten different seeds and employed a K-Fold cross-validation technique, where the data is split into five sets. This means the SHAP analysis shown here represents only a subset of the complete robustness check. Each violin plot shows the feature importance for a particular dataset, with features ranked from highest to lowest based on their contribution to the predicted survival times. The color of the points within the plot indicates whether a feature's value is higher or lower compared to other observations, and the horizontal axis shows the SHAP values. All individual dots represent a single observation, and the SHAP values on the horizontal axis illustrate how much each feature influences the prediction, with the color gradient indicating whether the feature value is high or low. This visual representation helps us interpret each feature's effects on survival predictions in the training set. For example, in the GBSG dataset, lower values of the *X4* feature have a positive impact on the time prediction. Furthermore, this approach allows us to examine the consistency between our model's interpretation and clinical understanding. As observed, the age feature consistently ranks as one of the most important in predicting survival times across all datasets, which is logical given that age is a well-established factor in survival analysis[2–4]. This alignment between model interpretation and clinical reasoning enhances the trustworthiness and applicability of the model's predictions in real-world healthcare scenarios.

This interpretability is particularly critical in the medical domain, where understanding the relationship between clinical features and predicted survival times can provide significant insights for practitioners. For example, certain clinical features,



(a) WHAS dataset

(b) GBSG dataset

(c) FLCHAIN dataset

(d) NWTCO dataset

Figure 1. Comparing blue car vs. red car.

such as age, comorbidities, or treatment variables, might strongly influence survival predictions based on their SHAP values. These insights help guide healthcare professionals in interpreting how these factors affect individual patient outcomes, leading to better-informed decisions. Therefore, this analysis serves as an additional check on our model's contribution, demonstrating that it is not only accurate but also interpretable, offering valuable insights that are consistent with clinical expectations. The ability to interpret the model's predictions further validates its robustness and enhances its potential for application in sensitive domains like healthcare.

Ablation Study

To evaluate the contribution of each component of the proposed SAVAE model and justify their inclusion, we conducted a comprehensive ablation study. This analysis aims to assess the impact of various architectural choices and their role in the model's overall performance. By systematically varying critical elements of the architecture, we measured the resulting changes in the performance metrics, the Concordance Index (C-index), and the Integrated Brier Score (IBS).
The SAVAE model architecture consists of three DNNs: one encoder and two decoders, each responsible for inferring covariates and time parameters. For the ablation study, we focused on the following key components:

- **Latent Space Dimensionality:** We initially fixed the latent space to 5 dimensions. In this study, we evaluated the effect of varying the dimensionality of this latent space (e.g., reducing to 3 or increasing to 10) to determine its influence on the model's ability to capture underlying data patterns effectively.
- **Neurons:** Each module includes two hidden layers with 50 neurons. We investigated whether reducing or increasing the number of neurons negatively impacts model performance, particularly in modeling complex non-linear relationships between covariates and survival times.
- **Dropout Rate:** The model uses a dropout rate of 20% in the first hidden layer of each decoder. We assessed the effect of removing dropouts and experimented with varying dropout rates (50%) to understand its role in preventing overfitting and ensuring model generalization.

We first selected the METABRIC dataset to illustrate the results, as it contains the highest dimensionality regarding the number of features. This aspect is essential for the VAE's latent space generation. Results are shown in terms of C-index and IBS values, which were obtained using five cross-validation folds for each of the ten training seeds, ensuring robust results as

Latent Space Dimension	Number of Neurons	Dropout Rate	C-index	IBS
3	10	0	0.594	0.186
		0.2	0.590	0.185
		0.5	0.589	0.187
	50	0	0.598	0.184
		0.2	0.596	0.188
		0.5	0.590	0.187
	500	0	0.595	0.186
		0.2	0.593	0.185
		0.5	0.592	0.185
5	10	0	0.596	0.185
		0.2	0.593	0.186
		0.5	0.588	0.188
	50	0	0.604	0.185
		0.2	0.598	0.185
		0.5	0.594	0.185
	500	0	0.598	0.185
		0.2	0.595	0.185
		0.5	0.599	0.187
50	10	0	0.587	0.187
		0.2	0.582	0.189
		0.5	0.575	0.189
	50	0	0.604	0.187
		0.2	0.602	0.186
		0.5	0.589	0.190
	500	0	0.602	0.185
		0.2	0.606	0.184
		0.5	0.598	0.186

Table 1. Ablation study results. Impact of latent space dimensionality, number of neurons, and dropout rate on C-index and IBS for the METABRIC dataset.

discussed in the manuscript. The results in Table 1 show a generally slight variation in performance across the different runs. Given this consistency, we selected the latent space dimensionality of 5, as it presents the least variability in the results and offers a good trade-off between performance and model complexity. Furthermore, we opted for a moderate number of neurons in the hidden layers (50), as this configuration also demonstrated low variability and yielded competitive values for both C-index and IBS. Notably, the relatively low complexity of these final settings results in faster execution times, enhancing the model's practical usability without sacrificing performance. .

As observed in the METABRIC results, we saw no significant differences regarding the dropout rate. Therefore, we decided to extend the analysis by testing the dropout impact on the dataset with the fewest samples to examine whether dropout can control overfitting in smaller datasets, STD. Table 2 presents the results in the same format as the METABRIC table. Overall, we observe a similar pattern to the METABRIC dataset. The results are relatively consistent across the different combinations of hyperparameters. Notably, using an intermediate latent space dimension (5) consistently results in higher C-index values and lower IBS values, reflecting better model performance. Similarly, the number of neurons in the hidden layers also follows this trend, with intermediate values providing optimal results. While the application of dropout does not significantly impact performance, we obtain slightly better results when included.

Based on these findings, we have chosen the component values specified in the manuscript. However, it is essential to highlight the robustness of our model, as it maintains stable performance across various hyperparameter settings, demonstrating its

Latent Space Dimension	Number of Neurons	Dropout Rate	C-index	IBS
3	10	0	0.575	0.213
		0.2	0.570	0.212
		0.5	0.571	0.213
	50	0	0.576	0.210
		0.2	0.573	0.210
		0.5	0.573	0.212
	500	0	0.573	0.213
		0.2	0.573	0.211
		0.5	0.568	0.211
5	10	0	0.574	0.210
		0.2	0.566	0.212
		0.5	0.584	0.211
	50	0	0.579	0.208
		0.2	0.585	0.209
		0.5	0.571	0.212
	500	0	0.585	0.209
		0.2	0.581	0.212
		0.5	0.576	0.209
50	10	0	0.566	0.213
		0.2	0.570	0.212
		0.5	0.584	0.212
	50	0	0.581	0.209
		0.2	0.577	0.211
		0.5	0.562	0.212
	500	0	0.584	0.208
		0.2	0.588	0.208
		0.5	0.574	0.208

Table 2. Ablation study results. Impact of latent space dimensionality, number of neurons, and dropout rate on C-index and IBS for the STD dataset.

resilience and adaptability. These decisions align with our goal of balancing model complexity and performance, ensuring that the chosen architecture remains computationally efficient while maintaining robust and reliable predictions.

Computational Runtime Comparison

To provide a comprehensive evaluation of the computational complexity of our proposed model, SAVAE, we compared the execution times of training and validation for each model. This comparison includes the entire process, from training the model to validating it on the test set, and it is essential to ensure that both stages are treated as a single process for this purpose. The rationale behind this decision arises from two factors: (1) after every epoch, we compute the performance metrics for the testing set to assess model behavior across all epochs, and (2) the computation of the C-index and Brier Score, the evaluation metrics used in this study, is independent of the model architecture. Hence, the entire process—training and validation—is considered together, ensuring that the steps followed for each model are consistent.

To ensure a fair comparison, we ran every model without Early Stopping mechanisms, allowing each model to train for the full number of epochs. This approach was adopted to ensure that we compared all models under the same training conditions. Although the manuscript results were obtained using Early Stopping with identical patience values across models, different models finished at varying points based on different criteria (e.g., higher metric values or lower loss values). Running the models without this technique enabled for a uniform comparison of training environments.

We ensured the comparability of the results by setting the same number of epochs ($3,000$) and batch size (64) for all models. However, the Early Stop mechanisms employed in the final experiments allowed models to stop at different points depending on validation outcomes, such as metric stabilization or loss minimization. By running the models without early stopping for comparison, we eliminated this variability. To provide a clear comparison, we report the average time taken for a single fold, using a 5-fold cross-validation technique across all models. Although execution times should remain relatively consistent across folds, the average provides a more robust overview.

For SAVAE specifically, given the variability inherent to a VAE model, we ran ten seeds for each fold to ensure robust results. Therefore, the average runtime reported for SAVAE corresponds to a single fold and a single seed. It is important to note that the total runtime for SAVAE will depend on the parallelization pool used, subject to the computational resources available in the environment. Nonetheless, this comparison method provides a fair and consistent approach across different models, ensuring that execution time differences can be attributed solely to model complexity rather than environmental factors.

Dataset	COXPH	DEEPSURV	DEEPHIT	SAVAE
WHAS	165.21	250.07	204.85	294.80
SUPPORT	18.26	24.52	48.58	123.44
GBSG	93.83	141.25	265.49	558.46
FLCHAIN	305.18	442.95	933.32	1880.55
NWTCO	236.73	325.04	638.76	922.92
METABRIC	76.52	110.50	225.07	1058.42
PBC	28.14	38.58	118.53	198.13
STD	40.02	58.89	115.63	397.00
PNEUMON	188.09	273.14	382.50	1860.52
Average	128.00	184.99	325.86	810.47

Table 3. Average execution times (in seconds) for training and validating different models across various datasets, including COXPH, DeepSurv, DeepHit, and SAVAE. The total time reflects the duration of a single fold in a 5-fold cross-validation setting.

Table 3 presents the average execution times in seconds for each model, providing insight into their computational demands. These times reflect the total duration of training and validating a model on a single fold. The execution times for SAVAE are generally higher compared to other models. This is expected due to the inherent complexity of our architecture, which includes multiple components, such as the VAE latent space and additional decoders. These elements increase the computational load, especially when modeling complex non-linear relationships in survival data. However, longer runtimes are justified by the model's improved flexibility and capacity to handle more intricate data patterns. Despite the increased complexity, the

execution times remain within reasonable bounds and are not excessively large for any dataset. In fact, the execution times for SAVAE are manageable, as in all cases, we are dealing with runtimes that last only a few minutes. This balance ensures that SAVAE's enhanced performance does not come at the cost of prohibitive computational demands, making it practical for real-world applications.

Adjustment of *p*-values for Multiple Hypothesis Testing

To enhance the statistical rigor of our performance evaluation, we conducted multiple hypothesis tests to compare our model's mean C-index and IBS values, SAVAE, with those of state-of-the-art models across multiple folds in a five-fold cross-validation setup. Specifically, we tested the null hypothesis that the mean performance metrics of the benchmark models are superior to those of SAVAE. We used *p*-values to assess the statistical significance of our findings, setting a conventional significance threshold of 0.05. However, multiple hypothesis tests increase the Family-Wise Error Rate (FWER), which is the probability of making one or more Type I errors (false positives) among all the hypotheses when the null hypotheses are true. According to the statistical literature[5–7], as the number of tests increases, the FWER also increases, thereby inflating the likelihood of incorrectly rejecting true null hypotheses.

To address this issue and control the FWER, we applied the Holm-Bonferroni method[8], a step-down procedure that adjusts the *p*-values to maintain the overall significance level. The Holm method is more powerful than the traditional Bonferroni correction[9] and does not assume independence among tests, making it suitable for our analysis. We implemented this adjustment using the *statsmodels* package in Python, which provides robust tools for multiple testing corrections.

The adjusted *p*-values for both the C-index and IBS metrics are presented in Tables 4 and 5, respectively. These tables compare the original and Holm-adjusted *p*-values for each dataset. By adjusting the *p*-values, we aim to control for multiple hypothesis testing and avoid inflation of the FWER. Each table displays the results as pairs of *original p-value - adjusted p-value*, where the Holm adjustment method has been applied to account for the number of comparisons performed across the models for each dataset. The comparison enables us to observe whether the adjusted *p*-values differ significantly from the original values and whether the adjustments affect the statistical significance of the results. As expected, the adjusted *p*-values for each metric increase in response to the correction applied to account for multiple tests. However, these adjustments do not alter the overall conclusions drawn from the statistical analysis. This consistency suggests that the findings remain valid even after accounting for the potential inflation of Type I errors. As indicated by the bold entries in the tables, several instances show *p*-values below the threshold of 0.05, reinforcing that SAVAE significantly outperforms the state-of-the-art models in these cases.

Model	WHAS	SUPPORT	GBSG	FLCHAIN	NWTCO	METABRIC	PBC	STD	PNEUMON
COXPH	0.579 - 1.000	0.058 - 0.058	**0.000 - 0.000**	**0.000 - 0.000**	0.268 - 0.536	**0.003 - 0.006**	0.450 - 0.683	0.887 - 1.000	**0.003 - 0.009**
DEEPSURV	1.000 - 1.000	**0.020 - 0.040**	0.149 - 0.149	**0.000 - 0.000**	0.135 - 0.406	0.549 - 0.549	0.280 - 0.683	0.927 - 1.000	0.382 - 0.765
DEEPHIT	1.000 - 1.000	**0.000 - 0.001**	**0.000 - 0.000**	**0.001 - 0.001**	0.644 - 0.645	**0.000 - 0.000**	0.228 - 0.683	0.727 - 1.000	0.935 - 0.935

Bold Implies a *p*-value below our threshold, 0.05. This means that SAVAE is significantly better than the other models.

Table 4. Comparison of original and Holm-adjusted *p*-values to evaluate whether the mean C-index of SAVAE exceeds those of state-of-the-art models across cross-validation folds. The results are presented as *original p-value - adjusted p-value*.

Model	WHAS	SUPPORT	GBSG	FLCHAIN	NWTCO	METABRIC	PBC	STD	PNEUMON
COXPH	1.000 - 1.000	0.470 - 1.000	0.998 - 1.000	1.000 - 1.000	**0.000 - 0.000**	0.995 - 1.000	0.888 - 1.000	0.575 - 1.000	**0.000 - 0.000**
DEEPSURV	**0.000 - 0.000**	0.341 - 1.000	0.561 - 1.000	1.000 - 1.000	**0.000 - 0.000**	1.000 - 1.000	0.868 - 1.000	0.746 - 1.000	**0.000 - 0.000**
DEEPHIT	**0.000 - 0.000**	0.950 - 1.000	1.000 - 1.000	1.000 - 1.000	**0.000 - 0.000**	1.000 - 1.000	1.000 - 1.000	0.995 - 1.000	**0.000 - 0.000**

Bold Implies a *p*-value below our threshold, 0.05. This means that SAVAE is significantly better than the other models.

Table 5. Comparison of original and Holm-adjusted *p*-values assessing whether the mean IBS values of SAVAE surpass those of state-of-the-art models across cross-validation folds. The results are presented as *original p-value - adjusted p-value*.

Our findings demonstrate that applying Holm's method remains the same conclusions drawn from the original *p*-values. In particular, the results confirm that SAVAE offers superior performance to existing models in several datasets, as evidenced by consistently low *p*-values. This further highlights the effectiveness of our model and its ability to generalize well across different datasets.

References

1. Lundberg, S. M. & Lee, S.-I. A unified approach to interpreting model predictions. NIPS'17 (Curran Associates Inc., Red Hook, NY, USA, 2017).
2. Bechis, S. K., Carroll, P. R. & Cooperberg, M. R. Impact of age at diagnosis on prostate cancer treatment and survival. *J. Clin. Oncol.* **29**, 235–241 (2011).
3. Cheung, Y. B., Gao, F. & Khoo, K. S. Age at diagnosis and the choice of survival analysis methods in cancer epidemiology. *J. clinical epidemiology* **56**, 38–43 (2003).
4. van Eeghen, E. E., Bakker, S. D., van Bochove, A. & Loffeld, R. J. Impact of age and comorbidity on survival in colorectal cancer. *J. gastrointestinal oncology* **6**, 605 (2015).
5. Tukey, J. W. The philosophy of multiple comparisons. *Stat. science* 100–116 (1991).
6. Lehmann, E. L. & Romano, J. P. *Generalizations of the familywise error rate* (Springer, 2012).
7. Van der Laan, M. J., Dudoit, S. & Pollard, K. S. Multiple testing. part ii. step-down procedures for control of the family-wise error rate. *Stat. applications genetics molecular biology* **3** (2004).
8. Holm, S. A simple sequentially rejective multiple test procedure. *Scand. journal statistics* 65–70 (1979).
9. Bonferroni, C. Teoria statistica delle classi e calcolo delle probabilita. *Pubblicazioni del R istituto superiore di scienze economiche e commericiali di firenze* **8**, 3–62 (1936).