Maximum Likelihood Estimation with Deep Learning for Multiple Sclerosis Progression Prediction

Notebook for the iDPP Lab on Intelligent Disease Progression Prediction at CLEF 2023

Tsvetan Asamov[1], Petar Ivanov[1], Anna Aksenova[1], Dimitar Taskov[2,3] and Svetla Boytcheva[1]

[1]*Ontotext, Bulgaria*
[2]*Medical University - Sofia, Bulgaria*
[3]*Multiprofile Hospital for Active Treatment in Neurology and Psychiatry "St. Naum" - Sofia, Bulgaria*

Abstract

We develop a maximum likelihood estimation approach for intelligent disease progression prediction. We use patients' covariates and employ a multi-layer perceptron to approximate the optimal distribution parameters for a given parametric family of probability distributions. As far as we know, this is the first time such a method has been applied to real multiple sclerosis data. Our numerical results indicate that the method can achieve AUROC scores exceeding 0.8.

Keywords

CLEF, multiple sclerosis, neurological disease, maximum likelihood estimation, deep learning

1. Introduction

Multiple sclerosis (MS) is a chronic autoimmune disease with a genetic predisposition which, in combination with environmental factors, leads to inflammatory demyelination of the white matter of the central nervous system (CNS). The majority of patients with multiple sclerosis begin with a relapsing-remitting (RR) course. Over time, however, most progress to secondary progressive MS (SPMS), which is characterized by a gradual and progressive worsening of the disease. MS is the second most common cause of disability in young adults. Identifying prognostic factors for disease progression early in its course is critical for evaluating possible therapeutic interventions.

This paper presents a maximum likelihood estimation approach for predicting the progression of multiple sclerosis. The work is part of the iDPP challenge at CLEF 2023. The iDPP challenge [1] includes (but is not limited to) the following two tasks:

- Task 1: Predicting the risk of disease worsening - predicting the risk of worsening and ranking subjects based on the risk scores. More specifically, the risk of worsening should

CLEF 2023: Conference and Labs of the Evaluation Forum, September 18–21, 2023, Thessaloniki, Greece
tsvetan.asamov@ontotext.com (T. Asamov); petar.ivanov@ontotext.com (P. Ivanov); anna.aksenova@ontotext.com (A. Aksenova); dimtaskov@gmail.com (D. Taskov); svetla.boytcheva@ontotext.com (S. Boytcheva)
0000-0002-7556-1350 (T. Asamov); 0000-0001-8448-1005 (P. Ivanov); 0000-0002-3489-874X (A. Aksenova); 0000-0002-0939-5382 (D. Taskov); 0000-0002-5542-9168 (S. Boytcheva)

be a value between 0 and 1 that reflects how early a patient experiences the worsening event.

- Task 2: Predicting the cumulative probability of worsening - assigning cumulative probability of worsening at different time windows, i.e. between years 0 and 2, 0 and 4, 0 and 6, 0 and 8, 0 and 10.

In addition, for each task, we consider two different subtasks based on two alternative definitions of worsening. Following clinical standards, worsening is defined on the basis of the Expanded Disability Status Scale (EDSS):

- Subtask A: the patient crosses the EDSS ≥ 3 threshold at least twice within a one-year interval.
- Subtask B: the first recorded EDSS value available in clinical records is defined as the baseline, and worsening occurs according to the following rules:
  - if the baseline is EDSS < 1, then worsening occurs when an EDSS increase of 1.5 points is first observed.
  - if the baseline is $1 \leq$ EDSS < 5.5, then worsening occurs when an EDSS increase of 1 point is first observed.
  - if the baseline is EDSS ≥ 5.5, then worsening occurs when an EDSS increase of 0.5 points is first observed.

Finally, for each subtask, we are given a separate dataset consisting of general patient information, as well as a series of observations over 2.5 years.

The paper is organized as follows: Section 2 introduces related works; Section 3 describes our approach; Section 4 explains our experimental setup; Section 5 discusses our main findings; finally, Section 6 draws some conclusions and outlooks for future work.

2. Related Work

Applications in various domains involve time-to-event modelling problems in the presence of censoring. Some examples include healthcare [2, 3], reliability [4, 5], finance [6], and other fields.

The Cox proportional hazards model [7] is commonly used in survival analysis. However, it employs a log-linear function to predict the outcome variable from the covariates. In this sense, it may not be suitable to properly predict a multiple sclerosis patient outcome without extensive feature engineering.

Recently, deep learning has been applied to extend the Cox proportional hazard model. More specifically, the DeepSurv model [8] employs a multi-layered perceptron to replace the log-linear function of the Cox proportional hazards model. Similar to the Cox proportional hazard model, DeepSurv assumes a constant baseline hazard.

In addition, DeepHit [9] has proposed discretizing the space of event times, and using a deep neural network to learn the distribution of survival times. Further, DeepHit does not make any assumptions about the underlying stochastic process, and has the ability to handle competing risks.

Random survival forests [10] is a non-parametric method that constitutes an extension of the random forest method approach [11]. More specifically, random survival forests learn an ensemble of trees for the analysis of right-censored survival data.

The idea of using a deep learning framework to estimate probability distribution parameters for maximum likelihood estimation for right-censored data was initially introduced by Nagpal et al. [12] as a part of their deep survival machines (DSM) framework. DSM employs a mixture of individual parametric survival distributions to fit a set of right-censored survival data. The work was further extended to recurrent deep survival machines (RDSM) by Nagpal et al. [13] with the introduction of recurrent neural networks in place of the learnt representations. Unlike DSM and RDSM, we do not use a mixture of parametric distributions but rather focus on fitting the parameters of a single parametric probability distribution.

3. Methodology

In this section we describe the methodology we have adopted.

3.1. Data

We assume that we are given right-censored data which consists of a set of I triples $\{(\mathbf{x}_i, t_i, \delta_i)\}_{i=1}^{I}$. For each $i = 1, \ldots, I$, the real vector $\mathbf{x}_i \in \mathbb{R}^d$ denotes the features of the $i-$th entry. Further, t_i is either the censoring time, or the time an event occurred. In addition, δ_i is an indicator variable taking a value of 0 if t_i is the censoring time, and a value of 1, if t_i is the time at which an event took place. It is assumed that for each $i = 1, \ldots, I$ either censoring occurs, or we observe the event but not both.

3.2. Maximum Likelihood Formulation

The method of maximum likelihood can be adapted to various applied problems. In this section, we develop a maximum likelihood estimation approach for intelligent disease progression prediction. Given a set of right-censored patient data $\{(\mathbf{x}_i, t_i, \delta_i)\}_{i=1}^{I}$, we can assume independence among patients, and thus define the likelihood function of the observed data as follows:

$$L(\theta) = \prod_{i,\delta_i=1} f(t_i|\theta_i) \prod_{i,\delta_i=0} (1 - F(t_i|\theta_i)) \tag{1}$$

where

- $f(t_i|\theta_i)$ is the probability density function evaluated at time t_i for distribution parameters θ_i.
- $F(t_i|\theta_i)$ is the cumulative probability density function evaluated at time t_i for distribution parameters θ_i.

Please note that we do not assume that the patient data is identically distributed. On the contrary, we consider different distribution parameters θ_i for each patient $i = 1, \ldots, I$. Further, please note that if we knew the distribution functions f and F, and if we could estimate the

distribution parameters θ_i for a previously unseen patient i, then we would also be able to estimate the patient's probability of worsening over a given time period. In order to achieve that, we would like to use a parametric family of distributions in the above formulation (1). Hence, we would need to choose a probability distribution with support over the positive real line. In this work, we focus on the Weibull distribution [14]. It is a continuous probability distribution that has closed form expressions for both its probability density function $f(t)$, and cumulative density function $F(t)$:

$$\begin{aligned} f(t|\alpha,\beta) &:= \begin{cases} 0, & t<0 \\ \frac{\beta}{\alpha}\left(\frac{t}{\alpha}\right)^{\beta-1}\exp\left(-\left(\frac{t}{\alpha}\right)^{\beta}\right), & t\geq 0 \end{cases} \\ F(t|\alpha,\beta) &:= \begin{cases} 0, & t<0 \\ 1-\exp\left(-\left(\frac{t}{\alpha}\right)^{\beta}\right), & t\geq 0 \end{cases} \end{aligned} \tag{2}$$

Thus, we can seamlessly apply automatic differentiation in the gradient-based search for the optimal distribution parameters. This allows us to use a feed-forward neural network with fully connected layers as the function mapping feature inputs $\mathbf{x}_i$ to estimated distribution parameters $\theta_i = (\alpha_i, \beta_i)$:

$$\Psi(\mathbf{A},\mathbf{b},\mathbf{x}) := \sigma(\mathbf{A}_N\sigma(\mathbf{A}_{N-1}\sigma(\dots\mathbf{A}_3\sigma(\mathbf{A}_2\sigma(\mathbf{A}_1\mathbf{x}+\mathbf{b}_1)+\mathbf{b}_2)+\mathbf{b}_3\dots)+\mathbf{b}_{N-1})+\mathbf{b}_N) \tag{3}$$

where $\mathbf{A}$ and $\mathbf{b}$ are collection of respectively real matrices $\mathbf{A}_n,\ n=1,\dots,N$, and real vectors $\mathbf{b}_n,\ n=1,\dots,N$, and σ is an activation function such as the rectified linear unit. Thus, we can write the problem of maximizing the likelihood function as follows:

$$\max_{\mathbf{A},\mathbf{b}} \prod_{i,\delta_i=1} f(t_i|\alpha_i,\beta_i) \prod_{i,\delta_i=0} (1-F(t_i|\alpha_i,\beta_i))$$

s.t.

$$\begin{aligned} &(\alpha_i,\beta_i) = \Psi(\mathbf{A},\mathbf{b},\mathbf{x}_i),\ i=1,\dots,I \\ &\alpha_i > 0,\ i=1,\dots,I \\ &\beta_i > 0,\ i=1,\dots,I \end{aligned}$$

where

$$\begin{aligned} f(t|\alpha,\beta) &:= \begin{cases} 0, & t<0 \\ \frac{\beta}{\alpha}\left(\frac{t}{\alpha}\right)^{\beta-1}\exp\left(-\left(\frac{t}{\alpha}\right)^{\beta}\right), & t\geq 0 \end{cases} \\ F(t|\alpha,\beta) &:= \begin{cases} 0, & t<0 \\ 1-\exp\left(-\left(\frac{t}{\alpha}\right)^{\beta}\right), & t\geq 0 \end{cases} \\ \Psi(\mathbf{A},\mathbf{b},\mathbf{x}) &:= \sigma(\mathbf{A}_n\sigma(\mathbf{A}_{n-1}\sigma(\dots\mathbf{A}_3\sigma(\mathbf{A}_2\sigma(\mathbf{A}_1\mathbf{x}+\mathbf{b}_1)+\mathbf{b}_2)+\mathbf{b}_3\dots)+\mathbf{b}_{n-1})+\mathbf{b}_n) \end{aligned} \tag{4}$$

In order to improve computational stability and avoid numerical issues, we choose to instead maximize the log-likelihood function. As the logarithm function is monotonic, we can

equivalently write problem (4) as follows:

$$\max_{\mathbf{A},\mathbf{b}} \sum_{i,\delta_i=1} \log(f(t_i|\alpha_i,\beta_i)) + \sum_{i,\delta_i=0} \log(1-F(t_i|\alpha_i,\beta_i))$$

s.t.

$$(\alpha_i,\beta_i) = \Psi(\mathbf{A},\mathbf{b},\mathbf{x}_i),\ i=1,\dots,I$$
$$\alpha_i > 0,\ i=1,\dots,I$$
$$\beta_i > 0,\ i=1,\dots,I$$

where

$$f(t|\alpha,\beta) := \begin{cases} 0, & t<0 \\ \frac{\beta}{\alpha}\left(\frac{t}{\alpha}\right)^{\beta-1}\exp\left(-\left(\frac{t}{\alpha}\right)^{\beta}\right), & t\geq 0 \end{cases}$$

$$F(t|\alpha,\beta) := \begin{cases} 0, & t<0 \\ 1-\exp\left(-\left(\frac{t}{\alpha}\right)^{\beta}\right), & t\geq 0 \end{cases}$$

$$\Psi(\mathbf{A},\mathbf{b},\mathbf{x}) := \sigma(\mathbf{A}_n\sigma(\mathbf{A}_{n-1}\sigma(\dots\mathbf{A}_3\sigma(\mathbf{A}_2\sigma(\mathbf{A}_1\mathbf{x}+\mathbf{b}_1)+\mathbf{b}_2)+\mathbf{b}_3\dots)+\mathbf{b}_{n-1})+\mathbf{b}_n) \tag{5}$$

Please note that even in the case when the activation function σ is the identity map, the proposed formulation is neither convex, nor concave. Thus, in general, we cannot find a global optimal solution to the proposed model using gradient-type methods. However, sub-optimal solutions of reasonable quality can be found, as indicated in the next sections.

4. Experimental Setup

In this section, we describe our experimental setup.

4.1. Implementation

The training pipeline is implemented in the julia programming language [15]. Further, we use the Knet library [16] for our neural network implementation. Our code is available on bitbucket. We run training and testing steps on a Core i7 CPU with 16 GB of RAM. What is more, we use a validation set to determine the values of our hyper-parameters. Moreover, we use the Adam optimizer [17] with a learning rate of 0.00001. While the value may seem somewhat small, we found it to work well in practice. In addition, we do not split the data into batches but rather use the entire training dataset for each step of Adam. Furthermore, we apply dropout regularization [18, 19] to the input layer with a dropout rate chosen among 0.01 and 0.2. We choose the number of hidden units in the neural network among 100 and 200. Finally, the number of training epochs (which also equals the number of training steps of Adam) is chosen among 5000 and 10000.

4.2. Datasets

We are given two different datasets, one for subtask A, and one for subtask B. In this way, task 1A and task 2A both use the first dataset, while task 1B and task 2B both use the second dataset. Each dataset contains 2.5 years of patient visits. In addition, the occurrence of the worsening event, as well as the time of its occurrence are also given.

The training and testing data of both datasets (subtask A and subtask B) is partitioned into static patient data and dynamic patient data. Furthermore, the dynamic patient data includes information on relapses, EDSS scores, evoked potentials, MRI results and multiple sclerosis course.

The training dataset for subtask A includes the following: 441 patients, 481 relapses, 2,661 EDSS scores, 1,211 evoked potentials, 960 MRIs, and 310 multiple sclerosis courses. The training dataset for subtask B includes the following: 511 patients, 553 relapses, 3,069 EDSS scores, 1,522 evoked potentials, 966 MRIs, and 325 multiple sclerosis courses. In addition, the testing dataset of subtask A includes the following: 111 patients, 95 relapses, 675 EDSS scores, 278 evoked potentials, 236 MRIs, and 68 multiple sclerosis courses. And the testing dataset of subtask B includes the following: 129 patients, 125 relapses, 813 EDSS scores, 299 evoked potentials, 266 MRIs, and 75 multiple sclerosis courses. For a detailed description of the datasets and the evaluation measures, please see the overview papers by the CLEF challenge organizers [20, 21].

5. Results

The challenge objectives consist of the following:

- Task 1 - predicting the risk of disease worsening
- Task 2 - predicting the cumulative probability of worsening

In order to handle both tasks, we use the available training data to build a model and estimate a maximum likelihood distribution for each patient given the patient's covariates (features). Ideally, for task 1 we would have preferred to use coherent risk measures [22, 23] to estimate the risk of disease worsening from the patients' distributions. However, in order to meet the requirement that risk values are in the range of $[0, 1]$ we decide to use a cumulative probability estimate instead of coherent risk measures. The performance of the submitted models is reported in Figures 1-11. Please note that the name of each model indicates the model's parameters' values. For instance, the first model in Figure 1 is named "T1b.0.2.1.0e-5.5000.200", indicating that it is a Task 1B model with the following parameters:

- A dropout rate of 0.2 used in the input layer.
- A learning rate of 1.0e-5 used by the Adam optimizer.
- The model is trained for 5000 epochs, i.e. the Adam optimizer performs 5000 steps.
- The number of hidden units is set to 200.

We can see that the highest Harrell's concordance index values fall in the interval $[0.6, 0.65]$. Ideally, we would like to improve those results in the future. One way we could do that is by incorporating event ordering into the model training procedure. Another approach we could

try is scaling down classical coherent risk measures to fit into the $[0, 1]$ interval for the given patient data. In Figures 2-6 we can see that the highest AUROC exceeds 0.8. In the future, we can further improve those results by better model optimization. In addition, in Figures 7-11 we can find the ratio of observed to expected events for all submitted models.

The model with the highest AUROC score T2a.0.01.1.0e-5.10000.100.adj has a couple of aspects that distinguish it from the rest of the models. First, for each patient dataframe (static patient data, relapses, EDSS scores, evoked potentials, MRI results and multiple sclerosis course) it explicitly takes into account the length of the dataframe. And second, it normalizes the age_at_onset variable using division by fifty. This suggests that current results can be further improved by additional data pre-processing.

Finally, in Table 1 we present illustrations of probability density functions for three randomly chosen patients from the test set for the 2a.0.01.1.0e-5.10000.100.adj model. Please note that the risk and the probabilities of worsening for each patient depend entirely on the computed values of the distribution parameters α_i and β_i.

6. Conclusion and Future Work

The development of predictive models of the disease is a step forward towards better clinical assessment and an individualized therapeutic approach for multiple sclerosis patients.

In this paper we have developed a maximum likelihood estimation approach for intelligent disease progression prediction. To the best of our knowledge, this is the first instance of such a method being applied to real multiple sclerosis data. Our numerical results indicate that the method can achieve AUROC scores exceeding 0.8. In the future we can explore several directions of further research. First, we can attempt to incorporate event ordering into the training procedure in order to improve Harrell's concordance index scores. Further, we can attempt to apply (scaled-down) coherent risk measures in order to obtain risk estimates in the [0, 1] interval. Finally, we may also look into improving the quality of the numerical solution with the use of second-order optimization methods such as K-FAC [24] or L-BFGS [25].

References

[1] Participation guidelines of idpp@ clef 2023, 2023. URL: https://brainteaser.dei.unipd.it/challenges/idpp2023/.

[2] D. W. Kim, S. Lee, S. Kwon, W. Nam, I.-H. Cha, H. J. Kim, Deep learning-based survival prediction of oral cancer patients, Scientific reports 9 (2019) 1–10.

[3] A. M. Jones, O. O'Donnell, O. O'Donnell, Econometric analysis of health data, Wiley Online Library, 2002.

[4] R. E. Barlow, F. Proschan, Statistical theory of reliability and life testing: probability models, Technical Report, Florida State Univ Tallahassee, 1975.

[5] M. Hollander, E. A. Peña, Dynamic reliability models with conditional proportional hazards, Lifetime Data Analysis 1 (1995) 377–401.

[6] M. Stepanova, L. Thomas, Survival analysis methods for personal loan data, Operations Research 50 (2002) 277–289.

[7] D. R. Cox, Regression models and life-tables, Journal of the Royal Statistical Society: Series B (Methodological) 34 (1972) 187–202.
[8] J. L. Katzman, U. Shaham, A. Cloninger, J. Bates, T. Jiang, Y. Kluger, Deepsurv: personalized treatment recommender system using a cox proportional hazards deep neural network, BMC medical research methodology 18 (2018) 1–12.
[9] C. Lee, W. Zame, J. Yoon, M. Van Der Schaar, Deephit: A deep learning approach to survival analysis with competing risks, in: Proceedings of the AAAI conference on artificial intelligence, volume 32, 2018.
[10] H. Ishwaran, U. B. Kogalur, E. H. Blackstone, M. S. Lauer, Random survival forests (2008).
[11] L. Breiman, Random forests, Machine learning 45 (2001) 5–32.
[12] C. Nagpal, X. Li, A. Dubrawski, Deep survival machines: Fully parametric survival regression and representation learning for censored data with competing risks, IEEE Journal of Biomedical and Health Informatics 25 (2021) 3163–3175.
[13] C. Nagpal, V. Jeanselme, A. Dubrawski, Deep parametric time-to-event regression with time-varying covariates, in: Survival Prediction-Algorithms, Challenges and Applications, PMLR, 2021, pp. 184–193.
[14] A. Kızılersü, M. Kreer, A. W. Thomas, The weibull distribution, 2018.
[15] J. Bezanson, S. Karpinski, V. B. Shah, A. Edelman, Julia: A fast dynamic language for technical computing, arXiv preprint arXiv:1209.5145 (2012).
[16] D. Yuret, Knet: beginning deep learning with 100 lines of julia, in: Machine Learning Systems Workshop at NIPS, volume 2016, 2016, p. 5.
[17] D. P. Kingma, J. Ba, Adam: A method for stochastic optimization, arXiv preprint arXiv:1412.6980 (2014).
[18] G. E. Hinton, N. Srivastava, A. Krizhevsky, I. Sutskever, R. R. Salakhutdinov, Improving neural networks by preventing co-adaptation of feature detectors, arXiv preprint arXiv:1207.0580 (2012).
[19] P. Baldi, P. J. Sadowski, Understanding dropout, Advances in neural information processing systems 26 (2013).
[20] G. Faggioli, A. Guazzo, S. Marchesin, L. Menotti, I. Trescato, H. Aidos, R. Bergamaschi, G. Birolo, P. Cavalla, A. Chiò, A. Dagliati, M. de Carvalho, G. M. Di Nunzio, P. Fariselli, J. M. García Dominguez, M. Gromicho, E. Longato, S. C. Madeira, U. Manera, G. Silvello, E. Tavazzi, E. Tavazzi, M. Vettoretti, B. Di Camillo, N. Ferro, Intelligent Disease Progression Prediction: Overview of iDPP@CLEF 2023, in: A. Arampatzis, E. Kanoulas, T. Tsikrika, S. Vrochidis, A. Giachanou, D. Li, A. Aliannejadi, M. Vlachos, G. Faggioli, N. Ferro (Eds.), Experimental IR Meets Multilinguality, Multimodality, and Interaction. Proceedings of the Fourteenth International Conference of the CLEF Association (CLEF 2023), Lecture Notes in Computer Science (LNCS), Springer, Heidelberg, Germany, 2023.
[21] G. Faggioli, A. Guazzo, S. Marchesin, L. Menotti, I. Trescato, H. Aidos, R. Bergamaschi, G. Birolo, P. Cavalla, A. Chiò, A. Dagliati, M. de Carvalho, G. M. Di Nunzio, P. Fariselli, J. M. García Dominguez, M. Gromicho, E. Longato, S. C. Madeira, U. Manera, G. Silvello, E. Tavazzi, E. Tavazzi, M. Vettoretti, B. Di Camillo, N. Ferro, Overview of iDPP@CLEF 2023: The Intelligent Disease Progression Prediction Challenge, in: M. Aliannejadi, G. Faggioli, N. Ferro, M. Vlachos (Eds.), CLEF 2023 Working Notes, CEUR Workshop Proceedings (CEUR-WS.org), ISSN 1613-0073., 2023.

[22] P. Artzner, F. Delbaen, J.-M. Eber, D. Heath, Coherent measures of risk, Mathematical finance 9 (1999) 203–228.
[23] T. Asamov, A. Ruszczyński, Time-consistent approximations of risk-averse multistage stochastic optimization problems, Mathematical Programming 153 (2015) 459–493.
[24] J. Martens, R. Grosse, Optimizing neural networks with kronecker-factored approximate curvature, in: International conference on machine learning, PMLR, 2015, pp. 2408–2417.
[25] J. NOCEDAL, J. W. STEPHEN, SPRINGER SERIES IN OPERATONS RESEARCH NUMERICAL OPTIMIZATION., Springer, 2006.

A. Numerical Results

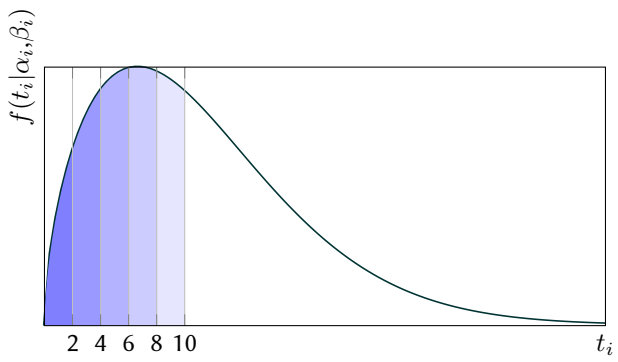
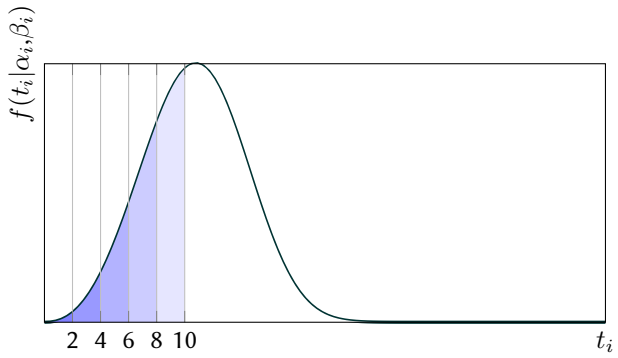
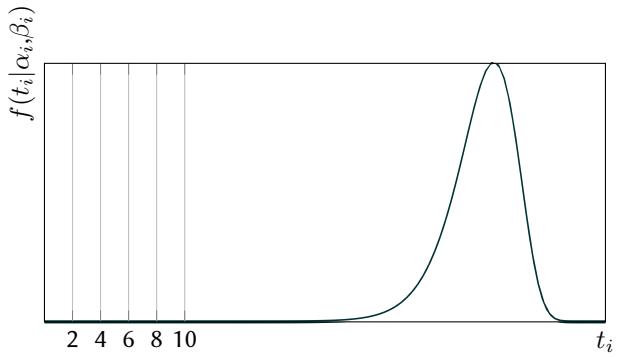
α_i	β_i	Probability Density Function
12.4033	1.58571	
11.9824	3.3673	
32.1712	16.4169	

Table 1
Probability density functions for three randomly selected patients from the testing set for the T2a.0.01.1.0e-5.10000.100.adj model.

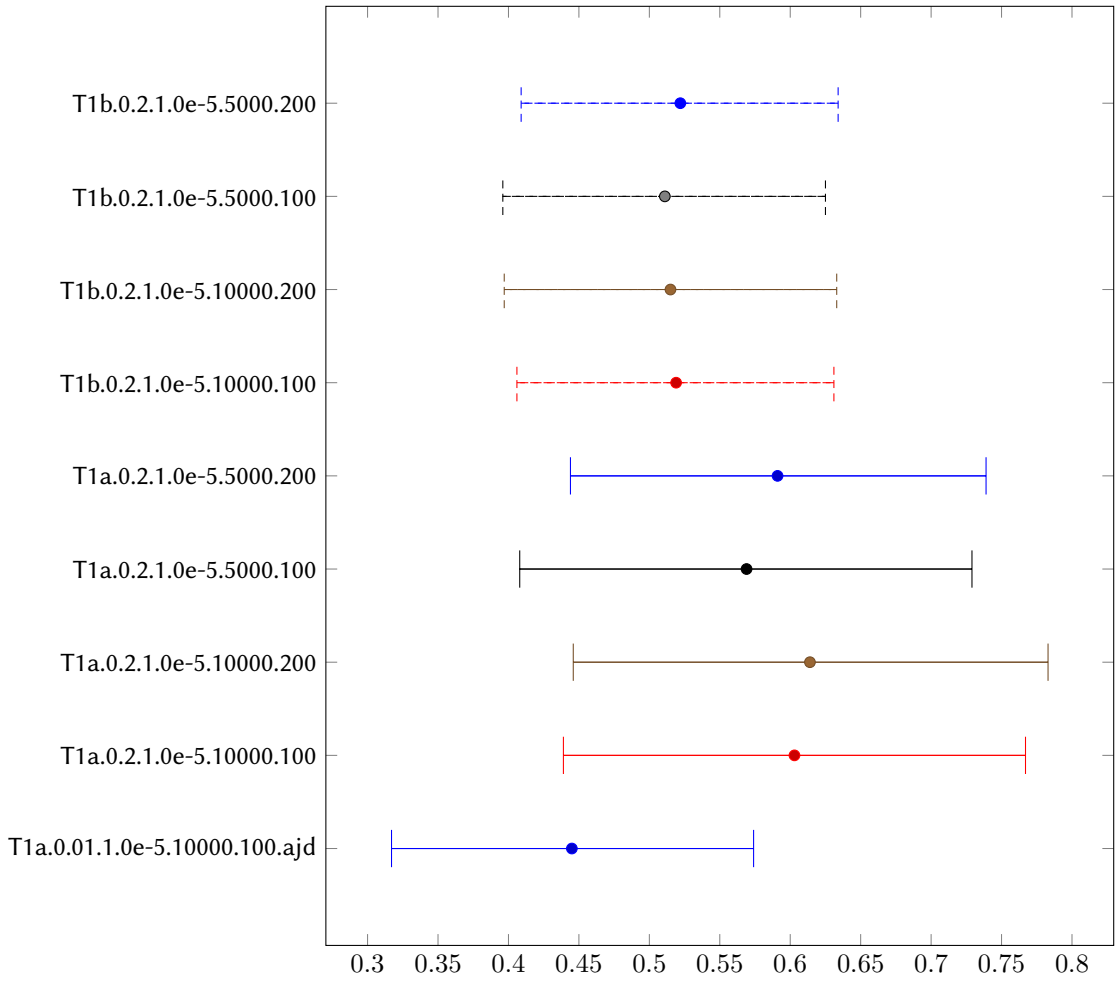


Figure 1: Task 1 - Harrell's Concordance Index computed for all submitted runs. The bars in the plot show the 95% confidence interval.

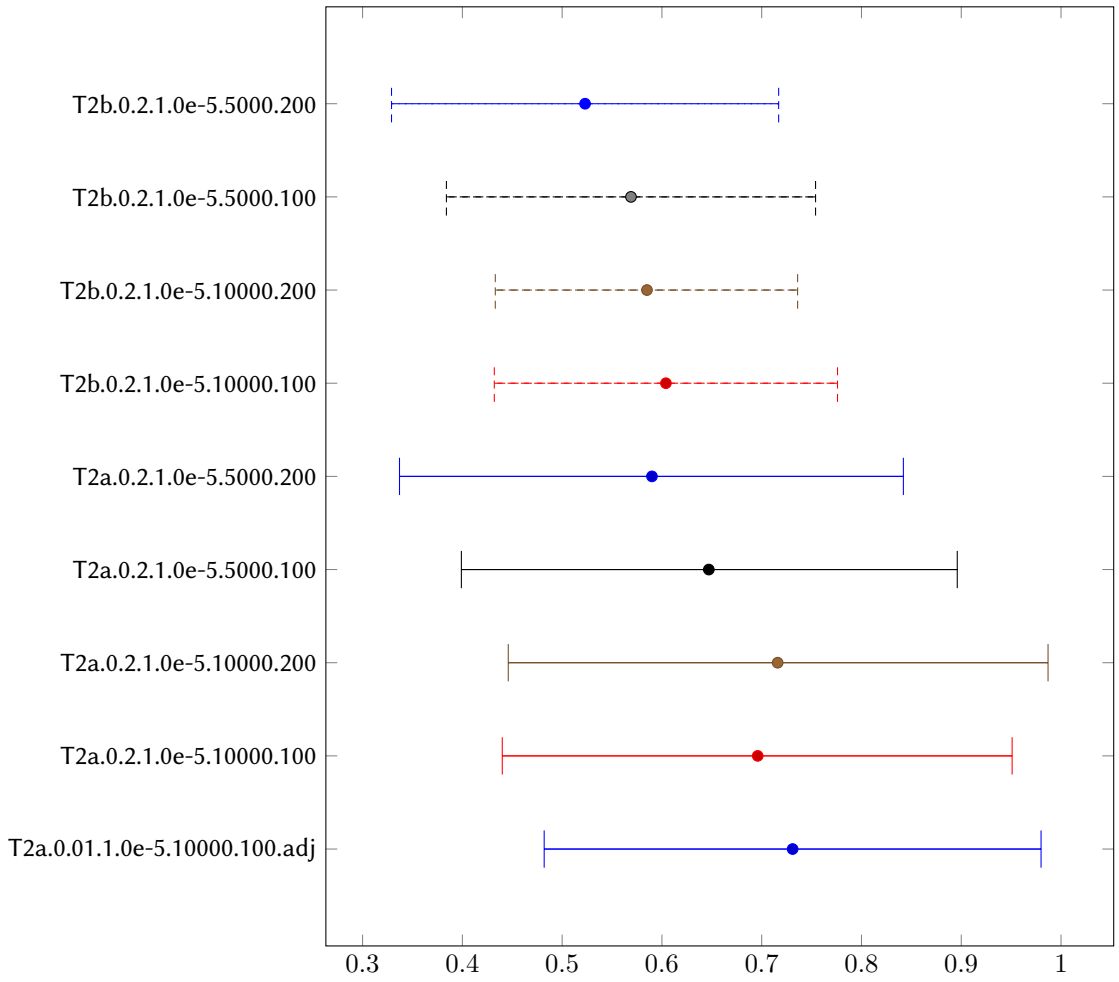


Figure 2: Task 2 - AUROC computed for all submitted runs with a 2-year time horizon. The bars in the plot show the 95% confidence interval.

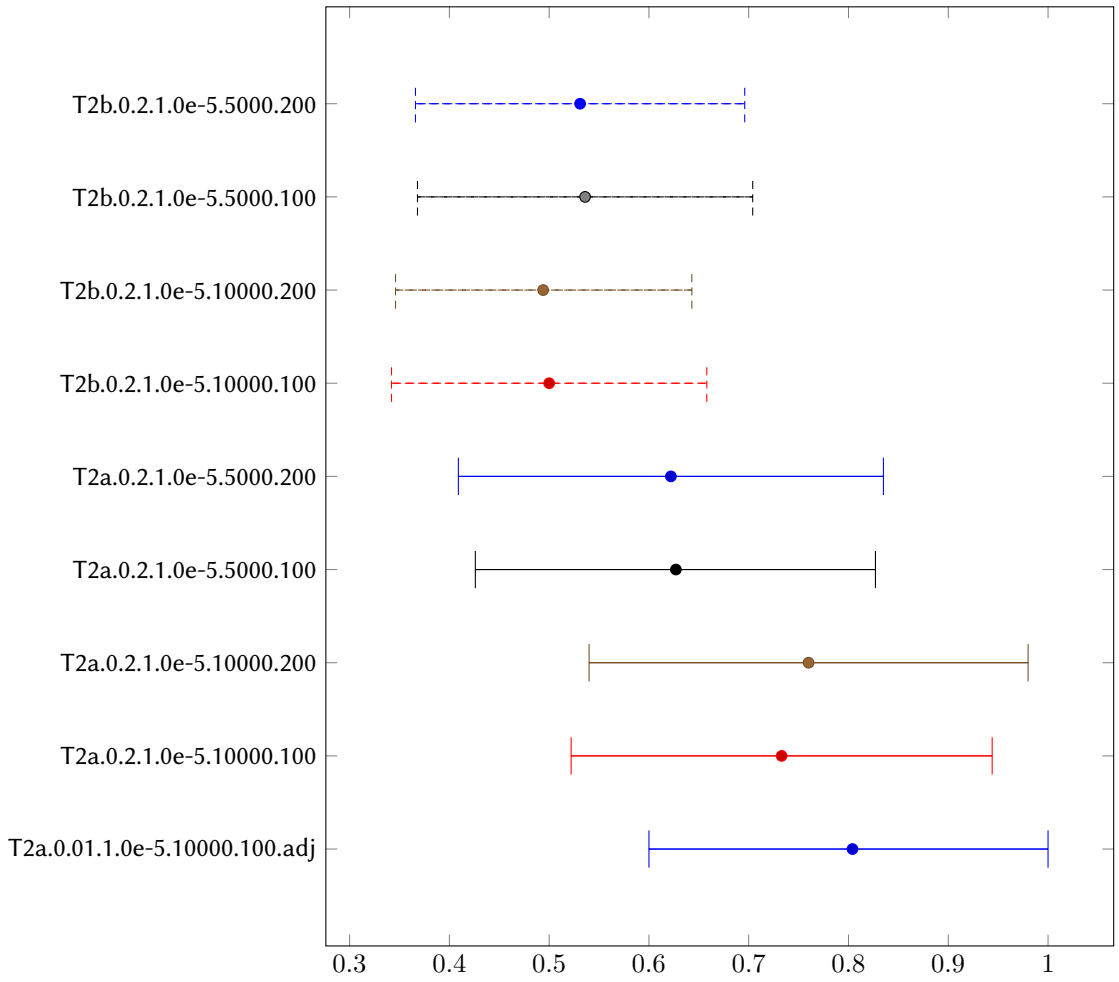


Figure 3: Task 2 - AUROC computed for all submitted runs with a 4-year time horizon. The bars in the plot show the 95% confidence interval.

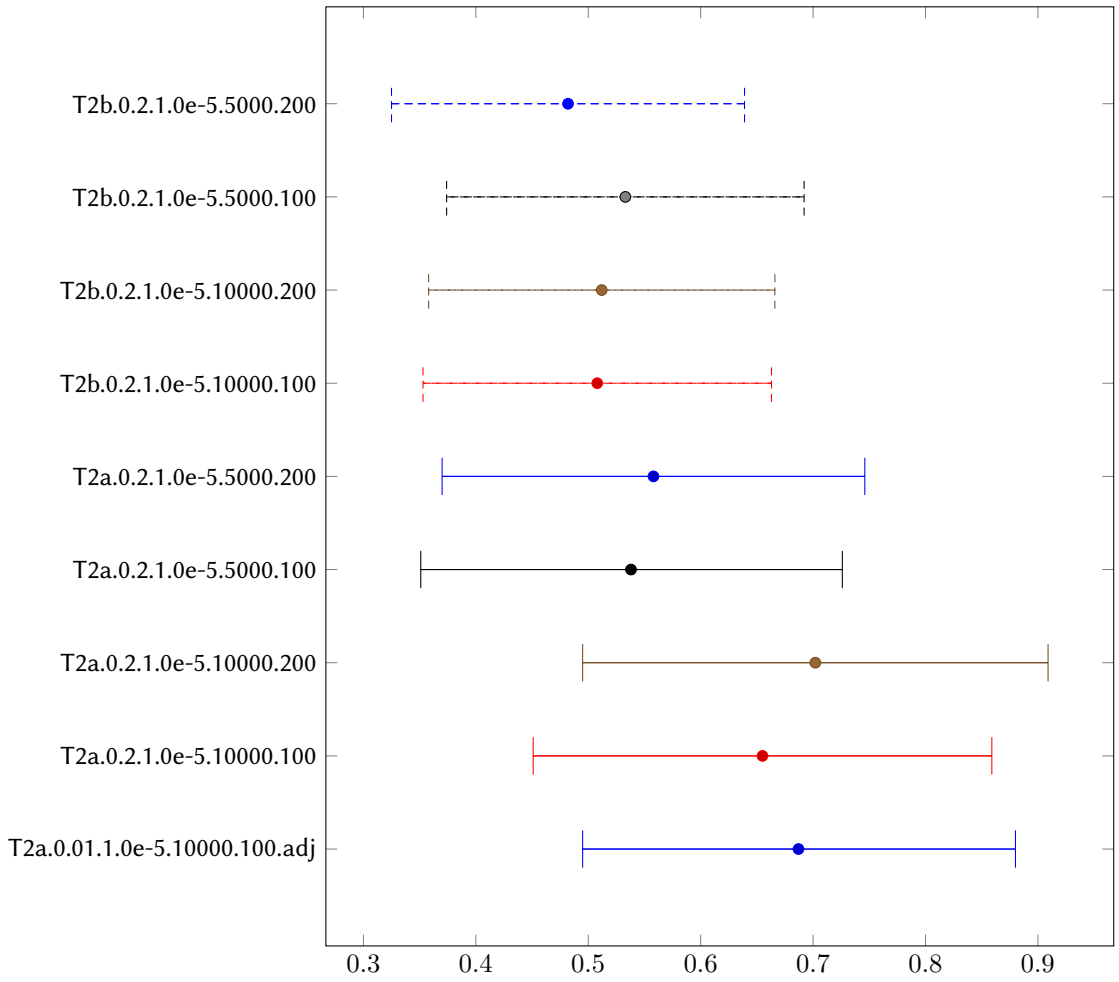


Figure 4: Task 2 - AUROC computed for all submitted runs with a 6-year time horizon. The bars in the plot show the 95% confidence interval.

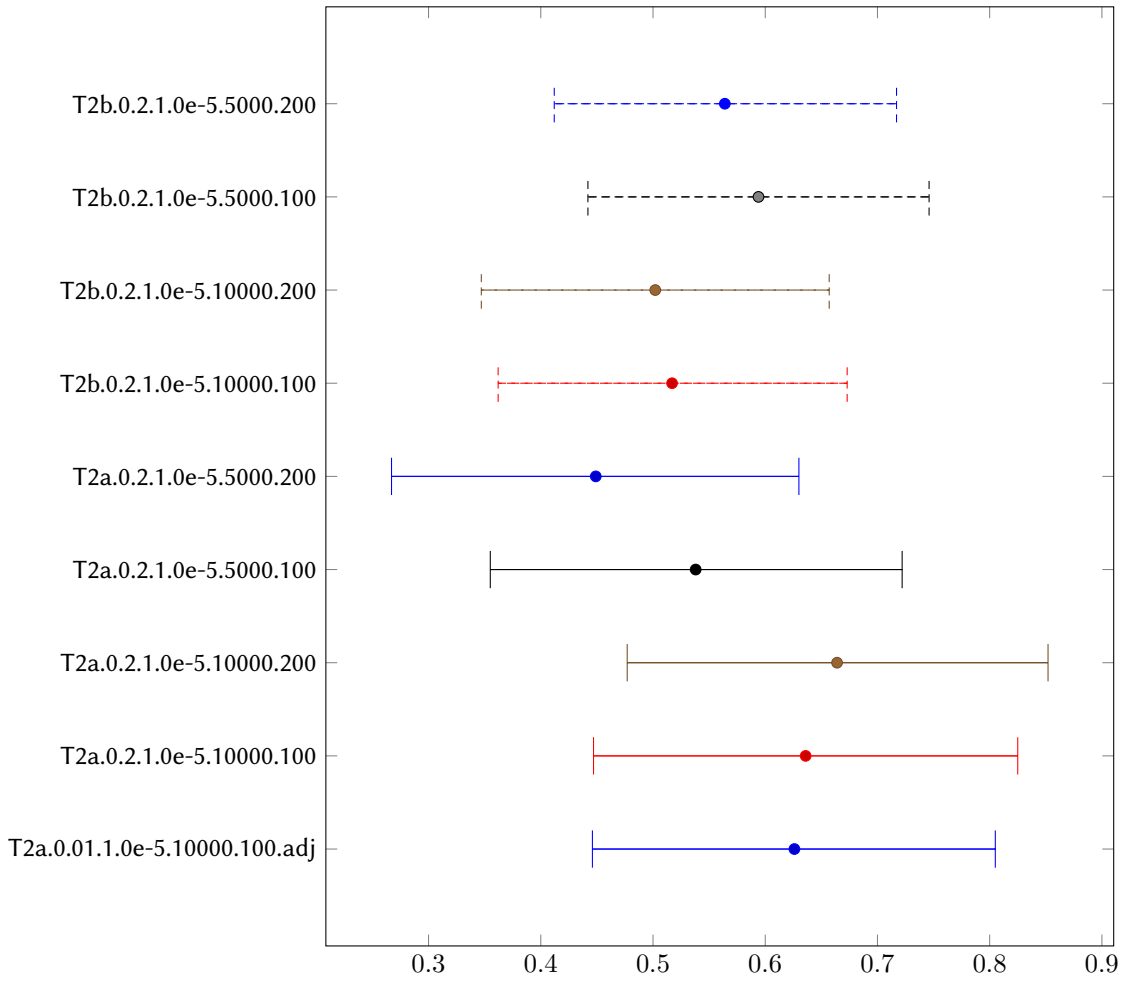


Figure 5: Task 2 - AUROC computed for all submitted runs with a 8-year time horizon. The bars in the plot show the 95% confidence interval.

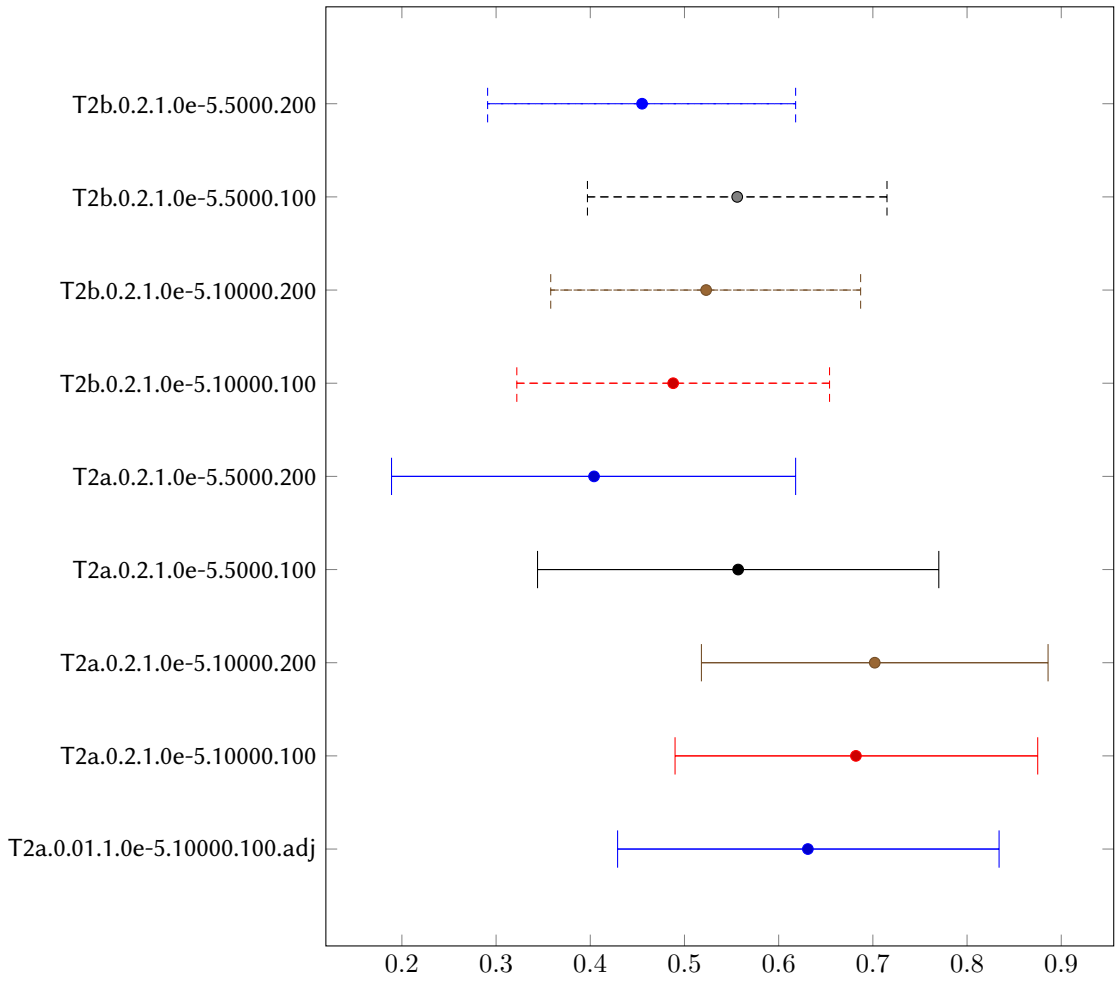


Figure 6: Task 2 - AUROC computed for all submitted runs with a 10-year time horizon. The bars in the plot show the 95% confidence interval.

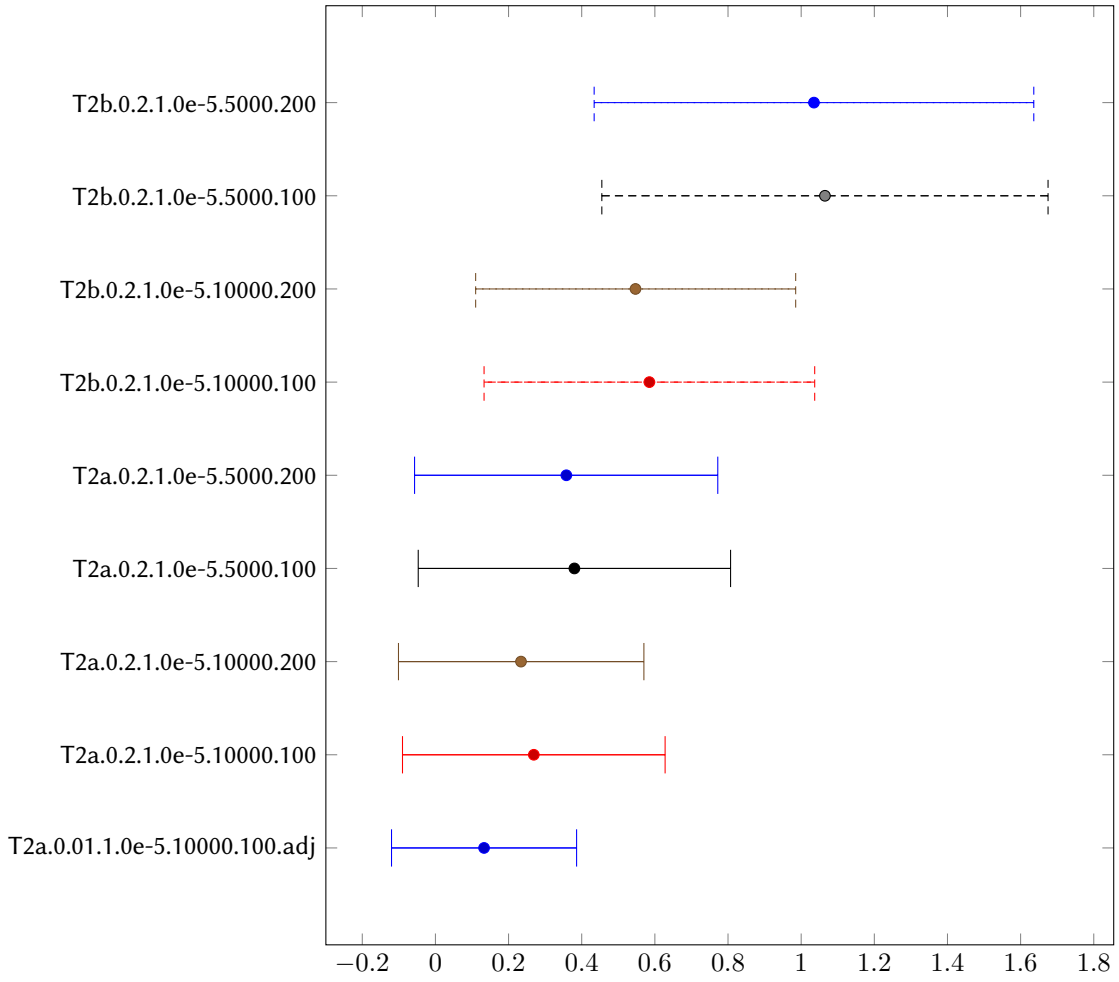


Figure 7: Task 2 - the ratio of observed to expected events computed for all submitted runs with a 2-year time horizon. The bars in the plot show the 95% confidence interval.

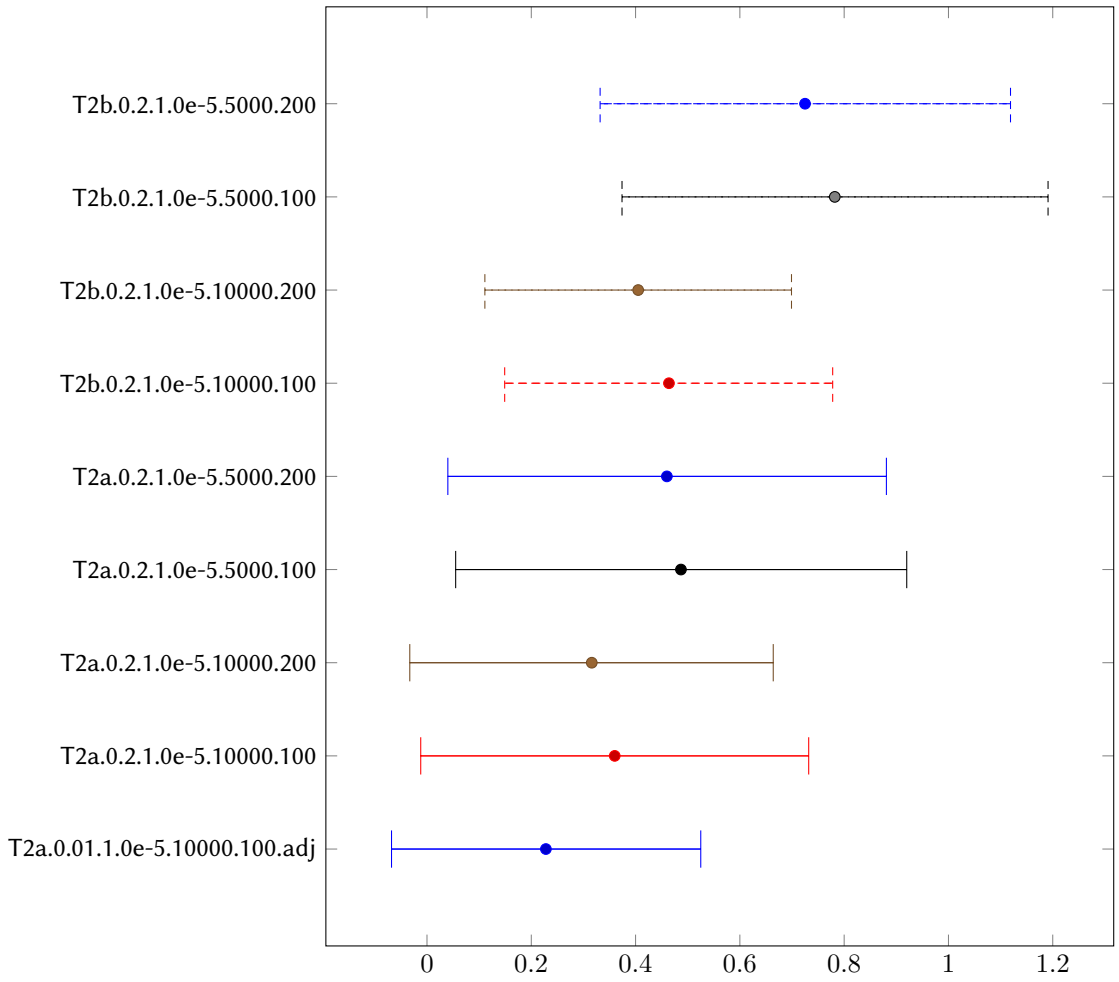


Figure 8: Task 2 - the ratio of observed to expected events computed for all submitted runs with a 4-year time horizon. The bars in the plot show the 95% confidence interval.

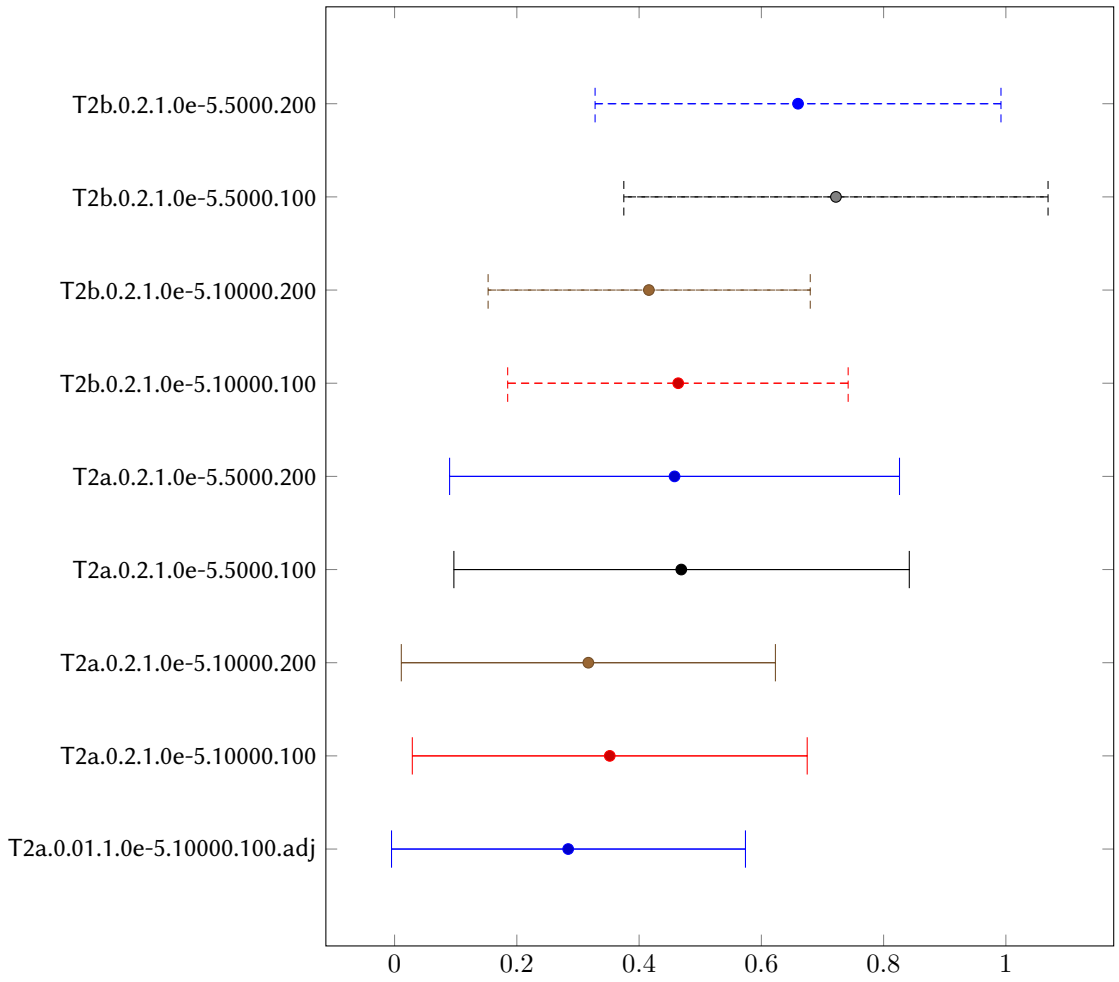


Figure 9: Task 2 - the ratio of observed to expected events computed for all submitted runs with a 6-year time horizon. The bars in the plot show the 95% confidence interval.

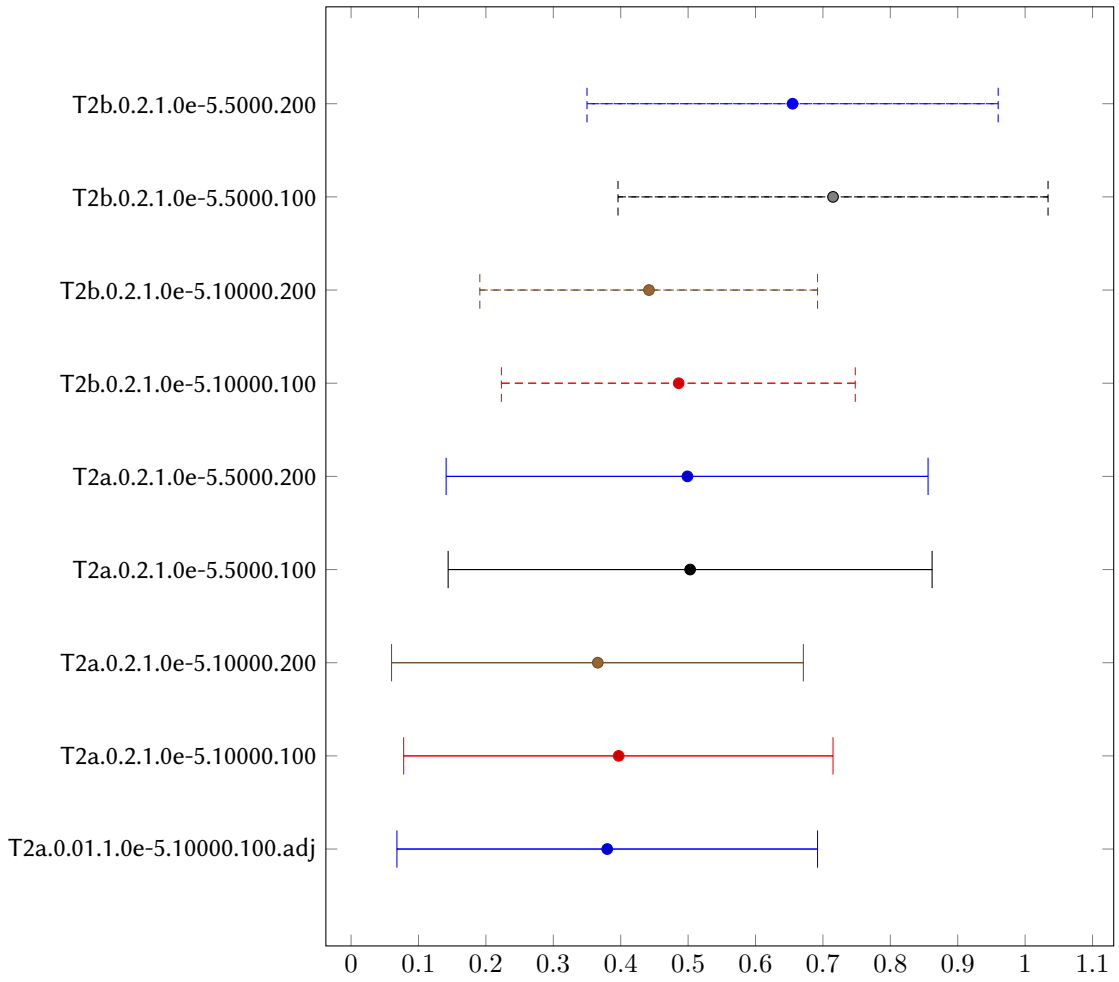


Figure 10: Task 2 - the ratio of observed to expected events computed for all submitted runs with a 8-year time horizon. The bars in the plot show the 95% confidence interval.

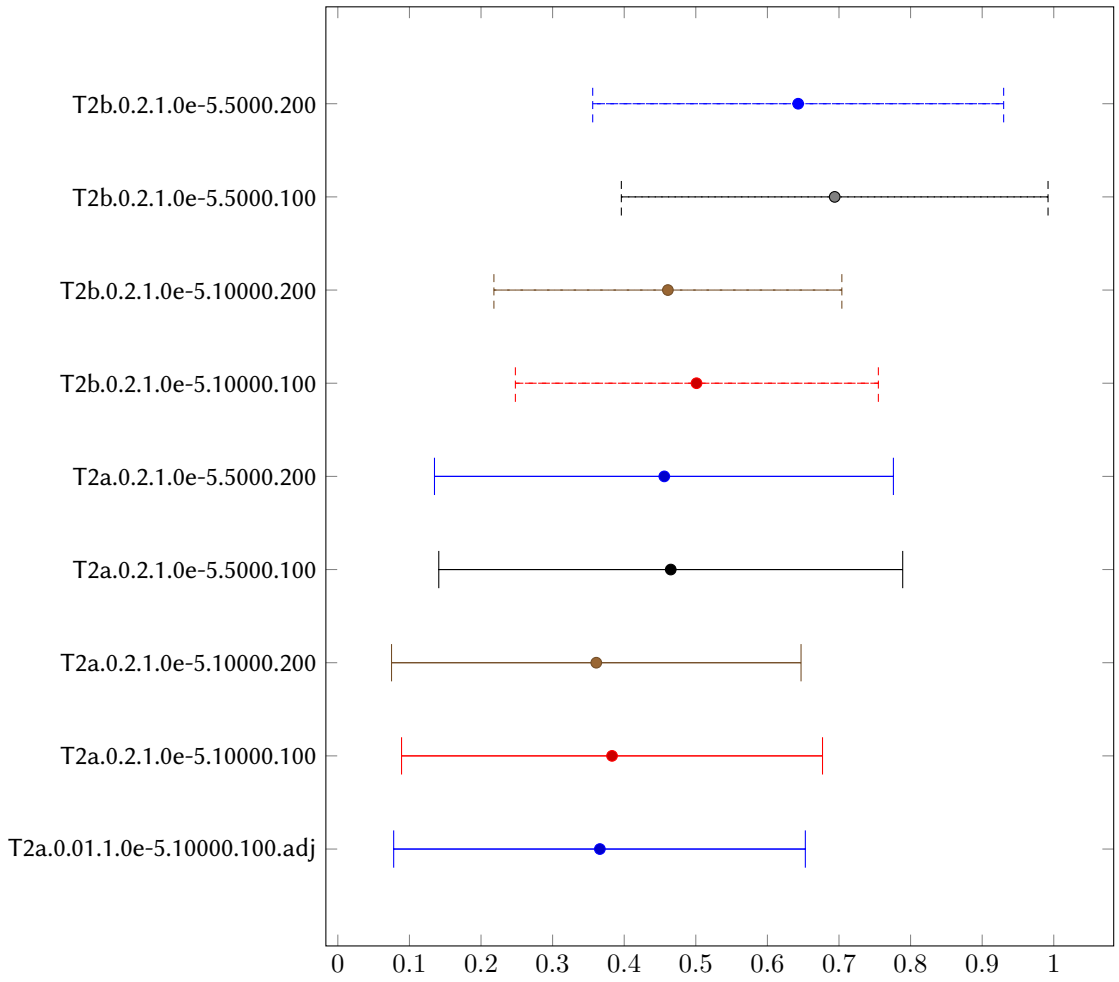


Figure 11: Task 2 - the ratio of observed to expected events computed for all submitted runs with a 10-year time horizon. The bars in the plot show the 95% confidence interval.