Darmstadt University of Applied Sciences

– Faculties of Mathematics and Natural Sciences & Computer Science –

Prediction Uncertainty in Weighted Machine Learning for Survival Data:

Concepts and Application to Organ Transplantation Registry Data

Submitted in partial fulfilment of the requirements for the degree of Master of Science (M.Sc.)

by

Rehan Butt

Matriculation number: 1114184

First Examiner	:	Prof. Dr. Antje Jahn
Second Examiner	:	Prof. Dr. Gunter Grieser
Issue date	:	29.04.2024
Submission date	:	28.10.2024

Declaration

I hereby declare that I have written the present thesis independently and that no other sources than those indicated in the bibliography have been used.

All passages that are literally or analogously taken from published or unpublished sources are identified as such.

The drawings or illustrations in this thesis have been created by myself or are provided with a corresponding source reference.

This thesis has not been submitted in the same or a similar form to any other examination authority.

Darmstadt, November 1, 2024

Rehan Butt

Abstract

Censored data frequently occurs in fields such as medical research and survival analysis, posing unique challenges for reliable variance estimation in predictive models. This study addresses these challenges by developing and evaluating a novel variance estimation method tailored to predictions from IPC-weighted classification models.

In this thesis, we develop the Infinitesimal-Jackknife-after-weighted-Bootstrap-unbiased (IJK-AWB-U) estimator. Building upon Wager's Infinitesimal Jackknife approach for unweighted bagged learners, the IJK-AWB-U estimator extends this methodology to IPC-weighted resampling and incorporates an effective bias correction to adjust for finite bootstrap samples. This novel estimator provides unbiased variance estimates for bagged learners, particularly those based on decision trees, when dealing with censored data.

An extensive simulation study was conducted following the ADEMP framework to compare the performance of the IJK-AWB-U estimator with traditional methods, including the nonparametric Bootstrap and the Jackknife-after-Bootstrap. The results demonstrated that the IJK-AWB-U estimator offers reliable and accurate variance estimates, especially under low to moderate censoring proportions and with larger training sample sizes. It effectively corrects the bias present in the original IJK-AWB estimator and achieves a favorable balance between accuracy and computational efficiency. Compared to the computationally intensive Bootstrap estimator, the IJK-AWB-U estimator provides similar accuracy with significantly reduced computational time. In contrast, while the Jackknife-after-Bootstrap estimator is unbiased under unweighted resampling, it consistently overestimated variance in IPC-weighted contexts, making it less reliable in such settings. Given these findings, the IJK-AWB-U estimator emerges as the preferred method for variance estimation in our context

The practical application of the IJK-AWB-U estimator to the TxReg dataset further validated its reliability and utility. The estimator produced confidence intervals closely aligned with those generated by the Bootstrap estimator, demonstrating its effectiveness in real-world scenarios where computational efficiency is crucial.

Future research directions include extending the IJK-AWB-U estimator to other machine learning architectures such as gradient-boosting models and neural networks, and applying it to diverse fields beyond medical research, including finance, engineering, and epidemiology. These extensions would further validate the estimator's versatility and adaptability.

Keywords: IPC-weighted Classification Models, Censored Data, Infinitesimal Jackknife, Jackknife, Jackknife-after-Bootstrap, Nonparametric Bootstrap, Survival Analysis, Bagged Learner, Decision Trees, Uncertainty Estimation

Zusammenfassung

Zensierte Daten treten häufig in Bereichen wie der medizinischen Forschung und der Überlebensanalyse auf und stellen einzigartige Herausforderungen für eine zuverlässige Varianzschätzung der Vorhersagen in prädiktiven Modellen dar. Diese Studie adressiert diese Herausforderungen durch die Entwicklung und Bewertung einer neuartigen Varianzschätzungsmethode, die auf IPC-gewichtete Klassifikationsmodelle zugeschnitten ist.

In dieser Arbeit entwickeln wir den *Infinitesimal-Jackknife-after-weighted-Bootstrap-unbiased* (IJK-AWB-U) Schätzer. Aufbauend auf Wagers Infinitesimal-Jackknife-Ansatz für ungewichtete Bagged Learner erweitert der IJK-AWB-U Schätzer diese Methodik auf IPC-gewichtetes Resampling und beinhaltet eine effektive Bias-Korrektur zur Anpassung für endliche Bootstrap-Stichproben. Dieser neuartige Schätzer liefert unverzerrte Varianzschätzungen für Bagged Learner, insbesondere solche, die auf Entscheidungsbäumen basieren, wenn mit zensierten Daten gearbeitet wird.

Eine umfangreiche Simulationsstudie wurde nach dem ADEMP-Rahmenwerk durchgeführt, um die Leistung des IJK-AWB-U Schätzers mit traditionellen Methoden, einschließlich des nichtparametrischen Bootstraps und des Jackknife-after-Bootstrap, zu vergleichen. Die Ergebnisse zeigten, dass der IJK-AWB-U Schätzer zuverlässige und genaue Varianzschätzungen bietet, insbesondere bei niedrigen bis moderaten Zensierungsanteilen und größeren Trainingsstichprobengrößen. Er korrigiert effektiv den Bias des ursprünglichen IJK-AWB Schätzers und erreicht ein günstiges Gleichgewicht zwischen Genauigkeit und rechnerischer Effizienz. Im Vergleich zum rechnerisch intensiven Bootstrap-Schätzer bietet der IJK-AWB-U Schätzer ähnliche Genauigkeit bei deutlich reduzierter Rechenzeit. Im Gegensatz dazu überschätzte der Jackknife-after-Bootstrap Schätzer, obwohl er unter ungewichtetem Resampling unverzerrt ist, die Varianz in IPC-gewichteten Kontexten konsequent, was ihn in solchen Fällen weniger zuverlässig macht. Angesichts dieser Ergebnisse erweist sich der IJK-AWB-U Schätzer als bevorzugte Methode für die Varianzschätzung in unserem Kontext.

Die praktische Anwendung des IJK-AWB-U Schätzers auf den TxReg-Datensatz bestätigte weiter seine Zuverlässigkeit und Nützlichkeit. Der Schätzer erzeugte Konfidenzintervalle, die eng mit denen übereinstimmen, die durch den Bootstrap-Schätzer generiert wurden, was seine Effektivität in realen Szenarien demonstriert, in denen rechnerische Effizienz entscheidend ist.

Zukünftige Forschungsrichtungen umfassen die Erweiterung des IJK-AWB-U Schätzers auf andere maschinelle Lernarchitekturen wie Gradient-Boosting-Modelle und neuronale Netze sowie die Anwendung in verschiedenen Bereichen jenseits der medizinischen Forschung, einschließlich Finanzen, Ingenieurwesen und Epidemiologie. Diese Erweiterungen würden die Vielseitigkeit und Anpassungsfähigkeit des Schätzers weiter validieren.

Contents

1 Introduction

The analysis of event times is a cornerstone of statistical research, especially in fields such as medical research and survival analysis. However, traditional methods that rely on event time models often face significant challenges when dealing with censored data, a common feature in real-world datasets. Censoring occurs when the objective is to model the time until a specific event occurs, but the event does not happen for all observations within the study period, or the exact event time cannot be precisely determined.

To address these challenges, reducing the problem to a classification framework, combined with inverse probability of censoring weights (IPCW), has emerged as a promising approach. This methodology allows for unbiased estimates from censored data, overcoming some of the limitations inherent in traditional event time models.[1]

The primary objective of this thesis is to evaluate the statistical properties of IPC-weighted classification methods in comparison to traditional event time models, with a particular focus on the uncertainty of predictions. The research specifically examines how the reduction to classification methods affects the uncertainty of predictions when applied to censored data. To follow this objective, a nonparametric method for variance estimation first has to be developed. The *Infinitesimal Jackknife* method for uncertainty estimation has been modified to better align with the unique requirements of IPC-weighted resampling, leading to the development of our *Infinitesimal-Jackknife-after-weighted-Bootstrap*.

A structured simulation study, conducted according to the ADEMP (Aim, Data-generating mechanism, Estimand, Methods, Performance measures) principle, serves as the basis for comparing these methods. The simulation study compares the performance of our Infinitesimal-Jackknife-after-weighted-Bootstrap, the Jackknife-after-Bootstrap, and the nonparametric Bootstrap in estimating the uncertainty of predictions from IPC-weighted classification methods. The results aim to provide insights into whether, and to what extent, the reduction to classification problems increases the uncertainty of predictions. These findings are further validated through their application to the TxReg dataset, serving as a practical example demonstrating the effectiveness of the newly developed method.

Overview of the Thesis:

Table 1.1 shows an overview of methods for estimating the variance of a prediction, generated with different type of learners. The methods marked in green in the table already exist in the literature and are discussed in our Methods chapter. In contrast, the methods marked in blue do not currently exist and are derived in our work. The Infinitesimal Jackknife method

[1] [Gon+21]

and it's bias corrected version for bagged learners with unweighted resampling was derived by Wager in Paper [WHE14] and is not part of this work.

Nonparametric Methods for estimating $\text{var}(\hat{p}(x))$	**Prediction** $\hat{p}(x)$ **derived from**		
	Unbagged Learner	**Bagged Learner with unweighted Resampling**	**Bagged Learner with IPC-weighted Resampling**
Nonparametric Bootstrap	Unbiased	Unbiased	Unbiased
Jackknife / Jackknife after Bootstrap	Unbiased for linear and smooth $\hat{p}(x)$	Unbiased for linear and smooth $\hat{p}(x)$	-
Infinitesimal Jackknife	Unbiased for smooth $\hat{p}(x)$	Unbiased for smooth $\hat{p}(x)$ and $B \to \infty$	Unbiased for smooth $\hat{p}(x)$ and $B \to \infty$
Bias corrected Infinitesimal Jackknife	Not applicable	Unbiased for smooth $\hat{p}(x)$ and $B < \infty$	Unbiased for smooth $\hat{p}(x)$ and $B < \infty$

Table 1.1: Methods for estimating the variance of a prediction, generated with different type of learners

- **Chapter 2.1** establishes the theoretical framework for IPC-weighted classification methods.
- **Chapter 2.2** discusses existing nonparametric variance estimation techniques for unbagged learners, such as the Jackknife, nonparametric Bootstrap and Infinitesimal Jackknife.
- **Chapter 2.3** introduces variance estimates for bagged learners under unweighted resampling, such as Jackknife and Jackknife-after-Bootstrap.
- **Chapter 2.4.1** introduces the *gold standard* of variance estimates, the nonparametric Bootstrap. Here it is adapted to fulfill the requirements for variance estimates of bagged learners under IPC-weighted resampling
- **Chapter 2.4.2 and 2.4.3** introduces the newly developed *Infinitesimal-Jackknife-after-weighted-Bootstrap* and it's bias corrected version, including its derivation. This method can be used for bagged learner with IPC-weighted resampling.
- **Chapter 3** presents the simulation study that compares the performance of traditional methods (Jackknife-after-Bootstrap and nonparametric Bootstrap) with the new method, following the ADEMP structure.
- **Chapter 4** applies the *Infinitesimal-Jackknife-after-weighted-Bootstrap*, the nonparametric Bootstrap and the Jackknife-after-Bootstrap methods to the TxReg dataset, highlighting the practical implications of the findings.
- **Chapter 5** concludes the thesis with a summary of the results and a discussion of potential directions for future research.

Through this structured approach, the thesis aims to provide a comprehensive understanding of uncertainty estimation in IPC-weighted classification methods and to demonstrate the advantages of the newly developed Infinitesimal-Jackknife-after-weighted-Bootstrap method in handling censored data within this framework.

2 Methods

2.1 Binary Classification with IPC-Weighted Resampling

In many real-world applications, especially in medical research and survival analysis, we encounter **right-censored data**, where the event of interest (e.g., disease occurrence, equipment failure) has not occurred for all subjects during the observation period. Traditional binary classification methods may not be suitable in this context, as they do not account for censoring and may lead to biased predictions. To address this challenge, we can employ *Inverse Probability of Censoring Weighting* (IPCW) in conjunction with *weighted resampling* to adjust for censoring and improve the predictive performance of classifiers. This will be the content of this chapter.

2.1.1 Inverse Probability of Censoring Weighting (IPCW)

IPCW is a technique used to handle right-censored data by weighting each observation inversely proportional to the probability of it being uncensored. This approach compensates for the loss of information due to censoring by giving more weight to observed data. To formally define the notations used for a right-censored dataset, we present the following definitions:

Definition 2.1.1: Right-Censored Dataset Notations
Let $\{(x_i, t_i, \delta_i)\}_{i=1}^n$ denote the dataset, where:

- $x_i \in \mathbb{R}^p$ is the feature vector for the i-th subject.
- t_i is the observed time, which is the minimum of the event time t_i^* and the censoring time c_i, i.e., $t_i = \min(t_i^*, c_i)$.
- $\delta_i = \mathbb{I}\{t_i^* \leq c_i\}$ is the event indicator, where $\delta_i = 1$ if the event is observed and $\delta_i = 0$ if the observation is censored.

To account for censoring, we define the IPC-weight for the i-th observation as follows:

Definition 2.1.2: IPC-Weights
Let τ be a specified time horizon for classification. The binary classification task aims to predict whether the event **does not** occur by time τ. Accordingly, we define the binary outcome y_i for each observation as:

$$y_i = \begin{cases} 1, & \text{if } t_i > \tau, \\ 0, & \text{otherwise.} \end{cases}$$

To account for censoring in this classification problem, the IPC-weights w_i are calculated based on the following three cases.[a]

1. **Case 1**: If $c_i < \tau$ and $c_i < t_i^*$ (i.e., the subject is censored before τ and before the event occurs),
$$w_i = 0.$$
2. **Case 2**: If $\tau < c_i$ and $\tau < t_i^*$ (i.e., the subject is censored after τ and the event has not occurred by τ),
$$w_i = \frac{1}{\hat{G}(\tau)}.$$
3. **Case 3**: If $t_i^* < c_i$ and $t_i^* < \tau$ (i.e., the event occurs before τ and before censoring),
$$w_i = \frac{1}{\hat{G}(t_i)}.$$

Here, $\hat{G}(t) = P(C > t)$ is the estimated survival function of the censoring distribution at time t. It can be estimated with the Kaplan-Meier Estimator (suitable when censoring is independent of event time).

[a] [Voc+16, p. 121]

By upweighting observations that provide more information about the **non-occurrence** of the event by τ, the method corrects for the bias introduced by censoring, leading to more reliable and unbiased predictions.[1]

2.1.2 IPC-Weighted Resampling for Binary Classification

To perform binary classification in the presence of right-censored data, it is essential to adjust the resampling process to account for censoring. The IPC-weighted resampling method integrates the IPC-weights into the bagging framework (the bagging framework, bagged learner, will be explained later in Section 2.3.1), ensuring that the resampling procedure appropriately reflects the censoring mechanism. This adjustment enables the creation of an ensemble of base classifiers that provide unbiased and accurate predictions.

[1] [Voc+16, p. 121f.]

Definition 2.1.3: IPC-Weighted Bagging Procedure
The IPC-weighted bagging procedure for binary classification with right-censored data is defined as follows:

1. **Compute IPC Weights**: For each observation i, calculate the IPC-weight w_i and binary outcome y_i using Definition 2.1.2.

2. **Weighted Resampling**: Generate B bootstrap samples by sampling observations with replacement from the original dataset. In each sampling step within a bootstrap sample, the probability of selecting observation i is proportional to its IPC-weight w_i. Specifically, the probability p_i of selecting observation i in each draw is:
$$p_i = \frac{w_i}{\sum_{j=1}^{n} w_j},$$
where p_i represents the selection probability for observation i in each individual resampling step.

3. **Model Training**: For each bootstrap sample $b = 1, \ldots, B$, train a base classifier $t^{(b)}$ on the resampled data $\{(x_i, y_i)\}_{i=1}^{n}$. This ensures that the base learner is trained on data adjusted for censoring.

4. **Bagged Learner's Prediction**: For a new observation x_{new}, aggregate the predictions from all base classifiers to form the final prediction of the Bagged Learner. This can be done by averaging the predicted probabilities:
$$\hat{y}(x_{\text{new}}) = \frac{1}{B} \sum_{b=1}^{B} t^{(b)}(x_{\text{new}}).$$

This procedure ensures that the resampling process accounts for the censoring mechanism, specifically tailored for the binary classification task of predicting events by time τ.

Choice of Base Learners

The performance of IPC-weighted bagging procedure can vary depending on the choice of base classifiers. Common options include:

- Decision Trees: Simple and interpretable but may have high variance.
- *Random Forests*: Ensemble of trees that can capture complex interactions.
- *Gradient Boosting Machines*: Powerful but may require careful tuning.

In this work we will work with **Decision Trees**.

Performance Evaluation with IPCW-MSE

Evaluating the performance of classifiers in the presence of censoring requires metrics that account for incomplete observations. The *IPCW Mean Squared Error* (IPCW-MSE) is one such metric. It measures the average squared difference between the predicted probabilities and the true outcomes, adjusted for censoring by weighting with IPC weights. It is computed as:

Definition 2.1.4: IPCW Mean Squared Error (IPCW-MSE)
The IPCW-MSE is defined as:

$$\text{IPCW-MSE} = \frac{1}{n} \sum_{i=1}^{n} p_i \left(y_i - \hat{y}(x_i)\right)^2,$$

where:

- y_i is the true binary outcome for observation i (i.e., $y_i = 1$ if the event is seen by τ, and $y_i = 0$ otherwise).
- $\hat{y}(x_i)$ is the predicted probability of the event occurring by τ for observation i.
- p_i are the normalized IPC weights as defined in Definition 2.1.3.

This metric accounts for censoring by upweighting the uncensored observations, providing an unbiased estimate of the mean squared error. Lower values of IPCW-MSE indicate better predictive performance. The IPCW-MSE is the same as the expected Brier Score.[2]

In this chapter, we introduced the **Inverse Probability of Censoring Weighting (IPCW)** method and its integration with **weighted resampling** to address right-censored data in binary classification tasks. By defining the necessary notations and outlining the **IPC-Weighted Bagging Procedure**, we demonstrated how weighted resampling and ensemble learning can mitigate the bias introduced by censoring. Additionally, we introduced the **IPCW Mean Squared Error (IPCW-MSE)** as an unbiased metric for evaluating classifier performance. The next chapter will explore **Nonparametric Variance Estimates**, providing a foundation to accurately estimate the variance of predictions generated by the IPC-weighted bagging procedure.

2.2 Nonparametric Variance Estimates

Variance estimation is a fundamental component in statistical analysis, providing insights into the variability and reliability of predictive models. Accurate variance estimates are crucial for assessing the uncertainty associated with model predictions, enabling informed decision-making and enhancing the credibility of analytical results. Unlike parametric methods, nonparametric variance estimation techniques make minimal assumptions about the underlying data distribution, offering greater flexibility and robustness in diverse applications.

This chapter explores a range of nonparametric methods for estimating variance, including the **Jackknife Estimate**, **Nonparametric Bootstrap**, and their **Geometric Interpretations**. We also introduce the **Infinitesimal Jackknife** as an advanced technique for variance approximation. Through theoretical discussions and simulation studies, we demonstrate the effectiveness and applicability of these methods in various scenarios. By establishing a comprehensive understanding of nonparametric variance estimation, this chapter lays the groundwork for subsequent discussions on variance estimation for bagged learners in Chapter 2.3 and 2.4.

[2] [Gra+99, p. 2538]

2.2.1 Jackknife

The jackknife method is a classic statistical technique used to estimate both the bias and variance of an estimator. This method, which predates the more widely known bootstrap technique, shares several conceptual similarities with it but is often simpler to implement. In this section, we will explore the fundamentals of the jackknife method and examine its limitations.

Definition of the Jackknife

The jackknife method is typically employed in situations involving one-sample problems, where the dataset, denoted by

$$\mathbf{X} = \{X_1, X_2, \ldots, X_n\}, \tag{2.2.1}$$

is assumed to consist of independent and identically distributed (iid) observations from an unknown probability distribution F. A real-valued statistic,

$$\hat{\theta} = s(\mathbf{X}), \tag{2.2.2}$$

is computed using a function $s : \mathbb{R}^n \to \mathbb{R}$, which is applied to the entire sample $\mathbf{X}$. The function s is permutation invariant, meaning that the order of the inputs X_i does not affect the result. The primary objective of the jackknife method is to estimate the variance of $\hat{\theta}$, which reflects the variability of $\hat{\theta}$ under the sampling model.

The jackknife procedure begins by systematically leaving out each observation X_i from the sample to form a reduced dataset,

$$\mathbf{X}^{(i)} = \{X_1, X_2, \ldots, X_{i-1}, X_{i+1}, \ldots, X_n\}. \tag{2.2.3}$$

Next, the statistic of interest, $\hat{\theta}^{(i)}$, is recalculated using this reduced sample:

$$\hat{\theta}^{(i)} = s(\mathbf{X}^{(i)}). \tag{2.2.4}$$

Definition 2.2.1: Jackknife estimate of Variance
The jackknife estimate of the variance of $\hat{\theta}$ is defined as[a]:

$$\hat{\mathrm{var}}_{\mathrm{JK}}(\hat{\theta}) = \frac{n-1}{n} \sum_{i=1}^{n} \left(\hat{\theta}^{(i)} - \hat{\theta}^{(\cdot)}\right)^2,$$

where

$$\hat{\theta}^{(\cdot)} = \frac{1}{n} \sum_{i=1}^{n} \hat{\theta}^{(i)}.$$

[a] [ET93, p. 141]

In addition to variance estimation, the jackknife method can also be used to detect outliers by evaluating the influence of each observation X_i on the overall variance estimate. The contribution of each observation X_i to the jackknife variance estimate can be assessed by calculating:

$$\frac{(\hat{\theta}^{(i)} - \hat{\theta}^{(\cdot)})^2}{\sum_{j=1}^{n}(\hat{\theta}^{(j)} - \hat{\theta}^{(\cdot)})^2}. \tag{2.2.5}$$

One of the key advantages of the jackknife method is its flexibility. Unlike parametric methods, the jackknife does not require any specific assumptions about the underlying distribution F. This nonparametric nature makes the jackknife a broadly applicable tool. Moreover, the jackknife method is automated and straightforward: a single algorithm can take the dataset $\mathbf{X}$ and function s as inputs and output the jackknife variance estimate. The method operates under the assumption that the statistic $\hat{\theta}$ behaves smoothly (i.e., small changes in the data lead to small changes in the statistic's value). As defined in Definition 2.2.15, the term *smooth* will be discussed in more detail later.

While the jackknife method offers simplicity and robustness, it also has limitations, particularly when dealing with highly skewed or heavy-tailed distributions, or in the presence of strong dependencies between observations. In such cases, more sophisticated methods like the bootstrap may provide better variance estimates.[3] [4]

Examples

The motivation behind the jackknife variance estimation formula (Definition 2.2.1) becomes clearer when considering a simple scenario where $\hat{\theta}$ represents the sample mean of the dataset $\mathbf{X}$.

Example 1
Jackknife Variance of the Sample Mean

For the sample mean, $\hat{\theta} = \bar{X}$, where each observation X_i belongs to $\mathbb{R}^1$, the jackknife estimate of variance can be derived as follows:

$$\hat{\theta}^{(i)} = \frac{1}{n-1}\left[\sum_{j=1}^{n}(X_j) - X_i\right] = \frac{n\bar{X} - X_i}{n-1},$$

where $\hat{\theta}^{(i)}$ is the estimate obtained by leaving out the i-th observation. The mean of these leave-one-out estimates is:

$$\hat{\theta}^{(\cdot)} = \frac{1}{n}\sum_{i=1}^{n}\hat{\theta}^{(i)} = \frac{1}{n}\sum_{i=1}^{n}\frac{n\bar{X} - X_i}{n-1} = \bar{X}.$$

The difference between each leave-one-out estimate and the overall mean is:

$$\hat{\theta}^{(i)} - \hat{\theta}^{(\cdot)} = \frac{n\bar{X} - X_i}{n-1} - \bar{X} = \frac{\bar{X} - X_i}{n-1}.$$

Substituting this into the jackknife variance formula gives:

$$\hat{\text{var}}_{\text{JK}}(\hat{\theta}) = \frac{n-1}{n}\sum_{i=1}^{n}\left(\hat{\theta}^{(i)} - \hat{\theta}^{(\cdot)}\right)^2 = \frac{\sum_{i=1}^{n}(\bar{X} - X_i)^2}{n(n-1)} = \frac{\hat{\sigma}^2_{\mathbf{X}}}{n}.$$

[3] [ET93, pp. 188-189]
[4] [ET93, pp. 309-311]

This example illustrates that the jackknife estimate of variance for the sample mean is essentially the empirical variance of the sample $\mathbf{X}$ divided by the sample size n, which aligns with the classical formula for variance estimation for the sample mean. This shows the consistency of the jackknife method for the sample mean. More generally, for a linear statistic

Definition 2.2.2: Linear Statistic
A statistic is considered *linear* if it can be expressed as follows:[a]

$$\hat{\theta} = s(\mathbf{X}) = c + \frac{1}{n}\sum_{i=1}^{n}\alpha(X_i),$$

where $\alpha(\cdot)$ is a function, and c is a constant. A simple example of a linear statistic is the sample mean, where $\alpha(X_i) = X_i$ and $c = 0$.

[a][ES81, p. 590f.]

, the jackknife method also performs well, as we will see later in the subsection *Bias of Jackknife* (2.2.1).

Example 2
Jackknife Variance of a Function of the Sample Mean

Now, let us consider a statistic that is a function of the sample mean, such as $\hat{\theta} = g(\bar{X})$. In such cases, the variance of this statistic is typically calculated using the delta method[5]:

$$\hat{\text{var}}_{\text{delta}}(\hat{\theta}) = g'(\mu)^2 \cdot \text{var}(\bar{X}),$$

where $\mu = E(X)$. Here, the mean of the X_i values is typically used for μ, and an estimate for the true variance $\text{var}(\bar{X})$ is substituted. Therefore, the delta method's estimate of variance for the statistic $g(\bar{X})$ is:

$$\hat{\text{var}}_{\text{delta}}(\hat{\theta}) = g'(\bar{X})^2 \cdot \frac{\hat{\sigma}^2_{\mathbf{X}}}{n}.$$

For the calculation of the jackknife estimate of variance for the statistic $\hat{\theta} = g(\bar{X})$, we need g to be a smooth function (cf. Definition 2.2.15). For simplicity we calculate it for the case, where each observation $X_i \in \mathbb{R}^1$:

$$\hat{\theta}^{(i)} = g(\bar{X}^{(i)}) = g\left(\frac{n\bar{X} - X_i}{n-1}\right).$$

Using a linear approximation (first-order Taylor expansion around $\bar{X}$):

$$\hat{\theta}^{(i)} = g(\bar{X}) + g'(\bar{X}) \cdot \left(\bar{X}^{(i)} - \bar{X}\right) = g(\bar{X}) + g'(\bar{X}) \cdot \frac{\bar{X} - X_i}{n-1}.$$

The mean of these leave-one-out estimates is:

$$\hat{\theta}^{(\cdot)} = \frac{1}{n}\sum_{i=1}^{n}\left(g(\bar{X}) + g'(\bar{X}) \cdot \frac{\bar{X} - X_i}{n-1}\right) = g(\bar{X}).$$

[5][EH16, p. 20]

Therefore, the Jackknife variance estimate is:

$$\hat{\text{var}}_{\text{JK}}(\hat{\theta}) = \frac{n-1}{n}\sum_{i=1}^{n}\left(g'(\bar{X})\cdot\frac{\bar{X}-X_i}{n-1}\right)^2 = g'(\bar{X})^2\cdot\frac{\hat{\sigma}^2_{\mathbf{X}}}{n}.$$

This example demonstrates that the delta method provides the same variance estimate as the jackknife when a linear approximation is used for $\hat{\theta}^{(i)}$. However, if the statistic $\hat{\theta}$ is nonlinear, the jackknife's reliance on a first-order Taylor expansion can lead to a loss of information, resulting in a biased variance estimate $\hat{\text{var}}_{\text{JK}}(\hat{\theta})$. **Is the square of the mean a linear statistic?** According to Definition 2.2.2, a statistic is linear if it can be expressed as:

$$\hat{\theta} = s(\mathbf{X}) = c + \frac{1}{n}\sum_{i=1}^{n}\alpha(X_i),$$

where $\alpha(\cdot)$ is a function and c is a constant. The square of the mean, $\hat{\theta} = \bar{X}^2$, does not fit this form because it involves a quadratic transformation of $\bar{X}$. Specifically:

$$\hat{\theta} = \bar{X}^2 = \left(\frac{1}{n}\sum_{i=1}^{n}X_i\right)^2 = \frac{1}{n^2}\sum_{i=1}^{n}\sum_{j=1}^{n}X_iX_j,$$

which includes product terms X_iX_j that are not covered by the linear definition. Therefore, since $\hat{\theta} = \bar{X}^2$ is nonlinear, using a linear approximation in the jackknife method can result in a biased variance estimate due to information loss.

Example 3
Jackknife Variance of the Median

Beyond the linearity of the statistic, another important assumption of the jackknife method is that the statistic is *smooth* (smoothness will be discussed in more detail in Definition 2.2.15 later). Some statistics, like the median, do not satisfy this assumption, leading to inconsistencies in the jackknife variance estimate, as illustrated in the following example.

Consider the following ordered values from a data sample:

$$\mathbf{X} = [10,\ 27,\ 31,\ 40,\ 46,\ 50,\ 52,\ 104,\ 146].$$

The median of this sample is 46. If we adjust one of the observations, say $X_4 = 40$, by increasing its value, the median remains unchanged until X_4 exceeds 46. Once X_4 surpasses 46, the median changes abruptly. This behavior illustrates that the median is not a smooth function of the data. As a result, the jackknife variance estimate $\hat{\text{var}}_{\text{JK}}(\hat{\theta})$ becomes inconsistent when applied to the median. Using the data sample above, the jackknife estimates $\hat{\theta}^{(i)}$ for the median yield only three distinct values: 43, 45, and 48. The resulting jackknife variance estimate is:

$$\hat{\text{var}}_{\text{JK}}(\hat{\theta}) = 44.64.$$

In contrast, the bootstrap variance estimate, based on $B = 100$ bootstrap samples, is 71.23, which is significantly larger. As $n \to \infty$, the jackknife variance estimate $\hat{\text{var}}_{\text{JK}}(\hat{\theta})$ fails to converge to the true variance, demonstrating its inconsistency for non-smooth statistics like the median. On the other hand, the bootstrap method accounts for variability in the data and provides a consistent estimate even for non-smooth statistics.[6]

[6] [ET93, p. 148]

Bias of Jackknife

As demonstrated in the previous examples, the jackknife method effectively estimates the variance of mean statistics and functions of mean statistics, when the statistic is both linear and smooth. However, for nonlinear and non-smooth statistics, such as the median, the jackknife method may introduce bias into the variance estimate. This section explores the sources of bias in the jackknife variance estimate.

The jackknife variance estimate, $\hat{\text{var}}_{\text{JK}}(\hat{\theta})$ (as defined in Definition 2.2.1), relies on leave-one-out samples $\mathbf{X}^{(i)}$ of size $n-1$. In contrast, the original statistic $\hat{\theta}$, whose variance we aim to estimate, is calculated from the full sample $\mathbf{X}$ of size n. This discrepancy in sample sizes can introduce bias into the variance estimate. The jackknife method estimates the true variance $\text{var}(\hat{\theta})$ through two key steps:[7]

1. **Estimate the Variance Based on $n-1$ Observations:**
   We estimate the variance of the statistic based on $n-1$ observations using:

$$\hat{\text{var}}_{n-1} = \sum_{i=1}^{n} \left(\hat{\theta}^{(i)} - \hat{\theta}^{(\cdot)}\right)^2, \tag{2.2.6}$$

   where $\hat{\theta}^{(i)} = s(\mathbf{X}^{(i)})$ is the statistic computed from the sample with the i-th observation omitted, and $\hat{\theta}^{(\cdot)} = \frac{1}{n}\sum_{i=1}^{n} \hat{\theta}^{(i)}$ is the average of the leave-one-out estimates.

2. **Adjust to Estimate the Variance Based on n Observations:**
   We adjust $\hat{\text{var}}_{n-1}$ to estimate the variance of the statistic based on n observations:

$$\hat{\text{var}}_{\text{JK}}(\hat{\theta}) = \frac{n-1}{n}\hat{\text{var}}_{n-1}. \tag{2.2.7}$$

   This adjustment accounts for the difference in sample sizes between the original statistic and the leave-one-out statistics.

Efron and Stein showed in [ES81] that:

Theorem 2.2.3
For linear statistics (cf. Definition 2.2.2) the jackknife variance estimate (cf. Definition 2.2.1) is unbiased.[a]

$$E\left(\hat{\text{var}}_{\text{JK}}(\hat{\theta})\right) = \text{var}(\hat{\theta})$$

[a] [ES81, pp. 590-591]

For nonlinear statistics, the jackknife variance estimate can be biased, and the direction and magnitude of the bias depend on the specific statistic and the underlying distribution. Despite the potential for bias, Efron and Stein identified that for specific classes of nonlinear statistics, the jackknife variance estimate remains asymptotically unbiased as the sample size n becomes large.

[7] [ES81, p. 586]

Theorem 2.2.4
For statistics that belong to the classes of U-statistics, von Mises functionals, or quadratic forms, the jackknife variance estimate (cf. Definition 2.2.1) is asymptotically consistent and conservative, meaning that as $n \to \infty$:[a]

$$E\left(\hat{\text{var}}_{\text{JK}}(\hat{\theta})\right) \geq \text{var}(\hat{\theta}) + o(1) \quad \text{and} \quad E\left(\hat{\text{var}}_{\text{JK}}(\hat{\theta})\right) \to \text{var}(\hat{\theta})$$

where $o(1)$ denotes a positive term that goes to zero as $n \to \infty$.

[a] [ES81, pp. 592-593]

These classes cover a significant portion of commonly used nonlinear statistics. Examples include:

- U-Statistics: Sample variance, Gini coefficient, Kendall's tau, Pearson correlation
- von Mises Functionals: Empirical cumulative distribution function (ECDF), sample moments
- Quadratic Functionals: Sample covariance, Hajek projection, eigenvalue estimates

For these statistics, the jackknife variance estimator becomes increasingly accurate as the sample size increases.

In conclusion, while the jackknife method demonstrates robustness in estimating the variance for linear statistics and certain classes of nonlinear statistics, it can introduce bias when applied to other nonlinear statistics, with the bias potentially being either positive or negative. Therefore, understanding the properties of the statistic in question is crucial when using the jackknife method for variance estimation. An alternative approach for variance estimation is the nonparametric bootstrap method, which we will explore in the next section.

2.2.2 Nonparametric Bootstrap

The nonparametric bootstrap is a versatile and robust resampling technique that extends the capabilities of traditional methods, such as the jackknife. Unlike the jackknife, which systematically excludes one observation at a time, the bootstrap resamples with replacement from the original dataset, allowing the same observation to appear multiple times in a bootstrap sample. This method is particularly effective for estimating the distribution of a statistic and its variance, especially when dealing with non smooth (cf. Definition 2.2.15) or non linear (cf. Definition 2.2.2) statistics.[8]

The nonparametric bootstrap process is illustrated in Figure 2.1.

[8] [DH97]

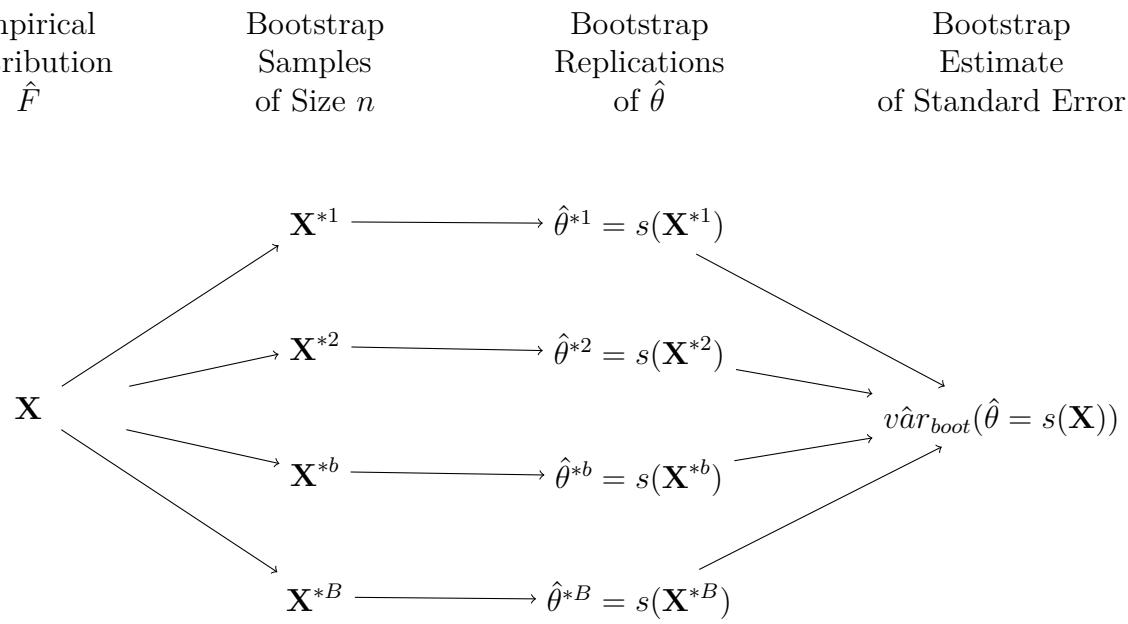


Figure 2.1: Nonparametric Bootstrap Process

The process involves the following steps:[9]

1. **Generate Bootstrap Samples:**
Create B bootstrap samples $\mathbf{X}^{*1}, \mathbf{X}^{*2}, \ldots, \mathbf{X}^{*B}$ by resampling n observations with replacement from the original dataset $\mathbf{X} = \{X_1, X_2, \ldots, X_n\}$.

2. **Compute Bootstrap Replicates:**
For each bootstrap sample, compute the statistic of interest: $\hat{\theta}^{*b} = s(\mathbf{X}^{*b})$.

3. **Estimate Variance:**
The empirical variance of the B bootstrap replicates $\hat{\theta}^{*b}$ provides an estimate of the variance of the statistic.

Definition 2.2.5: Bootstrap estimate of Variance
The bootstrap estimate of the variance of a statistic $\hat{\theta}$ is given by:

$$\hat{\text{var}}_{boot}(\hat{\theta}) = \frac{1}{B-1} \sum_{b=1}^{B} \left(\hat{\theta}^{*b} - \overline{\hat{\theta}^{*}} \right)^2,$$

where:

$$\overline{\hat{\theta}^{*}} = \frac{1}{B} \sum_{b=1}^{B} \hat{\theta}^{*b}.$$

Like the jackknife method, the bootstrap method does not require any specific assumptions about the form of the underlying distribution F. By relying on the empirical distribution of the data, the bootstrap method proves to be a robust tool for statistical inference. It

[9] [EH16, p. 159]

offers several advantages over traditional methods such as the jackknife. Notably, it is highly versatile and can be applied to various types of data and statistics, including those that are non-smooth or non-linear. In addition to variance estimation, the bootstrap method can be employed to construct confidence intervals and perform hypothesis tests. These applications further highlight the flexibility and power of the bootstrap method in statistical analysis, making it a valuable technique for a wide range of statistical problems.

In summary, while both jackknife and bootstrap offer robust solutions for variance estimation, their fundamental differences can be more clearly understood through a geometric perspective, which we explore in the next section.

2.2.3 Geometric Interpretation of Jackknife and Nonparametric Bootstrap

The preceding sections introduced the conceptual and mathematical foundations of the jackknife and bootstrap methods. While these approaches have been analyzed primarily through algebraic properties, understanding their geometric interpretation can provide a deeper and more intuitive grasp of how these resampling techniques operate. By visualizing jackknife and bootstrap within a geometric space, we can gain insight into how the mass distribution over the data points is modified and how these changes impact the resulting variance estimates.

Resampling Vectors and the Empirical Distribution

Consider a data sample $\mathbf{X} = \{X_1, X_2, \ldots, X_n\}$, where each observation X_i is independently and identically distributed (i.i.d.) from an unknown distribution F. The empirical distribution $\hat{F}$ assigns a uniform mass $\frac{1}{n}$ to each observation X_i, leading to the statistic

$$\hat{\theta} = s(\mathbf{X}) = h\left(\hat{F}\right). \tag{2.2.8}$$

Here, h is a *functional*, which maps a distribution function F to a real number. The statistic $\hat{\theta}$ is the quantity of interest, and in the following sections, we focus on estimating its variance.

Definition 2.2.6: Functional h
A *functional* h is a mapping from the space of distribution functions to the real numbers, defined by:

$$\theta = h(F) = \int g(x)\, dF(x),$$

where F is a distribution function and g is a measurable function.

For example, when we want to calculate the mean statistic, the function g would be the identity function. Therefore, we have:

$$\hat{\theta} = h(\hat{F}) = \int g(x)\, d\hat{F}(x) = \frac{1}{n}\sum_{i=1}^{n} g(X_i) = \frac{1}{n}\sum_{i=1}^{n} X_i. \tag{2.2.9}$$

This general formulation allows h to represent a broad class of statistics, depending on the choice of g.

The jackknife and bootstrap methods explore how this statistic changes when the mass distribution is altered. Geometrically, these methods can be understood by examining the resampling vectors.

Definition 2.2.7: Resampling Vector

$$M = (m_1, \ldots, m_n)^T, \quad \text{with } 0 \leq m_i \text{ and } \sum_{i=1}^{n} m_i = 1$$

A resampling vector M represents a distribution of mass among the observations X_i, where each m_i indicates the proportion of total mass assigned to X_i.

This collection of vectors forms an n-dimensional *simplex*.

Definition 2.2.8: n-Dimensional Simplex
An n-dimensional simplex is the set of all vectors $M = (m_1, m_2, \ldots, m_n)^T$ satisfying:

$$m_i \geq 0 \quad \text{for all } i, \quad \sum_{i=1}^{n} m_i = 1.$$

It represents all possible ways to distribute a unit mass among n non-negative components.

The concept of the simplex is fundamental as it allows us to map the relationship between the resampling method and the variability of the statistic in a concrete way. The empirical distribution under resampling is denoted by $\hat{F}^{(M)}$, which places mass m_i on each observation X_i. Consequently, the corresponding statistic becomes:

Definition 2.2.9: Statistic under Resampling with Weighted Empirical Distribution
The statistic under resampling with weighted empirical distribution is defined as:

$$\hat{\theta}^{(M)} = H(M) = h\left(\hat{F}^{(M)}\right) = \sum_{i=1}^{n} m_i g(X_i),$$

where:

- $M = (m_1, m_2, \ldots, m_n)^T$ is the resampling vector,
- g is the function associated with the functional h,
- $\hat{F}^{(M)}$ is the weighted empirical distribution assigning mass m_i to observation X_i.

If the mass is equally distributed, meaning each observation receives an equal share of the total mass, then the statistic under resampling with weighted empirical distribution simplifies to the statistic of interest whose variance we aim to estimate.

$$\hat{\theta} = H\left(M_0\right) = h\left(\hat{F}^{(M_0)}\right), \tag{2.2.10}$$

where

Definition 2.2.10: Uniform Resampling Vector
The *uniform resampling vector* M_0 is defined as:

$$M_0 = \left(\frac{1}{n}, \frac{1}{n}, \ldots, \frac{1}{n}\right)^\top,$$

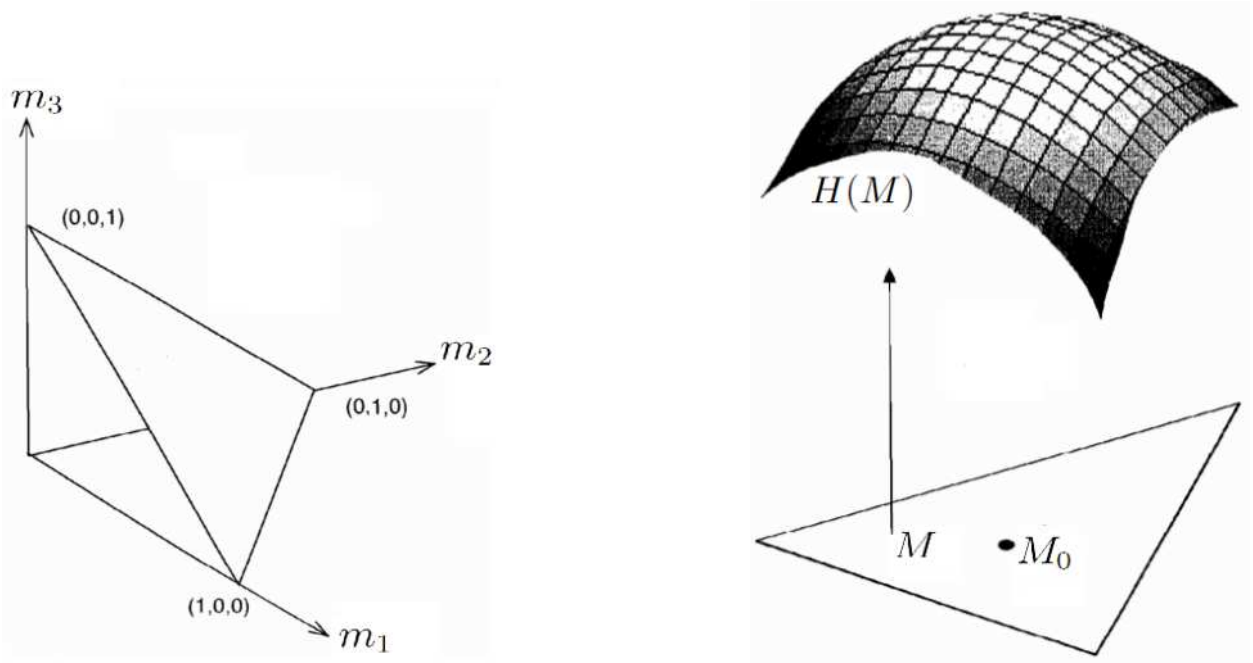
where n is the number of observations. Each component $m_i = \frac{1}{n}$ indicates that the mass is equally distributed among all observations.

This geometric framework helps visualize the impact of resampling on the distribution of mass and ultimately on the statistical estimates. The variability introduced by resampling can be interpreted as movements within this simplex, shifting the mass across different observations. This perspective is particularly useful in understanding the robustness and efficiency of the jackknife and bootstrap methods in various statistical applications.

Geometric Representation within the Simplex

As established in Definition 2.2.9, the statistic under resampling $H(M)$ depends on the resampling vector M, which lies within the n-dimensional simplex (cf. Definition 2.2.8). This simplex provides a geometric framework to visualize all possible distributions of mass among the observations X_i and to understand how different resampling methods explore this space. For the case of three observations ($n = 3$), the simplex becomes a two-dimensional equilateral triangle. Each point inside this triangle represents a unique resampling vector M, corresponding to a specific allocation of masses m_i to the observations X_i. Figures 2.2a and 2.2b illustrate this concept:

- **Figure 2.2a** shows the simplex as an equilateral triangle, with each vertex representing a resampling vector where all mass is placed on one observation ($m_i = 1$) and zero mass on the others ($m_j = 0$ for $j \neq i$). Points along the edges and inside the triangle represent resampling vectors with mass distributed among observations.
- **Figure 2.2b** depicts the surface of the statistic $H(M)$ over the simplex domain, illustrating how the value of the statistic changes with different mass distributions.



(a) Resampling vector M within the simplex

(b) Statistic $H(M)$ over the simplex domain

Figure 2.2: Geometric representation of resampling vectors and the statistic for $n = 3$. Adapted from [ET93, Chapter 20].

To understand how the jackknife and bootstrap methods relate to this simplex, we consider the specific resampling vectors they use. The definitions of these resampling vectors are as follows:

Definition 2.2.11: Jackknife Resampling Vectors $M_{\mathbf{JK}}$
The *jackknife resampling vectors* $M_{\mathrm{JK}(i)}$ are defined for each $i = 1, 2, \ldots, n$ as:

$$M_{\mathrm{JK}(i)} = \left(m_1^{(i)}, m_2^{(i)}, \ldots, m_n^{(i)}\right)^\top,$$

where

$$m_j^{(i)} = \begin{cases} \frac{1}{n-1}, & \text{if } j \neq i, \\ 0, & \text{if } j = i. \end{cases}$$

That is, $M_{\mathrm{JK}(i)}$ assigns zero weight to the i-th observation and equal weights $\frac{1}{n-1}$ to the remaining $n-1$ observations.

Definition 2.2.12: Bootstrap Resampling Vectors M_{boot}
The *bootstrap resampling vectors* M_{boot} are random vectors defined as:

$$M_{\text{boot}} = \left(m_1^{\text{boot}}, m_2^{\text{boot}}, \ldots, m_n^{\text{boot}}\right)^\top,$$

where M_{boot} follows the distribution:

$$M_{\text{boot}} \sim \frac{1}{n} \cdot \text{Multinomial}\,(n, W)\,,$$

with $W = (w_1, w_2, \ldots, w_n)^\top$ being an arbitrary probability vector satisfying:

$$w_i \geq 0 \quad \text{for all } i, \quad \sum_{i=1}^{n} w_i = 1.$$

Here, W represents the vector of probability weights associated with each of the n observations, indicating the likelihood of resampling each observation during the bootstrap process.

Each component m_i^{boot} represents the proportion of times observation X_i is selected in a bootstrap resample of size n and is given by:

$$m_i^{\text{boot}} = \frac{k_i}{n},$$

where $(k_1, k_2, \ldots, k_n)^\top$ is a realization from the multinomial distribution:

$$(k_1, k_2, \ldots, k_n)^\top \sim \text{Multinomial}\,(n, W)\,.$$

Thus, the bootstrap resampling vector M_{boot} assigns mass m_i^{boot} to observation X_i, based on the counts from the multinomial distribution scaled by $1/n$.

Figure 2.3 illustrates the resampling vectors used by the jackknife and bootstrap methods for $n = 3$:

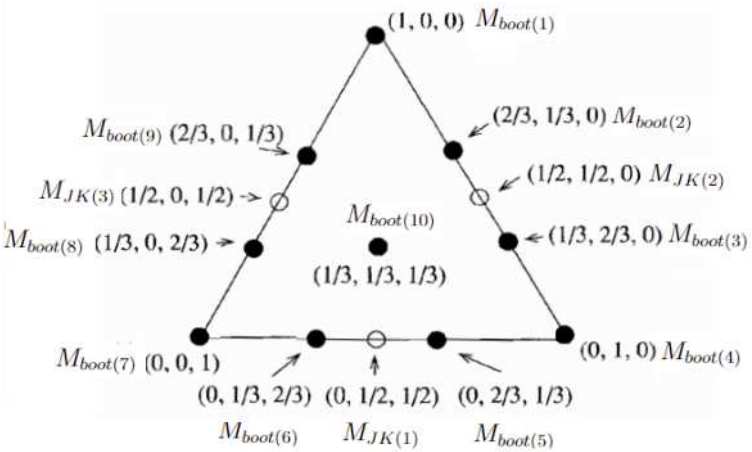


Figure 2.3: Geometric representation of resampling vectors used by the bootstrap (black dots) and jackknife (white dots) methods on a simplex for $n = 3$, laid flat on the page. Adapted from [ET93, Chapter 20].

In this figure:

- **Jackknife Method**: The resampling vectors $M_{\mathrm{JK}(i)}$ (white dots) correspond to the points where one observation is omitted (i.e., $m_i = 0$), and the remaining observations each receive equal mass $\frac{1}{n-1}$. These points lie at the medians of the simplex, reflecting the jackknife's systematic omission of each observation in turn.
- **Bootstrap Method**: The resampling vectors M_{boot} (black dots) represent all possible combinations of masses obtained by resampling n times with replacement from the observations. Each point corresponds to a specific allocation of masses m_i^{boot}, determined by the counts k_i from the multinomial distribution.

Let's consider an example to illustrate the definition of M_{boot} using $n = 3$. When resampling with replacement, the possible values of (k_1, k_2, k_3) are all combinations of non-negative integers summing to $n = 3$. The number of such combinations is given by the multiset coefficient:

$$\text{Number of combinations} = \binom{2n-1}{n} = \binom{5}{3} = 10.$$

These combinations correspond to the 10 black dots in Figure 2.3. Each dot represents a possible bootstrap resampling vector M_{boot} with components:

$$M_{\mathrm{boot}} = \left(\frac{k_1}{3}, \frac{k_2}{3}, \frac{k_3}{3}\right)^\top.$$

For example:

- If $(k_1, k_2, k_3) = (3, 0, 0)$, then $M_{\mathrm{boot}} = (1, 0, 0)^\top$. This corresponds to resampling X_1 three times.
- If $(k_1, k_2, k_3) = (1, 1, 1)$, then $M_{\mathrm{boot}} = \left(\frac{1}{3}, \frac{1}{3}, \frac{1}{3}\right)^\top$. This corresponds to each observation being selected once.
- If $(k_1, k_2, k_3) = (2, 1, 0)$, then $M_{\mathrm{boot}} = \left(\frac{2}{3}, \frac{1}{3}, 0\right)^\top$. This corresponds to X_1 being selected twice and X_2 once.

These examples demonstrate how the bootstrap resampling vectors M_{boot} cover more points within the simplex compared to the jackknife resampling vectors M_{JK}, which are limited to the medians.

Variance Estimation in Geometric Terms

The geometric perspective not only illustrates the differences in resampling vectors between the jackknife and bootstrap methods but also provides valuable insights into how these methods estimate variance.

In geometric terms, the ideal bootstrap variance estimate, denoted by $\hat{\mathrm{var}}_{\mathrm{boot}}^{\infty}(\hat{\theta})$, captures the spread of the statistic across the entire simplex, effectively serving as a "gold standard"

for variance estimation.[10]. It is derived by calculating the variance of the statistic under resampling (cf. Definition 2.2.9) with all possible bootstrap resampling vectors (cf. Definition 2.2.12).

Definition 2.2.13: Bootstrap variance estimate in geometric Terms

$$\hat{\text{var}}^{\infty}_{\text{boot}}(\hat{\theta}) = \text{var}\left(H\left(M_{\text{boot}}\right)\right)$$

It's important to note that the number of all possible resampling vectors for the bootstrap method grows combinatorially with n, specifically as $\binom{2n-1}{n}$. For $n = 3$, there are 10 possible bootstrap resampling vectors (cf. Figure 2.3), but as n increases, this number becomes impractically large to compute or enumerate. Consequently, in practice, a large but feasible number of bootstrap samples (B) is drawn to approximate the distribution of the statistic.

To further illustrate this concept, for $n = 3$ the ideal bootstrap variance estimate can be expressed as:

$$\begin{aligned}\hat{\text{var}}^{\infty}_{boot}\left(\hat{\theta}\right) &= \text{var}\left(H\left(M_{\text{boot}}\right)\right) \\ &= \sum_{k=1}^{10} p_k \left(H(M_{boot(k)}) - \left(\sum_{k=1}^{10} p_k H(M_{boot(k)})\right)\right)^2 \end{aligned} \tag{2.2.11}$$

where p_k is the probability of obtaining the resampling vector $M_{boot(k)}$ according to the multinomial distribution, based on the probabilitiy weights W (cf. Definition 2.2.12). For $n = 3$, this formula is exact, as it considers all possible resampling vectors without the need for approximation. The approximation would be:

$$\hat{\text{var}}^{\infty}_{boot}\left(\hat{\theta}\right) \approx \hat{\text{var}}^{B}_{boot}\left(\hat{\theta}\right) = \frac{1}{B-1}\sum_{i=1}^{B}(\hat{\theta}^{*b} - \overline{\hat{\theta}^{*}})^2, \tag{2.2.12}$$

where θ^{*b} is the the statistic of interest calculated on a bootstrap sample drawn with the probability weights W (cf. Section 2.2.2). The choice of the number of bootstrap replications B significantly impacts the accuracy of the bootstrap variance estimate $\hat{\text{var}}^{B}_{\text{boot}}$. While theoretically, an infinite number of replications ($B \to \infty$) would provide the most accurate estimate $\hat{\text{var}}^{\infty}_{\text{boot}}$, in practice, increasing B beyond a certain point yields diminishing returns. This is because $\hat{\text{var}}^{\infty}_{\text{boot}}$ itself varies with the observed sample X, introducing inherent randomness into any variance estimate. According to Efron and Tibshirani [ET93, Chapter 6.4], $B = 200$ is generally sufficient for reliable variance estimation. For constructing confidence intervals, much larger values of B are typically required to ensure accuracy in the tails of the bootstrap distribution.

The jackknife variance estimate, $\hat{\text{var}}_{\text{JK}}\left(\hat{\theta}\right)$, can be seen as an approximation of the bootstrap variance. It has been shown that for linear statistics:

[10][ET93, p. 287]

Theorem 2.2.14
For linear statistics (cf. Definition 2.2.2), the jackknife variance estimate (cf. Definition 2.2.1) is a scaled version of the bootstrap variance estimate (cf. Definition 2.2.5). Specifically, we have:[a]

$$\hat{\text{var}}_{\text{JK}}\left(\hat{\theta}\right) = \frac{n}{n-1}\hat{\text{var}}_{\text{boot}}(H^{\text{LIN}})$$

where H^{LIN} is the linear hyperplane approximation of $H(M)$ (cf. Definition 2.2.9) passing through the n jackknife points $H(M_{\text{JK}(i)})$

[a] [Efr82, p. 39 f.]

The scaling factor $\frac{n}{n-1}$ in the above Theorem not only ensures that $\hat{\text{var}}_{\text{JK}}\left(\hat{\theta}\right)$ is nearly unbiased for linear statistics, but also helps to correct for the slight downward bias that the bootstrap variance estimate can exhibit in such cases.

Why does a linear statistic results in a linear hyperplane? In the geometric representation of resampling methods, we consider the statistic $\hat{\theta}$ as a function of masses $M = (m_1, m_2, \ldots, m_n)$ assigned to the observations X_i (cf. Definition 2.2.9). Then the linear statistic can be rewritten as:

$$H(M) = c + \sum_{i=1}^{n} m_i \alpha(X_i), \tag{2.2.13}$$

where c is a constant $\alpha(\cdot)$ is any function. By recognizing that the linearity in the definition of a linear statistic refers to its dependence on the masses m_i, we understand why a linear hyperplane arises geometrically from a linear statistic.

Figure 2.4 provides a geometric view (for $n = 3$) of the linear hyperplane approximation of H. The curved surface $H(M)$ represents the true behavior of the statistic over all possible resampling vectors (cf. Definition 2.2.7). The linear hyperplane H^{LIN} passes through these jackknife points $H(M_{\text{JK}(i)})$, providing a linear approximation of the statistic's behavior. The jackknife variance estimate corresponds to the variability of the statistic on this linear hyperplane, while the bootstrap variance estimate reflects the variability across the entire simplex, including regions not captured by the linear hyperplane. This difference in scope explains why the jackknife variance struggles with non-linear statistics, as it does not capture the full range of variability that the bootstrap method accounts for.

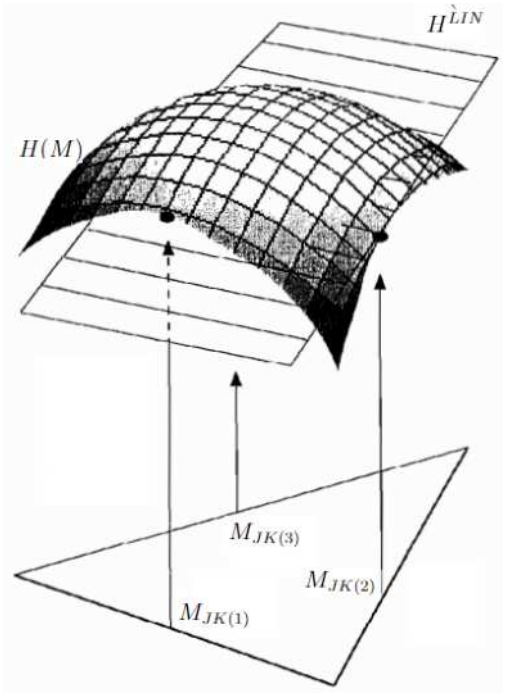


Figure 2.4: The hyperplane approximation H^{LIN} through the jackknife points $H(M_{\text{JK}(i)})$ in the simplex for $n = 3$. Adapted from [ET93, Chapter 20].

Alternative Approximations

Given the limitations of the jackknife method for nonlinear statistics, alternative approximations that better capture the complexity of such statistics are needed. One such alternative is the tangent-plane approximation at $H(M_0)$, as illustrated in Figure 2.5. Unlike the linear hyperplane approximation, this method accounts for the curvature of $H(M)$ around the central point M_0, offering a potentially more accurate representation of nonlinear behavior. This method is called the Infinitesimal Jackknife.

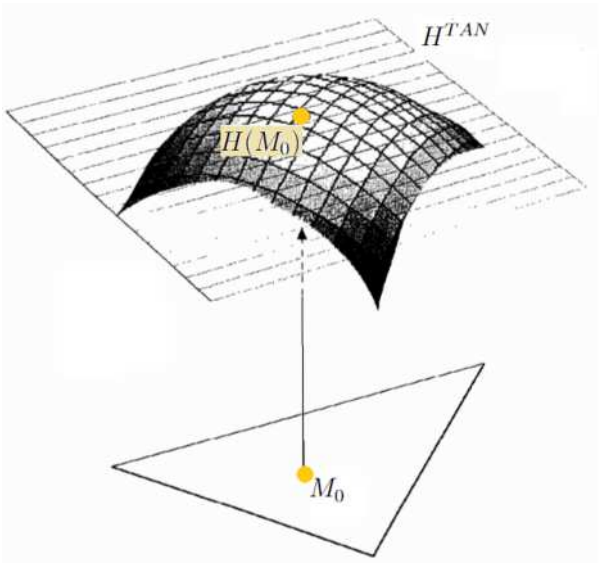


Figure 2.5: Tangent-plane approximation at $H(M_0)$. Adapted from [ET93, Chapter 20].

The geometric interpretation of the jackknife and bootstrap methods illuminates how these resampling techniques redistribute mass within the simplex and how this redistribution impacts variance estimation. By understanding the geometric properties of these methods, we can make more informed decisions about which technique is most appropriate for estimating the variability of a given statistic. While the jackknife is efficient for linear statistics and the bootstrap is better suited for complex nonlinear cases, the tangent-plane approach provides a promising alternative that strikes a balance between the simplicity of the jackknife and the robustness of the bootstrap, making it particularly suitable for nonlinear statistics. In situations where both accuracy and computational cost are critical, the tangent-plane approximation offers a compromise between accuracy and computational efficiency.

In the next section, we will explore the Infinitesimal Jackknife in greater detail, assessing its advantages and limitations, especially for nonlinear statistics. This exploration will enhance our understanding of variance estimation and improve the accuracy of statistical inferences across various applications.

2.2.4 Infinitesimal Jackknife

Building on the limitations identified in the classical jackknife method for nonlinear statistics, the *infinitesimal jackknife* offers a more refined approach to variance estimation. This method, proposed by Jaeckel[11], enhances our ability to assess the sensitivity of a statistic to small changes in the data distribution, making it particularly valuable for complex and nonlinear scenarios.

Influence Function

At the core of the infinitesimal jackknife is the concept of the *influence function* U_i. Unlike the classical jackknife, which assesses the impact of completely removing an sample X_i, the infinitesimal jackknife evaluates how the statistic reacts to an infinitesimal change in the mass m_i of a sample X_i. This nuanced approach allows for a more precise analysis of each observation's influence on the overall statistic, particularly in cases where the statistic may not respond linearly to changes in the data.

In Chapter 2.2.1, we discussed the assumption in the jackknife method that the statistic $\hat{\theta}$ must be smooth. Now, we can express this requirement more formally:

Definition 2.2.15: Smooth Statistic in the Context of Resampling Methods
A statistic $\hat{\theta} = H(M_0)$ is considered a *smooth function* if the function $H(M)$ is continuously differentiable with respect to the masses m_i. This means that for each $i = 1, 2, \ldots, n$:[a]

$$\frac{\partial H}{\partial m_i}(M) \text{ exists and is continuous in a neighborhood of } M_0.$$

[a] [ES81, p. 590f.]

This smoothness ensures that $\hat{\theta}^{(M)}$ can be consistently calculated based on the weighted

[11] [Jae72]

empirical probability distribution, which is a critical requirement for the validity of the jackknife and the infinitesimal jackknife method.

The influence function U_i for the infinitesimal jackknife method is defined as the directional derivative of the statistic H at M in the direction of a point mass at the i-th data sample X_i. Essentially, it measures the sensitivity of the estimator to small perturbations at a particular data sample, offering insights into the robustness of the statistic. The formal definition is given by:

Definition 2.2.16: Influence Function
The influence function $U(X_i)$ is defined as:[a]

$$U(X_i) = \lim_{\varepsilon \to 0} \frac{H\left((1-\varepsilon)M + \varepsilon e_i\right) - H(M)}{\varepsilon},$$

where e_i denotes the unit vector placing all mass on the i-th observation, and H is the statistic as a function of the mass vector M.

A key property of the influence function is that it is centered with respect to M:[b]

$$\frac{1}{n}\sum_{i=1}^{n} U(X_i) = 0$$

[a] [Efr82, p. 40]
[b] [ET93, p. 300]

When the data samples are unweighted, meaning $M = M_0$ with $m_i = \frac{1}{n}$ for all i, this expression simplifies to:

$$U(X_i) = \lim_{\varepsilon \to 0} \frac{H\left(\frac{1-\varepsilon}{n}, \ldots, \frac{1-\varepsilon}{n} + \varepsilon, \ldots, \frac{1-\varepsilon}{n}\right) - H(M_0)}{\varepsilon}, \tag{2.2.14}$$

where the i-th component of the mass vector is $\frac{1-\varepsilon}{n} + \varepsilon$, and all other components are $\frac{1-\varepsilon}{n}$. This simplification illustrates how the influence function quantifies the impact of each observation under the assumption of uniform weights. By providing a finer-grained analysis of influence, the infinitesimal jackknife method allows for more accurate variance estimation, particularly in scenarios where the classical jackknife may struggle.

Variance Estimation with Influence Functions

To deepen our understanding of how the influence function relates to the variance estimation of the statistic $\hat{\theta}$, we consider the following fundamental result from the theory of influence functions:

Theorem 2.2.17
When a distribution G is "close" to F, meaning e.g. that G is the empirical distribution based on a large sample from F, the statistic can be approximated by:[a]

$$h(G) = h(F) + \int U(x)\, dG(x)$$

where:

- $h(G) = \hat{\theta}$ is the calculated statistic based on the distribution G,
- $h(F) = \theta$ is the statistic under the true distribution F,
- $U(x)$ is the influence function of the statistic at the point x with respect to the distribution F.

[a] [Ham+86, pp. 85–86]

With the above theorem, we can now define the variance estimate of a statistic, calculated on a dataset $\mathbf{X} = (X_1, \ldots, X_n)$ with the weighted empirical distribution G, through:

$$\begin{aligned} \operatorname{var}\left(\hat{\theta}\right) &= \operatorname{var}\left(\theta + \int U(x)\, dG(x)\right) \\ &= \operatorname{var}\left(\int U(x)\, dG(x)\right) \\ &= \operatorname{var}\left(\sum_{i=1}^{n} m_i U(X_i)\right) \\ &= \sum_{i=1}^{n} m_i^2 \operatorname{Var}\left(U(X_i)\right), \end{aligned} \tag{2.2.15}$$

where m_i indicates the mass that sample X_i receives in calculating the statistic $\hat{\theta}$. Since the observations X_i are independent and identically distributed, we can further simplify the above equation:

$$\begin{aligned} \operatorname{var}\left(\hat{\theta}\right) &= \left(\sum_{i=1}^{n} m_i^2\right) \operatorname{var}\left(U(X)\right) \\ &\approx \left(\sum_{i=1}^{n} m_i^2\right) \hat{\operatorname{var}}\left(U(X)\right) \\ &= \left(\sum_{i=1}^{n} m_i^2\right) \sum_{i=1}^{n} m_i \left(U(X_i) - \bar{U}\right)^2, \end{aligned} \tag{2.2.16}$$

where $\bar{U} = \frac{1}{n}\sum_{i=1}^{n} U(X_i)$. Since the mean of the influence function over all samples X_i is zero (cf. Definition 2.2.16), i.e., $\bar{U} = 0$, we can now define the infinitesimal jackknife variance estimate as:

Definition 2.2.18: Infinitesimal Jackknife Variance Estimate
The infinitesimal jackknife variance estimate of the statistic $\hat{\theta}$ is defined as:

$$\hat{\text{var}}_{\text{IJK}}\left(\hat{\theta}\right) = \left(\sum_{i=1}^{n} m_i^2\right) \sum_{i=1}^{n} m_i U(X_i)^2,$$

where $U(X_i)$ are the influence functions (cf. Definition 2.2.16) evaluated at each observation X_i, and m_i are the masses assigned to the observations to calculate the statistic $\hat{\theta}$.

When the data samples X_i are unweighted to calculate the statistic $\hat{\theta}$, meaning $m_i = \frac{1}{n}$ for all $i = 1, \ldots, n$, the above definition simplifies to:

$$\begin{aligned} \hat{\text{var}}_{\text{IJK}}\left(\hat{\theta}\right) &= \left(\sum_{i=1}^{n} \frac{1}{n^2}\right) \sum_{i=1}^{n} \frac{1}{n} U(X_i)^2 \\ &= \frac{1}{n^2} \sum_{i=1}^{n} U(X_i)^2, \end{aligned} \tag{2.2.17}$$

which is the classical infinitesimal jackknife estimate for the variance of a statistic calculated on unweighted data samples.[12]

By adjusting the value of ε in the influence function, we can derive the jackknife variance estimate (cf. Definition 2.2.1) as well. If we set $\varepsilon = \frac{-1}{n-1}$ and consider unweighted data samples, the influence function (cf. Definition 2.2.16) becomes:

$$\begin{aligned} U(X_i) &= \lim_{\varepsilon \to \frac{-1}{n-1}} \frac{H\left((1-\varepsilon)M_0 + \varepsilon e_i\right) - H(M_0)}{\varepsilon} \\ &= \frac{H\left(\frac{1}{n-1}, \ldots, 0, \ldots, \frac{1}{n-1}\right) - H(M_0)}{\frac{-1}{n-1}}, \end{aligned} \tag{2.2.18}$$

where the i-th component of the mass vector is 0, and all other components are $\frac{1}{n-1}$. Using this influence function and Theorem 2.2.17, we can estimate the variance of the statistic $\hat{\theta}$:

$$\begin{aligned} \hat{\text{var}}\left(\hat{\theta}\right) &= \frac{1}{n^2} \sum_{i=1}^{n} U(X_i)^2 \\ &= \frac{1}{n^2} \sum_{i=1}^{n} \left((n-1)\left(\hat{\theta}_{(i)} - \hat{\theta}\right)\right)^2 \\ &= \frac{(n-1)^2}{n^2} \sum_{i=1}^{n} \left(\hat{\theta}_{(i)} - \hat{\theta}\right)^2, \end{aligned} \tag{2.2.19}$$

which is very similar, especially for large n, to the jackknife estimate of variance (cf. Definition 2.2.1).

[12] [Efr82, p. 41]

By examining how the choice of ε affects the influence functions in both methods, we can see how this impacts the variance estimation:

- **Infinitesimal Jackknife** ($\varepsilon \to 0$): Captures the local behavior of the statistic, accounting for small, incremental changes. This is particularly important for nonlinear statistics, where the relationship between the data and the estimator may not be well-approximated by finite differences.
- **Classical Jackknife** ($\varepsilon = \frac{-1}{n-1}$): Uses a finite change corresponding to the removal of an observation. While effective for linear statistics, it may not capture the curvature or higher-order interactions present in nonlinear statistics.

For nonlinear statistics, small perturbations may have different effects compared to larger ones due to the curvature in the estimator's functional form. The infinitesimal jackknife's focus on infinitesimal changes allows it to better capture this local sensitivity, leading to more accurate variance estimates.

However, it is important to note that in practice, a slight downward bias is often observed. This bias arises from the approximation methods used in calculating the influence function $U(X_i)$, particularly in finite samples and with complex satistics. Empirical studies have shown that this bias tends to result in variance estimates that are slightly lower than their true values.[13] To address this, methods like the bootstrap estimate of variance (cf. Definition 2.2.5), which can accommodate the full distributional complexity by resampling the data, may provide better variance estimates. Or bias correction techniques can be applied to bring the estimates closer to their true values.

Conversely for linear statistics, the infinitesimal jackknife estimate is the same as the ideal bootstrap variance estimate:

Theorem 2.2.19
For linear statistics (cf. Definition 2.2.2), the infinitesimal jackknife estimate (cf. Definition 2.2.18) is equally to the ideal bootstrap estimate (cf. Definition 2.2.13) and a scaled version of the jackknife estimate (cf. Definition 2.2.1): [a]

$$\hat{\mathrm{var}}_{\mathrm{IJK}}\left(\hat{\theta}\right) = \hat{\mathrm{var}}^{\infty}_{boot}(\hat{\theta}) = \frac{n-1}{n}\hat{\mathrm{var}}_{\mathrm{JK}}\left(\hat{\theta}\right)$$

[a] [ET93, p. 302]

Practical Computation

In practice, evaluating the limit in $U(X_i)$ (cf. Definiton 2.2.16) is often infeasible. Therefore, $U(X_i)$ is typically approximated numerically by choosing a small value for ε, such as 10^{-6}, and computing:

$$U(X_i) \approx \frac{H\left((1-\varepsilon)M_0 + \varepsilon e_i\right) - H(M_0)}{\varepsilon}. \tag{2.2.20}$$

This numerical approximation allows for practical computation of the infinitesimal jackknife variance estimator, even for complex or nonlinear statistics.

[13] [Efr82, p. 42]

Infinitesimal Jackknife - Example

To illustrate the infinitesimal jackknife method in practice, consider the case where the satistic $\hat{\theta}$ is the sample mean $\bar{X}$, with $X_i \in \mathbb{R}^1$. In this scenario, the influence function $U(X_i)$ can be explicitly calculated as follows:

$$U(X_i) = \lim_{\varepsilon \to 0} \frac{H\left(\frac{1-\epsilon}{n}, \ldots, \frac{1-\epsilon}{n}, \frac{1-\epsilon}{n} + \epsilon, \frac{1-\epsilon}{n}, \ldots, \frac{1-\epsilon}{n}\right)^T - H(M_0)}{\epsilon}, \tag{2.2.21}$$

which simplifies to:

$$U_i = \lim_{\varepsilon \to 0} \frac{\sum_{j=1}^n \frac{1-\epsilon}{n} x_j - \frac{1-\epsilon}{n} x_i + (\frac{1-\epsilon}{n} + \epsilon) x_i - \frac{1}{n} \sum_{j=1}^n x_j}{\epsilon} = x_i - \bar{x}. \tag{2.2.22}$$

Given this influence function, the infinitesimal jackknife variance estimator for the sample mean can be computed as:

$$\hat{var}_{IJK}(\hat{\theta}) = \frac{1}{n^2} \sum_{i=1}^n U_i^2 = \frac{1}{n^2} \sum_{i=1}^n (x_i - \bar{x})^2 = \frac{n-1}{n} \frac{\hat{\sigma}_{\mathbf{X}}^2}{n}.$$

This expression shows that the infinitesimal jackknife variance for the sample mean is closely related to the classical Jackknife variance $\hat{var}_{JK}(\hat{\theta}) = \frac{\hat{\sigma}_{\mathbf{X}}^2}{n}$ (cf. Example 1). Specifically, the relationship between the two can be expressed as:

$$\hat{var}_{IJK}(\hat{\theta}) = \frac{n-1}{n} \hat{var}_{JK}(\hat{\theta}), \tag{2.2.23}$$

demonstrating that the infinitesimal jackknife variance is a scaled version of the classical jackknife variance for the sample mean, a linear statistic.

This example demonstrates how the infinitesimal jackknife can be computed exactly for simpler statistics like the sample mean. However, for more complex statistics such as the Pearson correlation coefficient or the median, the calculation of the influence function can become algebraically complicated. Therefore, in practice, these are often estimated numerically. To illustrate this, we will present small simulations in the next section.

2.2.5 Simulations

In this section, we present simulation studies that validate the theoretical concepts discussed earlier. The simulations focus on three commonly used statistics: the mean (a linear statistic), the Pearson correlation (a non-linear statistic), and the median (a non-smooth statistic). Each simulation compares the performance of different variance estimation methods, specifically the jackknife, bootstrap, and infinitesimal jackknife, in alignment with the theory discussed in previous sections. The relative errors reported in the following sections are mean errors over the simulations, and the corresponding boxplots of these relative errors are shown in Figures 2.6, 2.7, and 2.8.

Variance for Mean Estimation

The first simulation examines the variance estimation for the mean, a linear statistic, under the assumption that the data follows a standard normal distribution. This choice of statistic

aligns with the discussions in Example 1, where the jackknife method is expected to perform well due to the linear nature of the mean. Moreover, as discussed in Section 2.2.1, the jackknife method is known to provide unbiased estimates in expectation for linear statistics (cf. Theorem 2.2.3), making it particularly effective in this scenario.

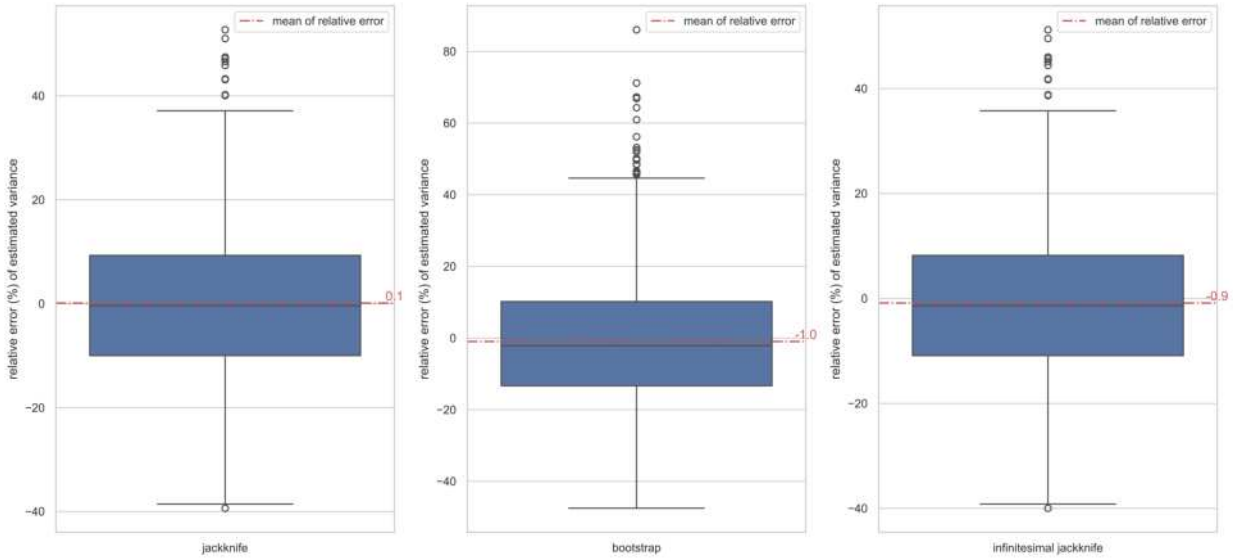


Figure 2.6: Variance estimates for $\hat{\theta} = mean(\mathbf{X})$, underlying data is $X \sim N(0,1)$. The boxplots contain the estimates over 2000 simulations (each with $n = 100$) and for the bootstrap method $B = 200$ was used. Relative errors are calculated with the true variance $\text{var}(\hat{\theta}) = \frac{\sigma_X}{n}$.

Results showed that the jackknife method achieved a mean relative error of 0.1%, validating its accuracy and unbiasedness for linear statistics (cf. Figure 2.6). The bootstrap method had a slightly *downward bias*, the mean relative error is -1%. This slight negative bias also supports the equation in Theorem 2.2.14, where the bootstrap variance estimate was noted to exhibit a slight downward bias in cases, where the statistic in linear. The infinitesimal jackknife showed a mean relative error of -0.9%, which is similar to that of the bootstrap, indicating their close relationship (cf. Theorem 2.2.19). The values are not identical because, in the bootstrap method, the number of bootstrap samples B is finite (here $B = 200$) rather than infinite.

Variance for Pearson Correlation Estimation

Next, we evaluated the performance of the variance estimation methods for the Pearson correlation coefficient, a non-linear statistic. As discussed in Section 2.2.1, the jackknife method may exhibit an upward bias when applied to non-linear statistics, particularly due to its reliance on linear approximations (cf. Theorem 2.2.4).

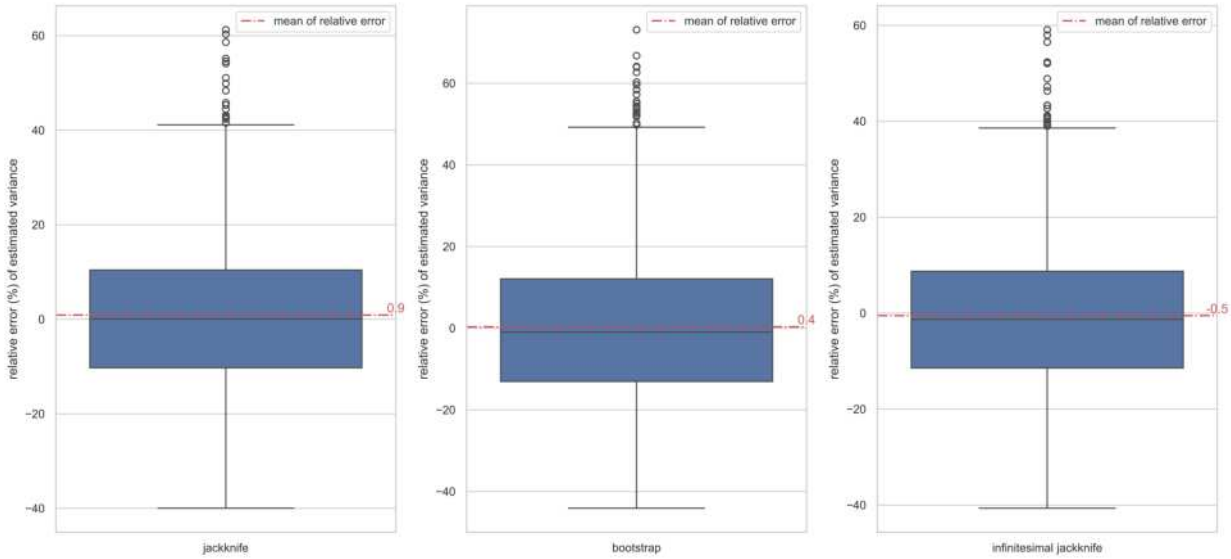


Figure 2.7: Variance estimates for $\hat{\theta} = pearsoncorrelation(\mathbf{X})$, underlying data is $(X, Y) \sim N\left(\begin{pmatrix}0\\0\end{pmatrix}, \begin{pmatrix}1 & 0.7\\0.7 & 1\end{pmatrix}\right)$. The Boxplots contain the estimates over 2000 simulations (each with $n = 400$) and for the bootstrap method $B = 200$ was used. Relative errors are calculated with the true variance $\text{var}(\hat{\theta}) = \frac{(1-\rho^2)^2}{n}$ (cf. [Bow28, p. 31]).

In this case, the jackknife method had a mean relative error of 0.9%, while the bootstrap method performed better with a mean relative error of 0.4%, and the infinitesimal jackknife showed a mean relative error of -0.5%. These results (see Figure 2.7) confirm the theoretical prediction of slight upward bias in the jackknife estimates for non-linear statistics. The bootstrap method, as expected, provided more accurate estimates, consistent with its theoretical advantages mentioned earlier in Section 2.2.2 . Additionally, the infinitesimal jackknife outperformed the jackknife for non-linear statistics and showed a slight downward bias, as discussed in Section 2.2.4.

Variance for Median Estimation

Finally, we explored variance estimation for the median, a non-smooth statistic, which presents a significant challenge for the jackknife and infinitesimal jackknife method. As predicted in Example 3 and Section 2.2.4, both methods failed to provide reliable estimates for the median due to its non-smooth nature.

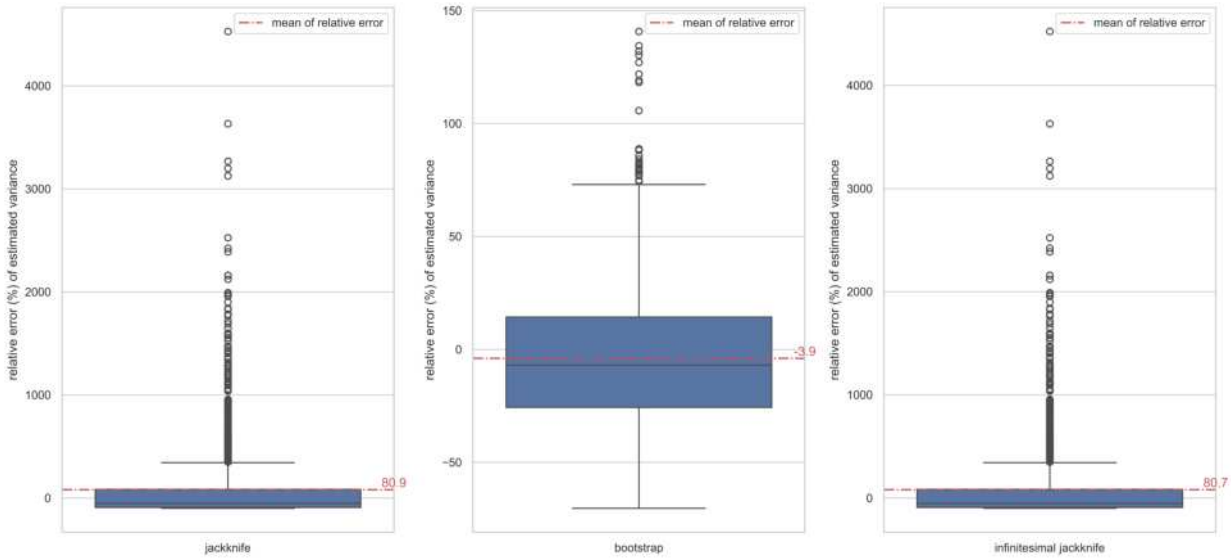


Figure 2.8: Variance estimates for $\hat{\theta} = median(\mathbf{X})$, underlying data is $X \sim N(0, 1)$. The Boxplots contain the estimates over 2000 simulations (each with $n = 400$ and for the bootstrap method $B = 200$ was used. Relative errors are calculated with the empirical variance of $\hat{\theta}$ from the 2000 simulations.

The simulation results (see Figure 2.8) showed a mean relative error of 80.9% for the jackknife method, illustrating its inadequacy for this statistic. Similarly, the infinitesimal jackknife had a high mean relative error of 80.7%. In contrast, the bootstrap method, known for its flexibility in dealing with non-smooth statistics, provided a much more reasonable approximation with a mean relative error of -3.9%. These findings align with the theoretical discussions in Section 2.2.2, where the bootstrap's robustness for non-linear and non-smooth statistics was emphasized .

The simulation results confirm that the jackknife method is highly accurate and nearly unbiased for linear statistics, as shown by its minimal error in mean estimation. However, it struggles with non-linear and non-smooth statistics, evidenced by its high error in median estimation. In contrast, the bootstrap method consistently provides reliable variance estimates across various statistics. The infinitesimal jackknife, performing similarly to the bootstrap for linear statistics and better than the jackknife for non-linear statistics, is a valuable extension. Nevertheless, its slight downward bias highlights the importance of applying bias correction techniques in practice.

2.3 Nonparametric Variance Estimates for Bagged Learners — Under Unweighted Resampling

In the previous chapter, we explored various methods for variance estimation, including the Jackknife, Bootstrap, and Infinitesimal Jackknife techniques. These methods provide valuable insights into the stability and reliability of statistical estimates by resampling data and assessing how changes in the dataset affect the estimates. Building on this foundation, we now focus on variance estimates for Bagged Learners (BL) under unweighted resampling.

In the following sections, we will first provide a comprehensive overview of Bagged Learners. Subsequently, we will introduce and evaluate two nonparametric variance estimation techniques—Jackknife and Jackknife-after-Bootstrap—to assess their effectiveness in estimating the variance of a Bagged Learner under unweighted resampling.

2.3.1 Bagged Learner (BL)

Bagged Learners, or Bootstrap Aggregating, are designed to reduce variance, improve robustness, and handle overfitting, particularly in high-variance models. By combining the predictions of multiple models generated from different bootstrap samples, bagging can stabilize predictions and prevent overfitting, making it particularly valuable in complex or noisy data environments. The process begins with a training dataset, denoted as:

Definition 2.3.1: Training Dataset

$$\mathbf{X} = (X_1, \ldots, X_n)^\top = ((x_1, y_1), \ldots, (x_n, y_n))^\top,$$

where $\mathbf{X}$ consists of input-output pairs (x_i, y_i).

The prediction of a base learner, trained on the dataset $\mathbf{X}$, for a given input x is represented as:

Definition 2.3.2: Base Learner's Prediction Function
The base learner's prediction function $t(x; \mathbf{X})$ provides a prediction $\hat{\theta}(x)$ for a given input x, based on the training dataset $\mathbf{X}$:

$$\hat{\theta}(x) = t(x; \mathbf{X}).$$

The goal of bagging is to stabilize the base learner t through resampling. This involves generating multiple bootstrap datasets, denoted as $\{\mathbf{X}^{*1}, \ldots, \mathbf{X}^{*B}\}$, and using them to create an ensemble of learners.

Definition 2.3.3: Bootstrap Samples and Count Vectors
The bootstrap samples $\{\mathbf{X}^{*1}, \ldots, \mathbf{X}^{*B}\}$ are datasets obtained by sampling with replacement from the original training dataset $\mathbf{X} = (X_1, X_2, \ldots, X_n)^\top$ using probability weights $W = (w_1, w_2, \ldots, w_n)^\top$, where $w_i \geq 0$ and $\sum_{i=1}^n w_i = 1$.

Each bootstrap sample $\mathbf{X}^{*b}$ consists of n observations drawn from $\mathbf{X}$. The number of times each observation X_i appears in the bootstrap sample $\mathbf{X}^{*b}$ is recorded in the count vector $N^{*b} = (N_1^{*b}, N_2^{*b}, \ldots, N_n^{*b})^\top$, where:

$$N_i^{*b} = \text{number of times } X_i \text{ appears in } \mathbf{X}^{*b}.$$

The count vector N^{*b} follows a multinomial distribution:

$$N^{*b} \sim \text{Mult}\,(n, W)\,.$$

The ideal bagged learner, which represents the expectation of the base learner over all possible bootstrap samples, is defined as:

Definition 2.3.4: Ideal Bagged Learner
The ideal bagged learner is defined as the expectation of the base learner over all possible bootstrap samples:

$$\hat{\theta}^{\infty}(x) = E_* \left[t(x; \mathbf{X}^*) \right],$$

where $\mathbf{X}^*$ represents a bootstrap sample drawn from the original dataset $\mathbf{X}$. The expectation E_* is taken with respect to the probability distribution of $\mathbf{X}^*$, which is determined by the resampling scheme using the probability weights W.

This ideal scenario is equivalent to letting the number of bootstrap samples, B, approach infinity. However, in most cases, this ideal bagged learner cannot be evaluated directly. Instead, we can approximate it using Monte Carlo methods. This approximation, which is practical for implementation, is given by:

Definition 2.3.5: Approximated Bagged Learner
The approximated bagged learner with B bootstrap samples is defined as:

$$\hat{\theta}^{B}(x) = \frac{1}{B} \sum_{b=1}^{B} t(x; \mathbf{X}^{*b}) \quad \text{or equivalently} \quad \hat{\theta}^{B}(x) = \frac{1}{B} \sum_{b=1}^{B} T(x; N^{*b}),$$

where:

- $T(x; N^{*b})$ is the prediction of the base learner T at input x, trained using the count vector N^{*b} associated with the bootstrap sample.

The functions $t()$ and $T()$ differ in how they utilize the bootstrap samples:

- $t()$ operates directly on the resampled dataset $\mathbf{X}^{*b}$.
- $T()$ uses the count vector N^{*b} to weight the original observations $\mathbf{X}$ accordingly.

This approximation enables bagging to combine predictions from multiple base learners, each trained on different bootstrap samples, resulting in a more stable and accurate final model. This reduction in variance is particularly valuable in complex or noisy data environments, as bagging enhances model robustness and mitigates overfitting.[14] Moreover, by averaging predictions across multiple resampled datasets, bagging inherently smooths the estimator. This smoothness is crucial for the validity of the jackknife and infinitesimal jackknife methods, as it ensures the consistent calculation of the statistic (cf. Section 2.2.4). Consequently, bagged learners naturally satisfy the smoothness requirements (cf. Definition 2.2.15).

In the next sections, we will discuss how we can estimate the prediction uncertainty of bagged learners. Understanding this uncertainty is crucial, as it helps assess the reliability of the model's predictions and guides decision-making, especially in high-stakes scenarios.

[14] [Bre96]

2.3.2 Jackknife for BL

The Jackknife procedure can also be employed to estimate the variance of a bagged learner $\hat{\theta}^B(x)$ under unweighted resampling. This method follows a similar approach to the classical Jackknife (cf. Definition 2.2.1) but adapts it to the bagging context. The steps are as follows:

1. Leave out sample X_i from $\mathbf{X}$ and generate new bootstrap datasets $\{\mathbf{X}^{(i)*1}, ..., \mathbf{X}^{(i)*B}\}$.
2. Calculate the new bagged learner estimate $\hat{\theta}^{B(i)}(x)$ on the new bootstrap datasets.
3. The Jackknife variance estimate for the bagged learner $\hat{\theta}^B(x)$ is then given by:

$$\hat{var}_{JK}\left(\hat{\theta}^B(x)\right) = \frac{n-1}{n}\sum_{i=1}^{n}\left(\hat{\theta}^{B(i)}(x) - \hat{\theta}^{B(\cdot)}(x)\right)^2, \quad \text{with } \hat{\theta}^{B(\cdot)}(x) = \frac{1}{n}\sum_{i=1}^{n}\hat{\theta}^{B(i)}(x) \tag{2.3.1}$$

However, this approach requires generating B bootstrap datasets for each $\hat{\theta}^{B(i)}(x)$, which can be computationally intensive. In total, this process necessitates generating $n \times B$ bootstrap samples and training $n \times B$ base learners for variance estimation, resulting in a significant increase in computational overhead. To overcome this challenge, an alternative approach called the *Jackknife-after-Bootstrap* was introduced by Efron in 1992, presented in the next section.

2.3.3 Jackknife-after-Bootstrap for BL

The *Jackknife-after-Bootstrap* method leverages a key result regarding the distribution of bootstrap samples when an observation is left out.

Theorem 2.3.6
Let $\mathbf{X} = \{X_1, X_2, \ldots, X_n\}$ be a dataset consisting of n observations. Consider generating bootstrap samples by sampling with replacement from $\mathbf{X}$. Let $\mathbf{X}^{(i)} = \mathbf{X} \setminus \{X_i\}$ denote the dataset with the i-th observation removed.

Then, the distribution of bootstrap samples drawn with replacement from $\mathbf{X}^{(i)}$ is identical to the distribution of bootstrap samples drawn with replacement from the full dataset $\mathbf{X}$, conditioned on the event that X_i does not appear in the sample. Mathematically, we have:[a]

$$\text{Distribution of } \mathbf{X}^{(i)*} \equiv \text{Distribution of } \left(\mathbf{X}^* \mid X_i \notin \mathbf{X}^*\right),$$

where:

- $\mathbf{X}^{(i)*}$ is a bootstrap sample drawn with replacement from $\mathbf{X}^{(i)}$.
- $\mathbf{X}^*$ is a bootstrap sample drawn with replacement from the full dataset $\mathbf{X}$.

[a] [Efr92, p. 89]

This Theorem allows us to compute leave-one-out estimates $\hat{\theta}^{B(i)}(x)$ without generating new bootstrap samples from the reduced dataset $\mathbf{X}^{(i)}$. Instead, we can use the existing bootstrap

samples from the full dataset $\mathbf{X}$ and consider only those samples where X_i does not appear. Using the Theorem 2.3.6, we define the Jackknife-after-Bootstrap variance estimate for bagged learners as follows:

Definition 2.3.7: Jackknife-after-Bootstrap Variance Estimate for Bagged Learners
Let $\hat{\theta}^B(x) = \frac{1}{B}\sum_{b=1}^{B} T(x; N^{*b})$ be the bagged learner's prediction at input x, where $T(x; N^{*b})$ is the prediction of the base learner trained on the b-th bootstrap sample represented by the count vector N^{*b}.

The Jackknife-after-Bootstrap variance estimate of $\hat{\theta}^B(x)$ is given by:

$$\hat{\text{var}}_{\text{JAB}}\left(\hat{\theta}^B(x)\right) = \frac{n-1}{n}\sum_{i=1}^{n}\left(\hat{\Delta}_i\right)^2,$$

where:

- $\hat{\Delta}_i = \hat{\theta}^{B(i)}(x) - \hat{\theta}^B(x)$ is the difference between the leave-one-out estimate and the overall bagged estimate.
- $\hat{\theta}^{B(i)}(x) = \frac{1}{B_i}\sum_{\{b=1, N_i^{*b}=0\}} T(x; N^{*b})$ is the averaged prediction over all bootstrap samples where the i-th observation X_i does not appear, with $B_i = |\{b : N_i^{*b} = 0\}|$ being the number of such samples.
- N_i^{*b} is the number of times X_i appears in the b-th bootstrap sample.

If $B_i = 0$ (i.e., X_i appears in all bootstrap samples), we set $\hat{\Delta}_i = 0$.

By utilizing the bootstrap samples where X_i is absent, we can efficiently compute the leave-one-out estimates $\hat{\theta}^{B(i)}(x)$ without the need for additional resampling. This significantly reduces computational effort compared to the naive approach.

Bias Correction for Jackknife-after-Bootstrap
When estimating variance using resampling methods like the Jackknife-after-Bootstrap (Definition 2.3.7), a finite number of bootstrap samples B can introduce bias into the variance estimate. This bias arises because the estimator based on a finite B differs from the ideal estimator as $B \to \infty$.

In general, the bias of an estimator $\hat{\theta}$ is defined as:

$$\text{Bias}(\hat{\theta}) = E[\hat{\theta}] - \theta, \tag{2.3.2}$$

where $E[\hat{\theta}]$ is the expected value of the estimator, and θ is the true parameter. For variance estimation, the bias of the variance estimator $\hat{\text{var}}(\hat{\theta})$ is:

$$\text{Bias}(\hat{\text{var}}(\hat{\theta})) = E[\hat{\text{var}}(\hat{\theta})] - \text{var}(\hat{\theta}), \tag{2.3.3}$$

where $\text{var}(\hat{\theta})$ is the true variance of $\hat{\theta}$. Applying this to the Jackknife-after-Bootstrap variance estimator $\hat{\text{var}}_{\text{JAB}}(\hat{\theta}^B(x))$, the bias due to finite B is:

$$\text{Bias}\left(\hat{\text{var}}_{\text{JAB}}(\hat{\theta}^B(x))\right) = E\left[\hat{\text{var}}_{\text{JAB}}(\hat{\theta}^B(x))\right] - \hat{\text{var}}_{\text{JAB}}(\hat{\theta}^\infty(x)), \tag{2.3.4}$$

where $\hat{\theta}^B(x)$ is the bagged estimator based on B bootstrap samples, and $\hat{\theta}^\infty(x)$ is the ideal bagged estimator as $B \to \infty$.

To derive the bias of $\hat{\text{var}}_{\text{JAB}}(\hat{\theta}^B(x))$, consider:

1. **Variance estimator with finite B:**

$$\hat{\text{var}}_{\text{JAB}}(\hat{\theta}^B(x)) = \frac{n-1}{n}\sum_{i=1}^{n}\left(\hat{\Delta}_i\right)^2, \tag{2.3.5}$$

where $\hat{\Delta}_i = \hat{\theta}^{B(i)}(x) - \hat{\theta}^B(x)$, and $\hat{\theta}^{B(i)}(x)$ is the leave-one-out estimator.

2. **Variance estimator as $B \to \infty$:**

$$\hat{\text{var}}_{\text{JAB}}(\hat{\theta}^\infty(x)) = \frac{n-1}{n}\sum_{i=1}^{n}\left(\Delta_i\right)^2, \tag{2.3.6}$$

where $\Delta_i = \hat{\theta}^{\infty(i)}(x) - \hat{\theta}^\infty(x)$.

The bias is then:

$$\begin{aligned}\text{Bias}\left(\hat{\text{var}}_{\text{JAB}}(\hat{\theta}^B(x))\right) &= E\left[\hat{\text{var}}_{\text{JAB}}(\hat{\theta}^B(x))\right] - \hat{\text{var}}_{\text{JAB}}(\hat{\theta}^\infty(x)) \\ &= \frac{n-1}{n}\sum_{i=1}^{n}\left(E\left[\left(\hat{\Delta}_i\right)^2\right] - \left(\Delta_i\right)^2\right).\end{aligned} \tag{2.3.7}$$

Recognizing that:

$$\text{var}\left(\hat{\Delta}_i\right) = E\left[\left(\hat{\Delta}_i\right)^2\right] - \left(E\left[\hat{\Delta}_i\right]\right)^2 = E\left[\left(\hat{\Delta}_i\right)^2\right] - \left(\Delta_i\right)^2, \tag{2.3.8}$$

assuming $E\left[\hat{\Delta}_i\right] = \Delta_i$. Substituting back:

$$\text{Bias}\left(\hat{\text{var}}_{\text{JAB}}(\hat{\theta}^B(x))\right) = \frac{n-1}{n}\sum_{i=1}^{n}\text{var}\left(\hat{\Delta}_i\right). \tag{2.3.9}$$

Thus, the bias of the variance estimator is the sum of the variances of $\hat{\Delta}_i$. Wager and Efron [WHE14] derived the bias of the Jackknife-after-Bootstrap estimator:

Theorem 2.3.8
The bias of the Jackknife-after-Bootstrap variance estimator (Definition 2.3.7) is given by:[a]

$$\begin{aligned}\operatorname{Bias}\left(\hat{\operatorname{var}}_{\mathrm{JAB}}\left(\hat{\theta}^B(x)\right)\right) &= \frac{n-1}{n}\sum_{i=1}^{n}\operatorname{var}\left(\hat{\Delta}_i\right) \\ &\approx \frac{n}{B}(e-1)\,\hat{\operatorname{var}}\left(T\left(x;N^*\right)\right),\end{aligned}$$

where:

- $\hat{\Delta}_i = \hat{\theta}^{B(i)}(x) - \hat{\theta}^B(x)$ is the difference between the leave-one-out estimate and the overall bagged estimate.
- $T(x;N^*)$ is the prediction of the base learner T at input x, trained using the bootstrap sample represented by the count vector N^*.
- $\hat{\operatorname{var}}\left(T\left(x;N^*\right)\right) = \frac{1}{B-1}\sum_{b=1}^{B}\left(T\left(x;N^{*b}\right) - \hat{\theta}^B(x)\right)^2$ is the empirical variance of the base learner's predictions across all bootstrap samples.

This approximation holds under the condition that equal probability weights $W = \left(\frac{1}{n}, \frac{1}{n}, \ldots, \frac{1}{n}\right)^\top$ are used during bootstrap sampling.

[a] [WHE14, Page 1646]

The approximation shows that the bias is proportional to the ratio $\frac{n}{B}$, the factor $e-1$, and the variance of the base learner predictions for the sample x across all bootstrap samples. This correction helps mitigate the upward bias and improves the accuracy of the variance estimate. The bias-corrected version of the Jackknife-after-Bootstrap variance estimator is given as follows:

Definition 2.3.9: Bias-Corrected Jackknife-after-Bootstrap Variance Estimate for Bagged Learners
The bias-corrected Jackknife-after-Bootstrap variance estimator for the bagged learner $\hat{\theta}^B(x)$ is:[a]

$$\hat{\text{var}}_{\text{JAB-U}}\left(\hat{\theta}^B(x)\right) = \frac{n-1}{n}\sum_{i=1}^{n}\left(\hat{\Delta}_i\right)^2 - \frac{n}{B}(e-1)\,\hat{\text{var}}\left(T\left(x;N^*\right)\right), \tag{2.3.10}$$

where:

- $\hat{\Delta}_i = \hat{\theta}^{B(i)}(x) - \hat{\theta}^B(x)$ is the difference between the leave-one-out estimate and the overall bagged estimate.
- $\hat{\theta}^{B(i)}(x)$ is the averaged prediction over all bootstrap samples where the i-th observation X_i does not appear:

  $$\hat{\theta}^{B(i)}(x) = \frac{1}{B_i}\sum_{\substack{b=1\\ N_i^{*b}=0}}^{B} T\left(x;N^{*b}\right),$$

  with $B_i = \left|\left\{b : N_i^{*b} = 0\right\}\right|$ being the number of bootstrap samples where X_i is absent.
- $T\left(x;N^{*b}\right)$ is the prediction of the base learner T at input x, trained using the bootstrap sample represented by the count vector N^{*b}.
- $\hat{\text{var}}\left(T\left(x;N^*\right)\right)$ is the empirical variance of the base learner's predictions across all bootstrap samples:

  $$\hat{\text{var}}\left(T\left(x;N^*\right)\right) = \frac{1}{B-1}\sum_{b=1}^{B}\left(T\left(x;N^{*b}\right) - \hat{\theta}^B(x)\right)^2.$$

This bias correction is valid under the assumption that **equal probability weights** are used during bootstrap sampling, meaning each observation X_i has an equal probability $w_i = \frac{1}{n}$ of being selected.

[a] [WHE14, p. 1629]

In many practical applications, the simple Jackknife-after-Bootstrap estimator (Definition 2.3.7) may require $B = \Theta(n^{1.5})$ bootstrap replicates to reduce Monte Carlo noise to acceptable levels. However, with the bias-corrected version, the number of required bootstrap replicates can often be reduced to $B = \Theta(n)$, making the estimation process more computationally feasible without compromising accuracy.[15]

However, as highlighted in Definition 2.1.3, the IPC-Weighted Bagging Procedure necessitates weighted resampling. This weighting introduces additional complexities in variance estimation, as traditional methods that assume equal weights may no longer provide accurate or unbiased

[15] [WHE14, p. 1638]

estimates. To address these challenges, the next chapter delves into variance estimation methods tailored for Bagged Learners under weighted resampling.

2.4 Nonparametric Variance Estimates for Bagged Learners — Under Weighted Resampling

2.4.1 Nonparametric Bootstrap for BL

To assess the prediction uncertainty of a Bagged Learner (BL), the bootstrap method outlined in Section 2.2.2 offers a robust approach. However, when dealing with scenarios that necessitate weighted resampling—as introduced in Definition 2.1.3—a straightforward bootstrap procedure may fall short in providing accurate variance estimates. In such cases, it becomes imperative to adapt the bootstrap method to accommodate the differential weights assigned to observations. This adaptation is achieved through a two-level bootstrap process, which effectively captures the complexities introduced by weighting.

The two-level bootstrap involves an initial resampling phase that approximates the underlying data distribution without considering weights, followed by a secondary resampling phase where weights are applied to the resampled observations. This layered approach ensures that the inherent variability due to weights is adequately represented, thereby enhancing the reliability of the variance estimates for the BL's predictions. Figure 2.9 illustrates the two-level bootstrap process, highlighting how weights are integrated into the resampling strategy.

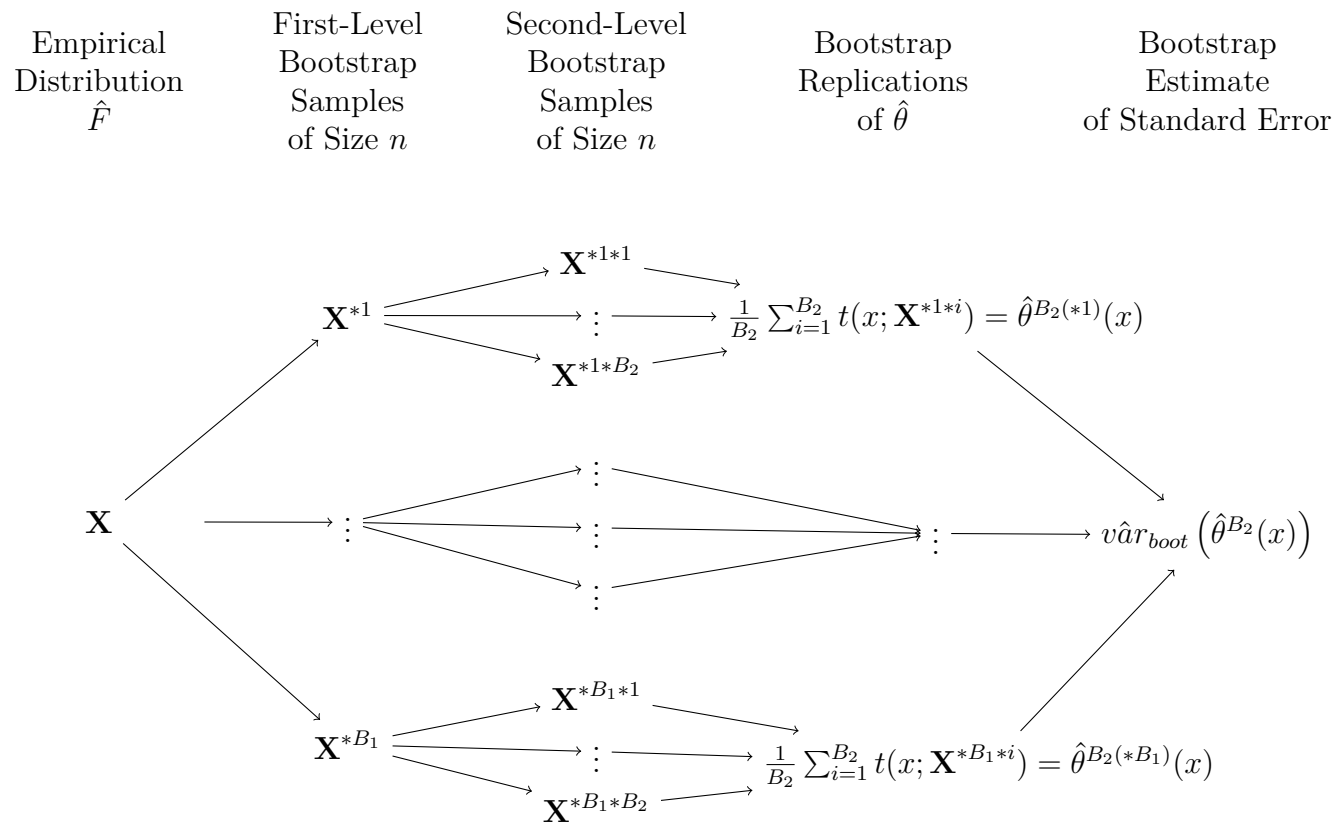


Figure 2.9: Two-Level Bootstrap Process for Bagged Learners

1. **First-Level Bootstrap**: We generate B_1 bootstrap samples under unweighted resampling from the original training dataset, denoted as $\{\mathbf{X}^{*1}, ..., \mathbf{X}^{*B_1}\}$. For each bootstrap sample, a separate bagged learner is trained, resulting in a set of predictions for the input x, denoted as $\{\hat{\theta}^{B_2(*1)}(x), ..., \hat{\theta}^{B_2(*B_!)}(x)\}$.

2. **Second-Level Bootstrap**: For each first-level bootstrap sample, we generate B_2 bootstrap samples under IPC-weights as described in Definiton 2.1.3 . We need this second-level resampling to build our separate bagged learners.

The empirical variance of the predictions from the different bagged learners is then used to estimate the prediction uncertainty:

Definition 2.4.1: Bootstrap Variance Estimate for Bagged Learners
The bootstrap variance estimate of a bagged learner $\hat{\theta}^{B_2}(x)$ at input x is defined as:

$$\hat{\text{var}}_{\text{boot}}\left(\hat{\theta}^{B_2}(x)\right) = \frac{1}{B_1 - 1}\sum_{i=1}^{B_1}\left(\hat{\theta}^{B_2(*i)}(x) - \bar{\theta}^{B_2}(x)\right)^2,$$

where:

- $\hat{\theta}^{B_2(*i)}(x) = \frac{1}{B_2}\sum_{j=1}^{B_2} t(x; \mathbf{X}^{*i*j})$ is the prediction of the i-th bagged learner, trained on the i-th first-level bootstrap sample $\mathbf{X}^{*i}$.
- $\bar{\theta}^{B_2}(x) = \frac{1}{B_1}\sum_{k=1}^{B_1}\hat{\theta}^{B_2(*k)}(x)$ is the mean of the bagged predictions over all first-level bootstrap samples.
- $t(x; \mathbf{X}^{*i*j})$ is the prediction of the base learner trained on the second-level bootstrap sample $\mathbf{X}^{*i*j}$.

This variance estimate $\hat{\text{var}}_{\text{boot}}\left(\hat{\theta}^{B_2}(x)\right)$ provides an assessment of the prediction uncertainty for the bagged learner $\hat{\theta}^{B_2}(x)$ trained on the dataset $\mathbf{X}$.

While the two-level bootstrap method provides a robust framework for variance estimation in Bagged Learners under weighted resampling, it is not without its challenges. The primary limitation lies in its significant computational demands, as generating and training $B_1 \times B_2$ base learners can be resource-intensive and time-consuming. To address these constraints and enhance the efficiency of variance estimation, the following chapter introduces a novel nonparametric variance estimation technique specifically tailored for Bagged Learners. This new method aims to reduce computational overhead while maintaining accuracy and reliability in predicting Bagged Learners uncertainty under weighted resampling.

2.4.2 Infinitesimal Jackknife for BL

The infinitesimal jackknife (IJK) method, as introduced in Section 2.2.4, can be adapted to estimate the variance of a Bagged Learner (BL) by utilizing influence functions to measure the sensitivity of predictions to small perturbations in the data. While existing methods, like Jackknife-after-Bootstrap (Definition 2.3.7), provide unbiased variance estimates under unweighted resampling, we have developed a novel modification of the infinitesimal jackknife specifically tailored to handle scenarios involving weighted resampling within BL. The name of our method is infinitesimal-jackknife-after-weighted-Bootstap-unbiased (IJK-AWB-U).

This extension addresses the complexities introduced by differential probability weights during resampling, ensuring accurate and reliable variance estimates for BL predictions. In the following sections, we derive the influence function for our method, examine the biased variance estimate, introduce a bias correction, and validate the approach through simulation studies.

Derivation of the Influence Function

In our setup, the resampling process involves weighting of observations, as defined by the normalized IPC-weights $W = (w_1, \ldots, w_n)^T$ (Definition 2.1.2). Unlike the traditional resampling vector M_0, which assigns weights $\frac{1}{n}$ to all observations, W incorporates differential weights that reflect the varying probabilities of each observation being included in the bootstrap samples. By substituting M with W, we align the influence function with the actual resampling strategy employed by the Bagged Learner under weighting. This substitution is essential for accurately capturing the influence of each observation on the model's predictions, thereby enabling the computation of unbiased and reliable variance estimates despite the presence of weights.

Following the ideas used by Wager for unweighted resampling in paper [WHE14], we can express the influence function U_i (Definition 2.2.16) under weighted resampling as:

$$\begin{aligned} U(X_i) &= \lim_{\varepsilon \to 0} \frac{H\left((1-\varepsilon)W + e_i\varepsilon\right) - H(W)}{\varepsilon} \\ &= \lim_{\varepsilon \to 0} \frac{H\left(W_{\text{adj}(i)}\right) - H(W)}{\varepsilon} \end{aligned} \tag{2.4.1}$$

Here, $H(W)$ represents the prediction of the BL (Definition 2.3.5) with the IPC Weighted Bagging Procedure (cf. Definition 2.1.3, the p_i in this Definition are the w_i in this chapter), resulting in the base learner predictions $T(x; N^{*b})$. The term $H\left(W_{\text{adj}(i)}\right)$ denotes the prediction of the BL using the adjusted probability weights $((1-\varepsilon)W + e_i\varepsilon)$. While this adjustment would typically require a new resampling process to obtain updated base learner predictions, it can be calculated using the already obtained base learner predictions $T(x; N^{*b})$ from the original resampling process, as we will see later. In Section 2.3.1 we discussed, that the prediction of the BL can be seen as an approximation of the ideal BL (Definition 2.3.4), that means:

$$\begin{aligned} E\left(T(x; N^*) | N^* \sim \text{Mult}(n, W)\right) &\approx H(W) \\ &= \sum_{b=1}^{B} T(x; N^{*b}) \cdot P(N^* = N^{*b} \mid N^* \sim \text{Mult}(n, W)) \end{aligned} \tag{2.4.2}$$

With this relation we can express the influence function $U(X_i)$ from Equation 2.4.1 in expectation of the multinomial distribution:

Definition 2.4.2: Expected Influence Function

$$U(X_i) = \lim_{\varepsilon \to 0} \frac{E\left(T(x; N^*) \mid N^* \sim \text{Mult}\left(n, W_{\text{adj}(i)}\right)\right) - E\left(T(x; N^*) \mid N^* \sim \text{Mult}(n, W)\right)}{\varepsilon}$$

As previously mentioned, we can calculate the expected value $E\left(T(x; N^*) \mid N^* \sim \text{Mult}\left(n, W_{\text{adj}(i)}\right)\right)$ using the existing base learner predictions $T(x; N^{*b})$ from the original resampling process. To achieve this, we introduce a weighting function $wb()$:

Definition 2.4.3: Weighting Function

$$wb\left(W_{\text{adj}(i)}\right) = \frac{P_{W_{\text{adj}(i)}}(N^{*b})}{P_W(N^{*b})}$$

where:

- $P_W(N^{*b}) = P\left(N^* = N^{*b} \mid N^* \sim \text{Mult}(n, W)\right)$
- $P_{W_{\text{adj}(i)}}(N^{*b}) = P\left(N^* = N^{*b} \mid N^* \sim \text{Mult}(n, W_{\text{adj}(i)})\right)$
- $W_{\text{adj}(i)}$
  $= (1-\varepsilon)W + e_i\varepsilon$
  $= ((1-\varepsilon)w_1, \ldots, (1-\varepsilon)w_{i-1}, (1-\varepsilon)w_i + \varepsilon, \ldots, (1-\varepsilon)w_n)^T$

that facilitates the computation of the expected value as shown in Equation 2.4.3.

$$E\left(T(x; N^*) \mid N^* \sim \text{Mult}\left(n, W_{\text{adj}(i)}\right)\right) = E\left(T(x; N^*) \cdot wb(W_{\text{adj}(i)}) \mid N^* \sim \text{Mult}\left(n, W\right)\right) \tag{2.4.3}$$

The Equation 2.4.3 illustrates that the expected prediction of the base learner under the adjusted weight vector $W_{\text{adj}(i)}$ is obtained by weighting the original predictions $T(x; N^{*b})$ with the function $wb()$ under the initial resampling distribution W. Essentially, it allows us to compute the expected value under the new weights by appropriately adjusting the contributions of existing bootstrap samples. The weighting function $wb()$ is defined as the ratio of the probability of observing a specific count vector N^{*b} under the adjusted weights $W_{\text{adj}(i)}$ to its probability under the original weights W. This ratio enables the reweighting of existing bootstrap samples to reflect the new sampling probabilities without necessitating additional resampling. The Equation 2.4.3 holds because:

$$\begin{aligned}
&E\left(T(x; N^*) \cdot wb(W_{\text{adj}(i)}) \mid N^* \sim \text{Mult}\left(n, W\right)\right) \\
&\approx \sum_{b=1}^{B} T(x; N^{*b}) \cdot wb(W_{\text{adj}(i)}) \cdot P_W(N^{*b}) \\
&= \sum_{b=1}^{B} T(x; N^{*b}) \cdot \frac{P_{W_{\text{adj}(i)}}(N^{*b})}{P_W(N^{*b})} \cdot P_W(N^{*b}) \\
&= \sum_{b=1}^{B} T(x; N^{*b}) \cdot P_{W_{\text{adj}(i)}}(N^{*b}) \\
&\approx E\left(T(x; N^*) \mid N^* \sim \text{Mult}\left(n, W_{\text{adj}(i)}\right)\right)
\end{aligned} \tag{2.4.4}$$

The approximation in Equation 2.4.4 arises due to the use of a finite number of bootstrap samples B. With Equation 2.4.3, 2.4.2 and the weighting function $wb()$ (Definition 2.4.3) we can rewrite the expected Influence Function (Definition 2.4.2) :

$$
\begin{aligned}
U(X_i) &= \lim_{\varepsilon \to 0} \frac{E\left(T(x; N^*) \mid N^* \sim \text{Mult}\left(n, W_{\text{adj}(i)}\right)\right) - E\left(T(x; N^*) \mid N^* \sim \text{Mult}(n, W)\right)}{\varepsilon} \\
&= \lim_{\varepsilon \to 0} \frac{E\left(T(x; N^*) \cdot wb(W_{\text{adj}(i)}) \mid N^* \sim \text{Mult}\left(n, W\right)\right) - E\left(T(x; N^*) \mid N^* \sim \text{Mult}(n, W)\right)}{\varepsilon} \\
&\approx \lim_{\varepsilon \to 0} \frac{\sum_{b=1}^{B} T(x; N^{*b}) \cdot wb(W_{\text{adj}(i)}) \cdot P_W(N^{*b}) - \sum_{b=1}^{B} T(x; N^{*b}) \cdot P_W(N^{*b})}{\varepsilon} \\
&= \lim_{\varepsilon \to 0} \frac{\sum_{b=1}^{B} T(x; N^{*b}) \cdot P_W(N^{*b}) \cdot \left(wb(W_{\text{adj}(i)}) - 1\right)}{\varepsilon}
\end{aligned}
\tag{2.4.5}
$$

For further improvement of the Equation 2.4.5, we need now to simplify the weighting function $wb()$ (Definition 2.4.3). Under the multinomial distribution, the probabilities of the count vectors can be rewritten as:

$$
\begin{aligned}
wb\left(W_{\text{adj}(i)}\right) &= \frac{P\left(N^* = N^{*b} \mid N^* \sim \text{Mult}(n, W_{\text{adj}(i)})\right)}{P\left(N^* = N^{*b} \mid N^* \sim \text{Mult}(n, W)\right)} \\
&= \frac{\prod_{k=1, k \neq i}^{n} ((1-\varepsilon) w_k)^{N_k^{*b}} \cdot ((1-\varepsilon) w_i + \varepsilon)^{N_i^{*b}}}{\prod_{k=1}^{n} (w_k)^{N_k^{*b}}} \\
&= \frac{\prod_{k=1}^{n} (1-\varepsilon)^{N_k^{*b}}}{(1-\varepsilon)^{N_i^{*b}}} \cdot \left(\frac{(1-\varepsilon) w_i + \varepsilon}{w_i}\right)^{N_i^{*b}} \\
&= (1-\varepsilon)^{n - N_i^{*b}} \cdot \left(1 + \varepsilon \left(\frac{1 - w_i}{w_i}\right)\right)^{N_i^{*b}}.
\end{aligned}
\tag{2.4.6}
$$

Applying the binomial theorem for natural exponents, we can further simplify this expression:

$$
wb\left(W_{\text{adj}(i)}\right) = \left(\sum_{k=0}^{n - N_i^{*b}} \binom{n - N_i^{*b}}{k} (-\varepsilon)^k\right) \cdot \left(\sum_{k=0}^{N_i^{*b}} \binom{N_i^{*b}}{k} \left(\varepsilon \left(\frac{1 - w_i}{w_i}\right)\right)^k\right)
$$

All terms that have a power of ε equal to 2 or greater can be neglected, since $\varepsilon \to 0$. Thus, the expression simplifies to:

$$
\begin{aligned}
wb\left(W_{\text{adj}(i)}\right) &\approx \left(1 + (n - N_i^{*b})(-\varepsilon)\right) \cdot \left(1 + N_i^{*b} \left(\frac{1 - w_i}{w_i}\right) \varepsilon\right) \\
&\approx 1 + \varepsilon \left(\frac{N_i^{*b}}{w_i} - n\right).
\end{aligned}
\tag{2.4.7}
$$

Now we can further improve the expected Influence Function from Equation 2.4.5 with the simplified weighting function fom Equation 2.4.7:

$$
\begin{aligned}
U(X_i) &= \lim_{\varepsilon \to 0} \frac{\sum_{b=1}^{B} T(x; N^{*b}) \cdot P_W(N^{*b}) \cdot \left(wb(W_{\text{adj}(i)}) - 1 \right)}{\varepsilon} \\
&= \sum_{b=1}^{B} T(x; N^{*b}) \cdot P_W(N^{*b}) \cdot \left(\frac{N_i^{*b}}{w_i} - n \right) \\
&= \sum_{b=1}^{B} \left(T(x; N^{*b}) - E\left(T(x; N^*) + E\left(T(x; N^*)\right) \cdot P_W(N^{*b}) \cdot \frac{1}{w_i} \cdot \left(N_i^{*b} - w_i n \right) \right. \right. \\
&= \frac{1}{w_i} \sum_{b=1}^{B} \left(T(x; N^{*b}) - E\left(T(x; N^*) + E\left(T(x; N^*)\right) \cdot P_W(N^{*b}) \cdot \left(N_i^{*b} - E(N_i^*) \right) \right. \right. \\
&= \frac{1}{w_i} \sum_{b=1}^{B} \left(T(x; N^{*b}) - E\left(T(x; N^*)\right) \cdot \left(N_i^{*b} - E\left(N_i^*\right) \right) \right. \cdot P_W(N^{*b}) \\
&\qquad + \frac{E\left(T(x; N^*)\right.}{w_i} \sum_{b=1}^{B} \left(N_i^{*b} - E\left(N_i^*\right) \right) \cdot P_W(N^{*b})
\end{aligned}
\tag{2.4.8}
$$

The expected values, $E(\cdot)$ in he above equation refers to the expectation under the multinomial distribution of the count vector N^*. As B approaches infinity, the sum $\sum_{b=1}^{B} \left(N_i^{*b} - E\left(N_i^*\right) \right) \cdot P_W(N^{*b}) = E(N_i^*) - E(N_i^*) = 0$. Therefore, the Equation 2.4.8 simplifies to:

$$
U(X_i) = \frac{\text{cov}\left(N_i^*, T\left(x; N^*\right) | N^* \sim Mult(n, W)\right)}{w_i}. \tag{2.4.9}
$$

According to Definition 2.2.18 the Infinitesimal Jackknife variance estimate is:

$$
\hat{\text{var}}_{\text{IJK}}\left(\hat{\theta}\right) = \left(\sum_{i=1}^{n} w_i^2 \right) \sum_{i=1}^{n} w_i U(X_i)^2 \tag{2.4.10}
$$

Here, we only sum the influence functions that have a non-zero weight. With this in mind, we can now define our Infinitesimal Jackknife variance estimate for the ideal BL's prediction under weighted resampling with Equation 2.4.9 and 2.4.10 as:

Definition 2.4.4: Infinitesimal Jackknife Variance Estimate for Ideal Bagged Learner's Prediction (IJK-AWB)

$$\hat{\text{var}}_{\text{IJK-AWB}}\left(\hat{\theta}^{\infty}(x)\right) = \left(\sum_{i=1}^{n} w_i^2\right) \sum_{\substack{i=1 \\ w_i \neq 0}}^{n} \frac{\text{cov}_i^2}{w_i}.$$

where:

- $\text{cov}_i = \text{Cov}\left(N_i^*, T\left(x; N^*\right) \mid N^* \sim \text{Mult}(n, W)\right)$
  cov_i represents the covariance between the entry N_i^* of the count vector N^* and the base learner's prediction $T(x; N^*)$ under the multinomial distribution with weights W.
- w_i are the normalized IPC weights used for resampling in the Bagged Learner. These weights correspond to the resampling procedure upon which the Bagged Learner was trained. Specifically, w_i can be an entry from any resampling vector (Definition 2.2.7)
- $\hat{\theta}^{\infty}(x)$ is the prediction of the Bagged Learner with an infinite number of bootstrap samples, representing the ideal scenario for variance estimation.

This result shows that our IJK-AWB variance estimate relies on the covariance between the resampling counts N_i^* and the base learner's predictions $T(x; N^*)$, scaled by the inverse of the resampling weights w_i. By aggregating these scaled covariances, the IJK-AWB effectively captures the variability introduced by weighted resampling.

2.4.3 Bias-corrected Infinitesimal Jackknife for BL

When estimating variance using resampling methods, a finite number of bootstrap samples B can introduce bias into the variance estimate. This bias arises because the estimator based on a finite B differs from the ideal estimator as $B \to \infty$. Therefore, we do not know the true covariances cov_i , which are required for the calculation of the IJK-AWB estimator, and instead estimate them using $\hat{\text{cov}}_i$. To ensure the accuracy of the variance estimates, it is essential to apply bias corrections. The bias of our IJK-AWB variance estimate (Defintion 2.4.4) under weighted resampling with a finite B can be expressed as:

$$
\begin{aligned}
\text{Bias}\left(\hat{\text{var}}_{\text{IJK-AWB}}\left(\hat{\theta}^B(x)\right)\right) &= E\left(\hat{\text{var}}_{\text{IJK-AWB}}\left(\hat{\theta}^B(x)\right)\right) - \hat{\text{var}}_{\text{IJK-AWB}}\left(\hat{\theta}^\infty(x)\right) \\
&= E\left(\left(\sum_{i=1}^n w_i^2\right)\sum_{i=1,w_i\neq 0}^n \frac{\hat{\text{cov}}_i^2}{w_i}\right) - \left(\sum_{i=1}^n w_i^2\right)\sum_{i=1,w_i\neq 0}^n \frac{\text{cov}_i^2}{w_i} \\
&= \left(\sum_{i=1}^n w_i^2\right)\sum_{i=1,w_i\neq 0}^n \frac{E\left(\hat{\text{cov}}_i^2\right) - \text{cov}^2}{w_i} \\
&= \left(\sum_{i=1}^n w_i^2\right)\sum_{i=1,w_i\neq 0}^n \frac{\text{var}\left(\hat{\text{cov}}_i\right)}{w_i} \\
&= \left(\sum_{i=1}^n w_i^2\right)\sum_{i=1,w_i\neq 0}^n \frac{\text{var}\left(\frac{\sum_{b=1}^B\left(N_i^{*b}-E\left(N_i^*\right)\right)\left(T(x;N^{*b})-E(T(x;N^*))\right)}{B}\right)}{w_i}
\end{aligned}
\tag{2.4.11}
$$

Since $T(x; N^{*b})$ depends on all n observations, in practice, N_i^{*b} and $T(x; N^{*b})$ can be considered independent when calculating $\text{var}\left(\hat{\text{cov}}_i\right)$, particularly when n is large.

$$
\begin{aligned}
\text{Bias}\left(\hat{\text{var}}_{\text{IJK-AWB}}\left(\hat{\theta}^B(x)\right)\right) &\approx \left(\sum_{i=1}^n w_i^2\right)\sum_{i=1,w_i\neq 0}^n\sum_{b=1}^B \frac{\text{var}\left(N_i^*\right)\text{var}\left(T(x;N^*)\right)}{w_i B^2} \\
&= \left(\sum_{i=1}^n w_i^2\right)\sum_{i=1,w_i\neq 0}^n\sum_{b=1}^B \frac{w_i n(1-w_i)\text{var}\left(T(x;N^*)\right)}{w_i B^2} \\
&= \left(\sum_{i=1}^n w_i^2\right)\text{var}\left(T(x;N^*)\right)\sum_{i=1,w_i\neq 0}^n\sum_{b=1}^B \frac{n(1-w_i)}{B^2} \\
&= \left(\sum_{i=1}^n w_i^2\right)\text{var}\left(T(x;N^*)\right)\frac{n}{B}\sum_{i=1,w_i\neq 0}^n (1-w_i)
\end{aligned}
\tag{2.4.12}
$$

The approximation of the bias shows that the bias of our IJK-AWB estimate depends on the variance of the base learner's predictions and the sample size. Specifically, the bias is inversely proportional to the number of bootstrap samples B, and thus increasing B can reduce bias.

Having discussed the bias correction approach, we can now present our unbiased form of IJK-AWB variance estimate for the BL's prediction with a finite B under weighted resampling as:

Definition 2.4.5: Unbiased Infinitesimal Jackknife Variance Estimate for Bagged Learner's Prediction (IJK-AWB-U)

$$\hat{\text{var}}_{\text{IJK-AWB}-U}\left(\hat{\theta}^B(x)\right) = \left(\sum_{i=1}^{n} w_i^2\right)\left(\sum_{\substack{i=1\\ w_i\neq 0}}^{n} \frac{\hat{\text{cov}}_i^2}{w_i} - \hat{\text{var}}\left(T(x;N^*)\right)\frac{n}{B}\sum_{\substack{i=1\\ w_i\neq 0}}^{n}(1-w_i)\right)$$

where:

- $\hat{\text{cov}}_i = \frac{1}{B}\sum_{b=1}^{B}\left(N_i^{*b} - w_i n\right)\left(T(x;N^{*b}) - \frac{\sum_{i=1}^{B} T\left(x;N^{*i}\right)}{B}\right)$
  $\hat{\text{cov}}_i$ represents the estimated covariance between the observed resampling counts N_i^{*b} and the base learner's prediction $T(x;N^{*b})$ under the multinomial distribution with weights W.
- w_i are the normalized IPC weights used for resampling in the Bagged Learner. These weights correspond to the resampling procedure upon which the Bagged Learner was trained. Specifically, w_i can be derived from any resampling vector (Definition 2.2.7).
- $\hat{\theta}^B$ is the prediction of the Bagged Learner, trained with B bootstrap samples.
- $\hat{\text{var}}\left(T\left(x;N^*\right)\right) = \frac{1}{B-1}\sum_{b=1}^{B}\left(T\left(x;N^{*b}\right) - \frac{\sum_{i=1}^{B} T\left(x;N^{*i}\right)}{B}\right)^2$
  $\hat{\text{var}}\left(T\left(x;N^*\right)\right)$ is the empirical variance of the base learner predictions across all bootstrap samples.

When unweighted resampling is applied during the resampling process, such that $w_i = \frac{1}{n}$ for $i = 1,\ldots,n$, the variance estimate in Definition 2.4.5 simplifies to:

$$\hat{\text{var}}_{IJK-AB-U}\left(\hat{\theta}^B(x)\right) = \sum_{i=1}^{n}\hat{\text{cov}}_i^2 - \frac{n-1}{B}\hat{\text{var}}\left(T(x;N^*)\right) \tag{2.4.13}$$

This result for unweighted resampling has also been derived by Wager and co-authors in their paper [WHE14].[16]

2.4.4 Simulations

To validate the proposed method IJK-AWB-U (Definition 2.4.5) we conducted two simulation studies.

1. Simulation: Weighted Bagged Mean Estimator

In the first simulation, we employ a weighted Bagged Learner that computes the mean of the input dataset $\mathbf{X}$. Specifically, we compare the theoretically derived variance of the mean estimator with the variance estimated using our method IJK-AWB-U and the biased method IJK-AWB. The simulation framework is outlined as follows:

[16] [WHE14, p. 1629]

Simulation Setup

We generate synthetic datasets with the following characteristics:

- **Sample Size (n)**: We fix the sample size at $n = 1{,}000$.
- **Variable (X)**: For simplicity, we generate observations from a standard normal distribution, i.e., $X_i \sim \mathcal{N}(0, 1)$, independently for each observation.
- **Synthetic Dataset**: $\mathbf{X} = (X_1, \ldots, X_n)$.
- **Weight Distribution (W)**: The first 500 observations are assigned a weight of $w_i = \frac{2}{1000}$ for $i = 1, \ldots, 500$, and the last 500 observations are assigned a weight of $w_i = 0$ for $i = 501, \ldots, 1000$. These weights reflect the probability of each observation being selected in the bootstrap samples.
- **Number of Bootstrap Samples (B)**: We vary the number of bootstrap samples among $B = 500$, 1000, 2000, 4000, and 10000 to assess the stability of our estimator with respect to the number of bootstrap samples.

The weighted Bagged Learner computes the mean of the dataset $\mathbf{X} = (X_1, \ldots, X_n)$ using bootstrap samples drawn according to the weight distribution W. Formally, for each bootstrap sample $b = 1, \ldots, B$, we draw n observations with replacement according to the probabilities W, and compute the mean for the base learner $T(\cdot)$:

$$T(N^{*b}) = \frac{1}{n} \sum_{i=1}^{n} N_i^{*b} X_i,$$

where N_i^{*b} is the count of the i-th observation in the b-th bootstrap sample. The mean computation of the weighted Bagged Learner is then:

$$\hat{\theta}^B(\mathbf{X}) = \frac{1}{B} \sum_{b=1}^{B} T(N^{*b}). \tag{2.4.14}$$

In expectation, the mean computation of the weighted Bagged Learner approximates:

$$E\left(T(N^*) \mid N^* \sim \text{Mult}(n, W)\right) \approx \hat{\theta}^B(\mathbf{X}). \tag{2.4.15}$$

Theoretical Variance Calculation

Under the multinomial resampling scheme with weights W, the theoretical variance of the expected mean computation of the weighted Bagged Learner can be derived as follows. Given that $X_i \sim \mathcal{N}(0, 1)$ and are independent of the resampling process, the true variance is:

$$\begin{aligned}
\operatorname{var}\left(E\left(T(N^*) \mid N^* \sim Mult(n, W)\right)\right) &= \operatorname{var}\left(E\left(\frac{1}{n}\sum_{i=1}^{n} N_i^* \cdot X_i \mid N^* \sim Mult(n, W)\right)\right) \\
&= \operatorname{var}\left(\sum_{i=1}^{n} \frac{E\left(N_i^* \mid N_i^* \sim B(n, w_i)\right)}{n} \cdot X_i\right) \\
&= \operatorname{var}\left(\sum_{i=1}^{n} \frac{n \cdot w_i}{n} \cdot X_i\right) \\
&= \sum_{i=1}^{n} w_i^2 \cdot \operatorname{var}\left(X_i\right) \\
&= \operatorname{var}\left(X\right) \cdot \sum_{i=1}^{n} w_i^2 \cdot
\end{aligned} \tag{2.4.16}$$

and with our simulation setup we get

$$\begin{aligned}
\operatorname{var}\left(E\left(T(N^*)|N^* \sim Mult(n, W)\right)\right) &= 1 \cdot \left(\sum_{i=1}^{500} \left(\frac{2}{1000}\right)^2 + \sum_{i=501}^{1000} 0\right) \\
&= 0.002
\end{aligned} \tag{2.4.17}$$

This theoretical variance serves as the ground truth for evaluating the accuracy of our IJK-AWB-U variance estimator.

Simulation Run

We conduct 2,000 simulation runs for each value of the number of bootstrap samples B. For each simulation run, the following steps are performed:

1. Generate synthetic dataset $\mathbf{X} = (X_1, \ldots, X_{1000})$ as desribed in the simulation setup.
2. Assign weights $W = (w_1, \ldots, w_{1000})^T$, with $w_i = \frac{2}{1000}$ for $i = 1, \ldots, 500$ and $w_i = 0$ for $i = 501, \ldots, 1000$.
3. Draw B bootstrap samples from the multinomial distribution $\text{Mult}(1000, W)$ and with these samples calculate the count vectors N^{*b}
4. For each count vector N^{*b} , compute the base learner prediction $T(N^{*b}) = \frac{1}{1000}\sum_{i=1}^{1000} N_i^{*b} X_i$.
5. Compute variance estimates IJK-AWB (Definition 2.4.4) and IJK-AWB-U (Definition 2.4.5) using N^{*b} and $T(N^{*b})$

Simulation Results

The simulation results presented in Table 2.1 demonstrate the performance of the IJK-AWB-U and IJK-AWB variance estimators relative to the theoretical variance.

Table 2.1: Simulation Results: Theoretical Variance vs. IJK-AWB-U Estimated Variance vs. IJK-AWB

B	**Theoretical Variance**	**IJK-AWB-U estimate**	**IJK-AWB estimate**
500	0.002000	0.0020 ± 0.0003	0.0040 ± 0.0005
1,000	0.002000	0.0020 ± 0.0002	0.0030 ± 0.0003
2,000	0.002000	0.0020 ± 0.0002	0.0025 ± 0.0002
4,000	0.002000	0.0020 ± 0.0002	0.0022 ± 0.0002
10,000	0.002000	0.0020 ± 0.0001	0.0021 ± 0.0001

IJK-AWB-U Estimator: The IJK-AWB-U estimates are remarkably consistent with the theoretical variance of 0.002 across all values of B. The estimated variances are all approximately 0.002, with standard deviations decreasing as B increases. For example, at $B = 500$, the estimate is 0.002 ± 0.0003, and at $B = 10000$, it is 0.002 ± 0.0001. This consistency indicates that the IJK-AWB-U estimator is unbiased and reliable, even with a smaller number of bootstrap samples. The decreasing standard deviations with larger B suggest increased precision due to the averaging effect of more bootstrap samples. These results align with the theoretical expectation that the IJK-AWB-U estimator corrects for bias inherent in finite bootstrap samples.

IJK-AWB Estimator: The IJK-AWB estimates exhibit a positive bias, particularly noticeable at smaller values of B. At $B = 500$, the estimate is 0.004 ± 0.0005, which is double the theoretical variance. As B increases, the bias diminishes, and the estimates converge toward the theoretical variance. At $B = 1000$, the estimate decreases to 0.0030 ± 0.0003, and by $B = 10000$, it reaches 0.0021 ± 0.0001, closely matching the theoretical value. The decreasing standard deviations with larger B reflect improved precision. This behavior suggests that while the IJK-AWB estimator is consistent in the limit as $B \to \infty$, it is biased in finite samples due to the lack of bias correction for the finite number of bootstrap samples.

Conclusion The results underscore the importance of using the IJK-AWB-U estimator when an unbiased variance estimate is crucial. The IJK-AWB-U estimator provides accurate variance estimates without requiring a large B. This simulation validates the theoretical properties of the IJK-AWB-U estimator, confirming its unbiasedness and efficiency in variance estimation for weighted Bagged Learners.

2. Simulation: Weighted Bagged Tree Model

In this simulation, we investigate the performance of our IJK-AWB-U variance estimator (Definition 2.4.5) in the context of a weighted Bagged Tree model, similar to the analysis presented in Figure 2 of [WHE14]. We consider a scenario where half of the observations are assigned a weight of $\frac{2}{n}$, and the other half are assigned a weight of zero for the weight distribution W.

Simulation Setup

We conduct the simulation with the following specifications:

- **Sample Size (n)**: We fix the sample size at $n = 1{,}000$.

- **Input Variable (x_i)**: Each observation x_i is drawn independently from a uniform distribution on the interval $[0, 1]$, i.e., $x_i \sim \text{Uniform}(0, 1)$ for $i = 1, \ldots, n$.
- **Response Variable (y_i)**: The response variable is generated according to a step function with added noise:
$$y_i = f(x_i) + \varepsilon_i,$$
where $f(x)$ is defined as in Figure 2.10, and $\varepsilon_i \sim \mathcal{N}\left(0, \left(\frac{1}{2}\right)^2\right)$ represents Gaussian noise with standard deviation $\frac{1}{2}$.

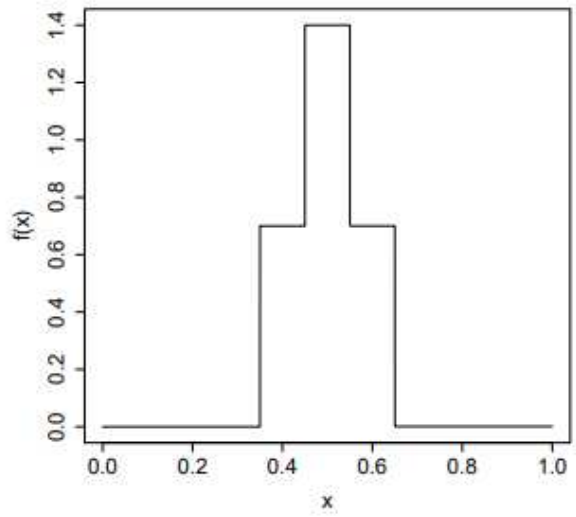


Figure 2.10: Step Function $f(x)$. Figure adapted from [WHE14, p. 1648]

- **Weight Distribution (W)**: Randomly select $\frac{n}{2}$ observations to be assigned a weight of $w_i = \frac{2}{n}$; the remaining observations are assigned a weight of $w_i = 0$.
- **Model**: We use a weighted Bagged Regression Tree. The base learner is a regression tree with 5 terminal nodes (leaves), grown using the bootstrap samples drawn according to the weight distribution W.
- **Number of Bootstrap Samples (B)**: We set the number of bootstrap samples to $B = 1{,}000$.

Variance Estimation Methods

For each simulation run, we estimate the variance of the Bagged Tree predictions using the IJK-AWB-U estimator (Definition 2.4.5) and compare it with the empirical variance of the Bagged Tree predictions over all 2,000 simulations.

Simulation Procedure

For each of the $n_{\text{sim}} = 2{,}000$ simulation runs, we perform the following steps:

1. **Data Generation**:
   a) Generate the input variables $x_i \sim \text{Uniform}(0, 1)$ for $i = 1, \ldots, 1{,}000$.

b) Compute the response variables $y_i = f(x_i) + \varepsilon_i$, where $\varepsilon_i \sim \mathcal{N}\left(0, \left(\frac{1}{2}\right)^2\right)$.

c) create training dataset $\mathbf{X} = ((x_1, y_1), ..., (x_{1000}, y_{1000}))$

2. **Assign Weights**: Create the weight vector $W = (w_1, \ldots, w_{1000})^\top$ by randomly selecting $\frac{n}{2} = 500$ indices without replacement to assign $w_i = \frac{2}{n}$; the remaining indices are assigned $w_i = 0$.

3. **Bootstrap Sampling**: For each bootstrap sample $b = 1, \ldots, 1{,}000$, draw $n = 1{,}000$ observations with replacement according to the multinomial distribution $\text{Mult}(n, W)$ to obtain the bootstrap samples and count vectors N^{*b}.

4. **Model Training**: Fit a regression tree T_b with 5 leaves to each bootstrap sample

5. **Prediction**: For 1,000 fixed test points x_0 evenly spaced between 0 and 1, compute the Bagged Tree predictions:

$$\hat{\theta}^B(x_0) = \frac{1}{B}\sum_{b=1}^{B} T_b(x_0).$$

6. **Variance Estimation**: Estimate the variance of $\hat{\theta}^B(x_0)$ using the IJK-AWB-U estimator for each test point.

At the end, we compute the empirical variance of the Bagged Tree predictions over all 2,000 simulations for each test point x_0.

Simulation Results

The results are presented in Figure 2.11, which shows the **mean variance estimates** obtained from the IJK-AWB-U estimator over all simulations, compared to the empirical variance of the Bagged Tree predictions over all simulations. The empirical variance can be considered as the **true variance** of the Bagged Tree predictions. The purple shaded area represents the mean variance estimates plus or minus one standard deviation of the IJK-AWB-U variance estimates across the simulations.

The results in Figure 2.11 demonstrate that, on average, the IJK-AWB-U estimator accurately captures the variability of the Bagged Tree predictions across different values of the input variable x_0. The mean variance estimates from the IJK-AWB-U estimator closely match the empirical variance computed over all simulations, indicating that the estimator effectively accounts for the uncertainty in the ensemble predictions.

The empirical varaince of the Bagged Tree predictions, shown as the solid blue line, can be considered the true variance in this context. The purple shaded area, representing ± 1 standard deviation of the IJK-AWB-U variance estimates, illustrates the variability of our variance estimator across the simulations. This area encompasses the true variance throughout of the range of x_0, demonstrating the reliability of the IJK-AWB-U estimator.

Notably, the variance is higher near the step in the function $f(x)$ at $x = 0.35, 0.45, 0.55, 0.65$ reflecting the increased difficulty of predicting in regions where the step function (cf. Figure

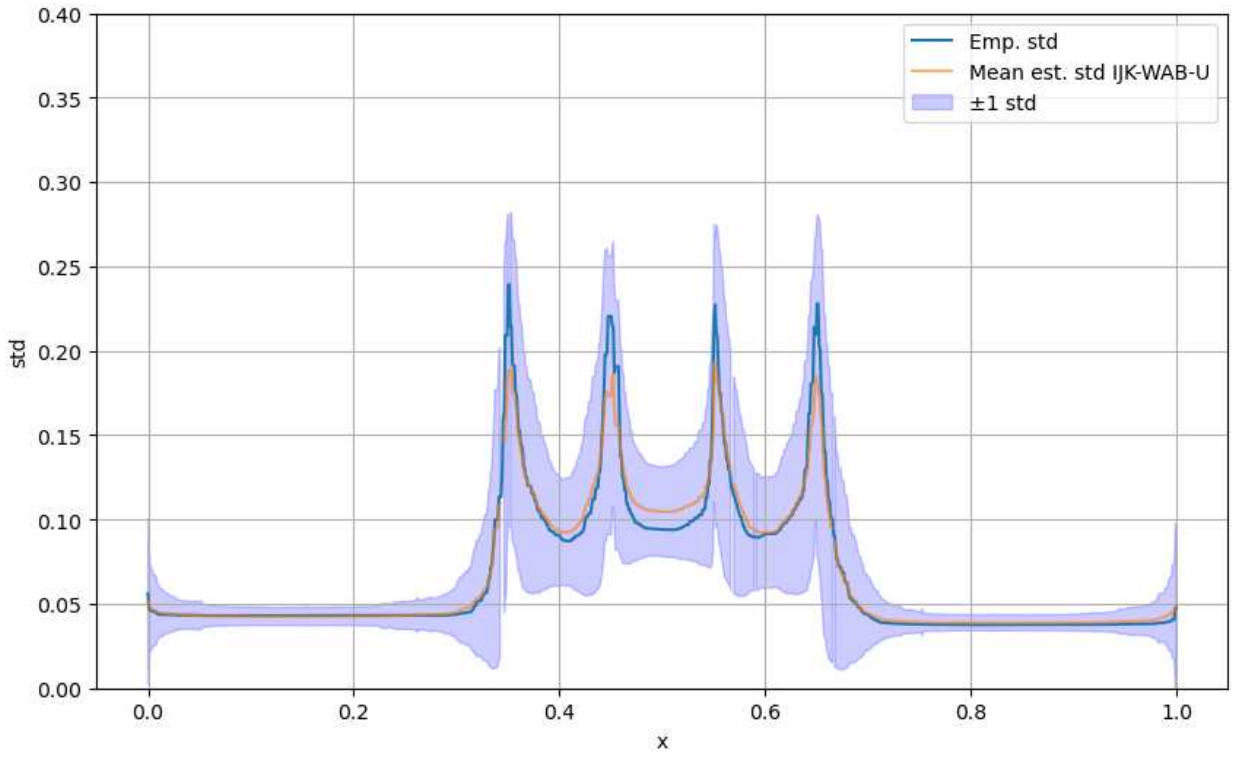


Figure 2.11: Variance estimates of the Bagged Tree predictions using the IJK-AWB-U estimator and the empirical variance over all simulations.

2.10) changes abruptly. Our variance estimator accurately identifies the location and magnitude of these spikes in variance, successfully capturing the behavior of the true variance across the entire range of x_0.

These results underscore the effectiveness of the IJK-AWB-U estimator in providing reliable variance estimates, even in challenging scenarios with abrupt changes in the underlying function. The close alignment between the mean IJK-AWB-U variance estimates and the empirical variance confirms that the estimator can be confidently used for quantifying uncertainty in ensemble predictions.

We now proceed to a comprehensive simulation study. This study is structured following the ADEMP framework—Aim, Data-generating mechanisms, Estimands, Methods, and Performance measures—to systematically evaluate the variance estimation methods in the context of IPC-weighted resampling.

3 Simulation Study Documentation Following the ADEMP Framework

3.1 Simulation Design

3.1.1 Aim (A)

The primary aim of this simulation study is to investigate the performance of the Infinitesimal-Jackknife-after-weighted-Bootstrap-unbiased (IJK-AWB-U) method for survival probability predictions in the context of right-censored survival data. In particular, this study will focus on the accuracy and robustness of IJK-AWB-U when applied to a bagging ensemble of decision trees (*DecisionTreeBaggingClassifier*). As part of this investigation, we will also compare the results of IJK-AWB-U with other variance estimation methods to provide a broader understanding of its relative performance. Following variance estimation methods are considered:

Variance Estimation Methods

1. *Infinitesimal-Jackknife-after-weighted-Bootstrap (IJK-AWB)* :
    - Accounts for IPC weighted bagging procedure.
    - The variance estimate is calculated as described in Definition 2.4.4.
2. *Infinitesimal-Jackknife-after-weighted-Bootstrap-unbiased (IJK-AWB-U)*:
    - Accounts for IPC weighted bagging procedure.
    - Accounts for bias introduced through finite bootstrap samples, generated with IPC weights.
    - The variance estimate is calculated as described in Definition 2.4.5.
3. *Jackknife-after-Bootstrap-unbiased (JK-AB-U)*:
    - Does not account for IPC weighted bagging procedure
    - Assumes equal probability weights, used during bootstrap sampling.
    - Accounts for bias introduced through finite bootstrap samples, generated with equal probability weights.
    - The variance estimate is calculated described in Definiton 2.3.9.
4. *Bootstrap*:
    - The method accounts for IPC weighted bagging procedure.

- Variance is estimated by resampling the data B_1 times (with replacement with equal probability weights), retraining the bagged-model (in the model training IPC weights are used), and computing the variance of the predictions.
- Number of Bootstrap Samples (B_1): 200
- The variance estimate is calculated as described in Definition 2.4.1.

Additionally, the study investigates how these methods perform under varying sample sizes, different levels of censoring in the dataset, and different shape parameters of the Weibull distribution used for data generation (specifically, shape parameters $k = 1$ and $k = 1.5$). For the models, various numbers of bootstrap samples B are considered during training of the DecisionTreeBaggingClassifier. The ultimate goal is to determine the accuracy and reliability of IJK-AWB-U method when applied to survival predictions in the presence of censoring.

3.1.2 Data-Generating Mechanisms (D)

Synthetic survival data are generated based on a Weibull distribution, incorporating several covariates that affect the survival times. The data generation process simulates event times and censoring times independently to reflect right-censored survival data commonly encountered in practice.

Parameters

The parameter values selected for this simulation study were chosen arbitrarily, but they were designed to span a broad spectrum of possible scenarios. This allows for a comprehensive evaluation of the performance of the models under different conditions that could be encountered in practice.

- Number of Simulation Runs (n_{sim}): The simulation study consists of $n_{\text{sim}} = 1000$ replicates for each combination of parameters.
- Random Seed: A base random seed is set (e.g., seed = 42), and different seeds are used for each simulation replicate to ensure variability (seed + i for the i-th simulation).
- Sample Sizes (n): Simulations are conducted for different sample sizes:

  $n = \{714, 1428, 2857\}$
  (The sample sizes were chosen to ensure that the desired training sizes can be obtained later after the train-test split.)
- Weibull Shape Parameters (k):
  - $k = 1$: Corresponds to an exponential distribution (constant hazard).
  - $k = 1.5$: Represents a Weibull distribution with increasing hazard over time.
- Censoring Proportion (cens-p): The censoring proportion represents the fraction of samples that are censored before time τ:

$$\text{cens-p} = \frac{\text{number of censored samples before } \tau}{\text{total number of samples}}$$

  This proportion is set to one of the following values: 0.1, 0.3, 0.5, or 0.7, indicating 10%, 30%, 50%, and 70% of the samples being censored, respectively.

- Event Proportion (event-p): The event proportion denotes the proportion of samples in which the event of interest is observed before time τ. It is derived by multiplying the remaining proportion of uncensored samples by a factor p, where p takes values in $\{0.1, 0.2, 0.3, 0.4\}$. Mathematically, it is expressed as:

  $$\text{event-p} = (1 - \text{cens-p}) \times p$$

  For each censoring proportion, the event proportion is set to 10%, 20%, 30%, and 40% of the uncensored samples. This ensures that the number of events scales appropriately with the level of censoring.

  Example: If the censoring proportion is 0.3 (30%), the remaining uncensored proportion is 0.7. The event proportion would then be:

  $$\text{event-p} = 0.7 \times p$$

  where p can be 0.1, 0.2, 0.3, or 0.4, resulting in event proportions of 7%, 14%, 21%, and 28%, respectively.

  Note: The censoring and event proportions represent the average values across 1000 simulated datasets. Each simulation run may exhibit slightly different proportions.

- Number of bootstrap samples (B): Defines the number of decision trees used in the DecisionTreeBaggingClassifier. We used 4 different values $B = 500, 1000, 2000, 4000$ in our simulation study.

Therefore, there are $2 \cdot 3 \cdot 4^3 = 384$ parameter combinations. However, Bootstrap variance estimates are only calculated for the combinations with $n = 2857$, $B = 1000$ and $k = 1.5$ due to computational runtime constraints.

Covariates

For each individual X_i in the dataset we generate 5 covariates $(x_{i1}, x_{i2}, x_{i3}, x_{i4}, x_{i5})$ with following characteristics (using the `NumPy` library):

$$x_1 \sim \mathcal{N}(50, 10^2) \qquad x_2 \sim \text{Bernoulli}(p = 0.3) \qquad x_3 \sim \mathcal{N}(25, 5^2)$$

$$x_4 \sim \text{Bernoulli}(p = 0.2) \qquad x_5 \sim \text{LogNormal}(\mu = 5, \sigma = 1)$$

(x_5 Values are clipped to the range [30, 8000] to avoid extreme values)

Survival Times

Survival times are generated using a Weibull distribution with the following characteristics (using the `NumPy` library):

- Baseline Scale Parameter (λ_0):

  Adjusted to achieve different levels of censoring in the data. Specific values are provided in the Appendix under Section 6.0.2.

- Scale Parameter (λ_i):

  The individual-specific scale parameter λ_i is calculated using a baseline scale parameter (λ_0) and a linear predictor (LP_i) based on the covariates:

$$\lambda_i = \lambda_0 \times \exp(\text{LP}_i)$$

  The linear predictor LP_i is defined as:

$$\text{LP}_i = -0.405 \cdot x_{i1} - 0.4 \cdot x_{i2} - 0.05 \cdot x_{i3} - 0.01 \cdot (x_{i4} - 25)^2 - 0.2 \cdot \log(x_{i5})$$

- Shape Parameter (k):

  As specified above ($k = 1$ or $k = 1.5$).

- Event Times (t_i^*):

  Generated as:

$$t_i^* \sim \text{Weibull}(k, \lambda_i)$$

  The Weibull probability density function is:

$$f(t) = \frac{k}{\lambda_i} \left(\frac{t}{\lambda_i}\right)^{k-1} \exp\left(-\left(\frac{t}{\lambda_i}\right)^k\right)$$

  The true survival probability at a specific time point τ and for a specific observation X_i can be derived as:

$$S(\tau|X_i) = \exp\left(-\left(\frac{\tau}{\lambda_i}\right)^k\right)$$

  and being interpreted as the probability that an individual with covariate values X_i survives beyond time τ. In other words, it quantifies the likelihood that the event of interest (e.g., death) has **not** occurred by time τ.

Censoring Times (c_i)

Censoring times are generated from an exponential distribution to simulate right-censoring (using the `NumPy` library):

$$c_i \sim \text{Exponential}(\text{rate} = \lambda_c)$$

The rate parameter λ_c is adjusted to achieve desired censoring proportions in the data. Specific values are provided in the Appendix under Section 6.0.2.

Observed Times and Events

For each individual, the observed time and event indicator are determined:

- Observed Time (t_i):

$$t_i = \min(t_i^*, c_i)$$

- Event Indicator (δ_i):

$$\delta_i = \begin{cases} 1, & \text{if } t_i^* \leq c_i \\ 0, & \text{otherwise} \end{cases}$$

Train-Test Split

The dataset is split into training and testing sets using a stratified split that maintains the proportion of events and censored observations (using the `train_test_split` function from the `sklearn.model_selection` library).

- Training Set (n_{train}): 70% of the data
  - $n_{train} = \{499, 999, 1999\}$
- Test Set (n_{test}): 30% of the data
  - $n_{test} = \{215, 429, 858\}$

Inverse Probability of Censoring Weights (IPCW)

- A Kaplan-Meier estimator is used to approximate the censoring distribution in the training set (using the `lifelines` library).
- Cut-off Time (τ): The time horizon for classification is set to $\tau = 37$.
- IPC weights are calculated for each individual in the training set using the Kaplan-Meier estimator of the censoring distribution (using Definition 2.1.2).
- IPC weights are normalized, so $\sum_{i=1}^{n_{train}} w_i = 1$.
- Survival Status (= Target Column) (y_i): For each individual in the training and test set, the target value is determined:

$$y_i = \begin{cases} 1, & \text{if } t_i > \tau \\ 0, & \text{if } t_i \leq \tau, \delta_i = 1 \\ \text{NA}, & \text{otherwise} \end{cases}$$

3.1.3 Estimands (E)

The estimands of interest in this simulation study refer to the prediction for a specific individual with predetermined covariate values denoted as $X_{\text{pred}} = (50, 0, 25, 0, e^{5+0.5})$. These covariate values represents the expected individual.

1. **Predicted Survival Probability at Time τ**:

   Estimate the survival probability at time τ for X_{pred}

   $$\hat{S}(\tau | X_{\text{pred}})$$

   using different models (`Weibull AFT` and `DecisionTreeBaggingClassifier`).

2. **Estimated Variance of Predicted Survival Probability at Time τ**:

   Utilize various variance estimation methods to estimate the variability of the predicted survival probability for X_{pred} at time τ using the `DecisionTreeBaggingClassifier` model:

3. **Model Performance Metrics on Test Data**:

   IPC-weighted MSE of the survival predictions on the test data.

3.1.4 Methods (M)

The following 2 models are used to generate survival predictions:

1. **Weibull Accelerated Failure Time (W-AFT)** (using the `lifelines` library):
   - A parametric survival model assuming a Weibull distribution.
   - Fitted to the training data using the covariates and observed times/events.
   - Survival probabilities are predicted for the test data and X_{pred} at time τ.
2. **DecisionTreeBaggingClassifier (DTBC)**:
   - A custom bagging ensemble of decision trees (not using the `RandomForest` or `BaggingClassifier` implementation from scikit-learn).
     - The custom DTBC class is designed to train multiple decision trees (using `DecisionTreeClassifier` implementation from scikit-learn) and aggregate their predictions to output the probability for class 1 (e.g., survival).
     - Unlike standard implementations, this class provides a method to return bootstrapped samples, used for training the descion trees.
   - Each decision tree is trained on a bootstrapped sample of the training data using the covariates and the target column.
   - Bootstrapping is weighted using the normalized IPC weights.
   - The DTBC estimates the probability of the Survival Status being 1 at Time τ, representing the likelihood of surviving beyond time τ.
   - Survival probabilities are predicted for the test data and X_{pred} at time τ.

- The hyperparameters of the DTBC (max_depth= 4, min_samples_split= 5, max_features= $log2$) were selected through cross-validation on a sampled dataset, optimizing for the IPC-weighted MSE. Once determined, these hyperparameters were kept constant across all simulations.

And with the following metric we will evaluate the performance of the models:

- **IPC-weighted MSE**:

$$\text{MSE} = \frac{1}{n} \sum_{i=1}^{n} w_i \left(y_i - \hat{y}_i\right)^2$$

where w_i are the normalized IPC weights, y_i are the observed survival statuses, and $\hat{y}_i$ are the predicted survival probabilities.

Simulation runs are parallelized using a `ProcessPoolExecutor` to efficiently utilize computational resources.

3.1.5 Performance Measures (P)

The performance of the models and variance estimation methods is assessed using the following measures:

1. **Mean Prediction**
   - The mean predicted survival probabilitiy for X_{pred} at time τ across all simulation replicates
   $$\overline{\hat{S}\left(\tau | X_{\text{pred}}\right)} = \frac{1}{1000} \sum_{sim=1}^{1000} \hat{S}(\tau | X_{\text{pred}})_{sim}$$
   is calculated for the models W-AFT and DTBC.
2. **Empirical Standard Deviation of Predictions**
   - The empirical standard deviation s of the predicted survival probabilities for X_{pred} at time τ across all simulation replicates
   $$s\left(\hat{S}\right) = \frac{1}{999} \sum_{sim=1}^{1000} \left(\hat{S}(\tau | X_{\text{pred}})_{sim} - \overline{\hat{S}(\tau | X_{\text{pred}})}\right)^2$$
   is calculated for the models W-AFT and DTBC.
   - Serves as the benchmark for the true variability in predictions.
3. **Estimated Standard Deviations of Predictions**
   - Standard deviations are estimated for the predictions of the DTBC
   - For each variance estimation method (IJK-AWB, IJK-AWB-U, JK-AB-U, Bootstrap), the estimated standard deviation is calculated as the square root of the estimated variance:
   $$\hat{\sigma}_{sim} = \sqrt{\hat{\text{var}}_{sim}}$$

- The mean estimated standard deviation across simulations is computed for each method:

$$\overline{\hat{\sigma}} = \frac{1}{1000} \sum_{sim=1}^{1000} \hat{\sigma}_{sim}$$

4. **Mean Relative Bias (MRB) of Estimated Standard Deviations**

$$\text{MRB } (\%) = \left(\frac{\overline{\hat{\sigma}} - s\left(\hat{S}\right)}{s\left(\hat{S}\right)} \right) \times 100\%$$

5. **Coefficient of Variation (CV) of Estimators**

- The Coefficient of Variation (CV) of the estimated standard deviations across all simulations is calculated as:

$$\text{CV} = \left(\frac{\frac{1}{999} \sum_{sim=1}^{1000} \left(\hat{\sigma}_{sim} - \overline{\hat{\sigma}}\right)^2}{\overline{\hat{\sigma}}} \right) \times 100\%$$

- Enables the assessment of the relative stability and reliability of the different variance estimation methods with different levels of censoring in the data.

6. **Model Performance Metric: Mean MSE**

- IPC-weighted MSE on test data, averaged across the 1000 simulation runs:

$$\overline{MSE} = \frac{1}{1000} \sum_{sim=1}^{1000} MSE_{sim}$$

3.2 Simulation Results

3.2.1 Model's Performance

This section presents the findings from the simulation study, focusing on the performance of the Weibull Accelerated Failure Time (W-AFT) model and the DecisionTreeBaggingClassifier (DTBC) model under various simulation settings. We examine how the models behave under different simulation settings, and the impact on the accuracy and variability of survival probability predictions.

Table 3.1 summarizes the model performance metrics for the W-AFT model and the DTBC model under different simulation settings. The results are presented for both Weibull shape parameters (k), a censoring proportion (cens-p) of 0.5, an event proportion (event-p) of 0.15, and for the DTBC, a bootstrap sample size of $B = 1000$ was used.

Table 3.1: Model Results with cens-p= 0.5, event-p= 0.15 and for DTBC here was used $B = 1000$

k	n_{train}	$s\left(\hat{S}\right)$ for **W-AFT**	$s\left(\hat{S}\right)$ for **DTBC**	$\overline{MSE}$ for **W-AFT**	$\overline{MSE}$ for for **DTBC**
1	499	0.0204	0.0485	0.1581	0.1618
	999	0.0150	0.0380	0.1580	0.1588
	1999	0.0100	0.0317	0.1565	0.1562
1.5	499	0.0192	0.0508	0.1550	0.1578
	999	0.0140	0.0426	0.1548	0.1544
	1999	0.0094	0.0362	0.1537	0.1522

Impact of Sample Size (n_{train}) From Table 3.1, we observe that increasing the training sample size n_{train} leads to a reduction in the empirical standard deviation $s\left(\hat{S}\right)$ of the predicted survival probabilities for both models. Specifically:

- For the W-AFT model with $k = 1$, $s\left(\hat{S}\right)$ decreases from 0.0204 at $n_{\text{train}} = 499$ to 0.0100 at $n_{\text{train}} = 1999$.
- Similarly, for the DTBC with $k = 1$, $s\left(\hat{S}\right)$ decreases from 0.0485 to 0.0317 as n_{train} increases.

This trend indicates that both models produce more consistent predictions with larger training datasets, as expected due to the increased information available for model training. Furthermore, this trend is observed across all combinations of censoring proportions, event proportions and the number of bootstrap samples B as the training sample size increases. Notably, the number of bootstrap samples B used in the DTBC does not appear to significantly influence the empirical standard deviation of its predictions, as long as B is chosen to be in the order of $\mathcal{O}(n)$, meaning that a number of bootstrap samples proportional to the size of the training data is sufficient to maintain prediction stability. These findings can also be verified in the figures located in the appendix under Sections 6.0.5 and 6.0.6.

Comparison Between W-AFT and DTBC Models Following observations can be made, when comparing the performance of the W-AFT and DTBC models:

- Empirical Standard Deviation of Predictions $s\left(\hat{S}\right)$:
  - The DTBC consistently exhibits a higher $s\left(\hat{S}\right)$ compared to the W-AFT model across all settings. For instance, with $k = 1$ and $n_{\text{train}} = 499$, the DTBC has $s\left(\hat{S}\right) = 0.0485$ versus 0.0204 for the W-AFT.
  - This suggests that the DTBC predictions are more variable across simulation replicates, potentially reflecting greater sensitivity to the training data.
  - These findings can also be observed across all combinations of simulation parameters (cf. figures in Section 6.0.5 and 6.0.6).
- Mean MSE ($\overline{\text{MSE}}$):
  - The $\overline{\text{MSE}}$ values for both models decrease slightly as n_{train} increases, indicating improved prediction accuracy with more data.

- The W-AFT model generally achieves lower $\overline{\text{MSE}}$ values than the DTBC, although the differences are small and diminish with larger sample sizes.
- For example, at $n_{\text{train}} = 1999$ and $k = 1$, the $\overline{\text{MSE}}$ for W-AFT is 0.1565 compared to 0.1562 for DTBC, showing comparable performance.
- These findings can also be observed across all combinations of simulation parameters. Notably, the number of bootstrap samples B used in the DTBC does not influence the $\overline{MSE}$ of DTBC's predictions (cf. figures in Section 6.0.5 and 6.0.6).

Effect of Weibull Shape Parameter (k) The shape parameter k influences the hazard function of the Weibull distribution. Comparing the results between $k = 1$ and $k = 1.5$ we observed:

- For the W-AFT model, both the empirical standard deviation $s\left(\hat{S}\right)$ and the $\overline{\text{MSE}}$ decrease as k increases.The observed decrease in $\overline{\text{MSE}}$ suggests that the specific simulation conditions favor improved performance with higher k.
- In contrast, for the DTBC model, the empirical standard deviation $s\left(\hat{S}\right)$ increases with k, indicating more variability in predictions under increasing hazard rates. However, the mean squared error $\overline{\text{MSE}}$ decreases slightly, similar to the W-AFT model.
- These findings are also observed across different combinations of censoring proportions and event proportions in the figures located in the appendix under Sections 6.0.5 and 6.0.6.

Examining the 24 figures located in the appendix under Sections 6.0.5 and 6.0.6, we observed consistent patterns regarding the performance of the DTBC and W-AFT models under varying simulation parameters. Since these patterns are similar across different training sample sizes (n_{train}) and numbers of bootstrap samples (B), we focus our analysis on Figure 3.1 to illustrate these findings. All 24 figures follow the same structure and layout as Figure 3.1, allowing for easy comparison of results across different settings.

Key Observations In Figure 3.1:

- Impact of Event Proportion on $s\left(\hat{S}\right)$ and $\overline{MSE}$: Panels with smaller event proportions show narrower error bars for both models, indicating smaller empirical standard deviations in the predicted survival probabilities. Additionally, the $\overline{MSE}$ is also lower when event proportions are smaller. Fewer events simplify the survival patterns that the models need to learn, leading to more consistent predictions across simulation replicates and reduced variability.
- Impact of Censoring Proportion on $s\left(\hat{S}\right)$ and $\overline{MSE}$: When comparing subplots with similar event proportions but varying censoring proportions, we observe that higher censoring proportions result in wider error bars for both the W-AFT and DTBC models. This indicates higher empirical standard deviations $s\left(\hat{S}\right)$ in the predicted survival probabilities. Additionally, the $\overline{\text{MSE}}$ increases with higher censoring proportions. Increased censoring in the data reduces the amount of observed event information, which in turn diminishes the models' ability to accurately learn and predict survival patterns. This loss of information introduces greater uncertainty and variability in the

survival probability estimates, leading to less consistent predictions across simulation replicates.

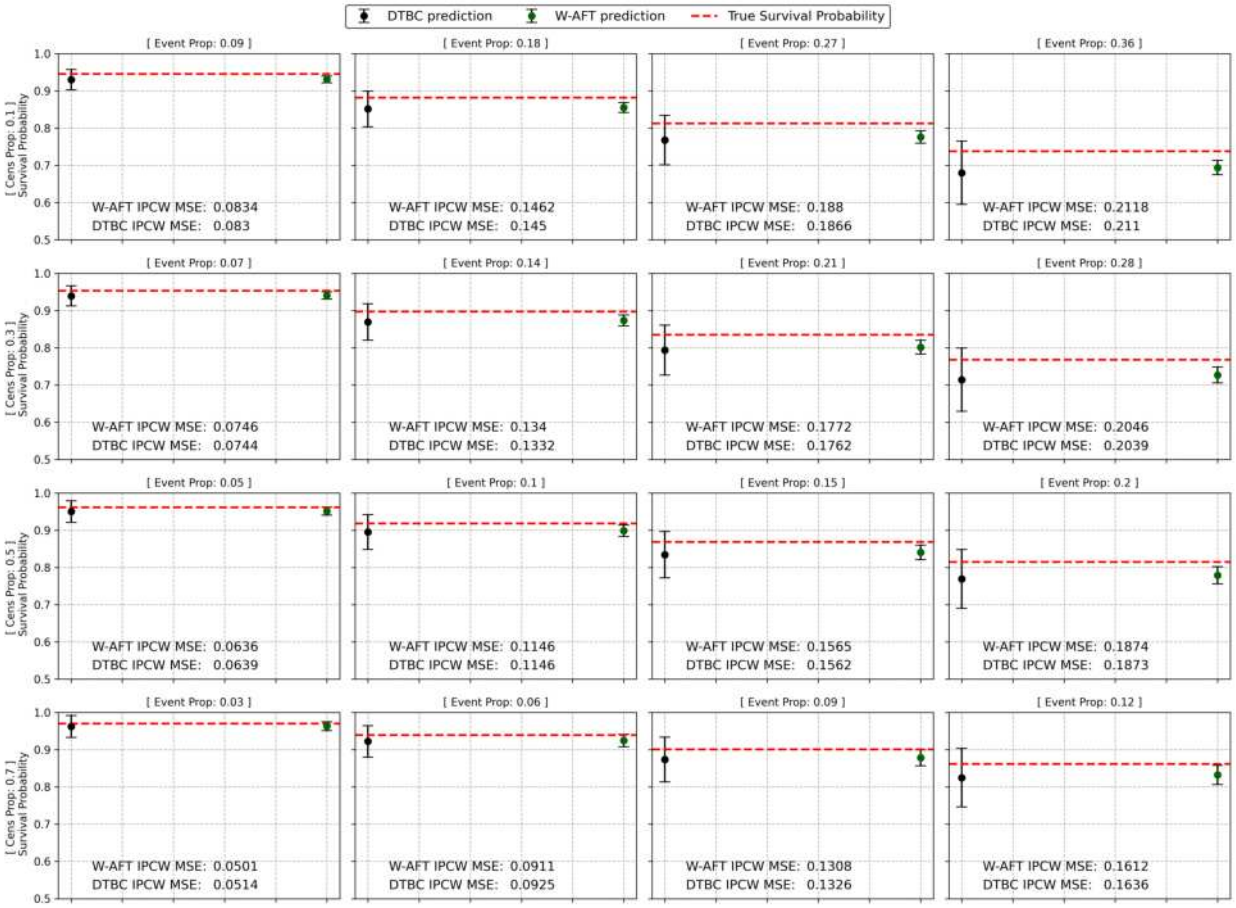


Figure 3.1: Simulation study results for model's performance with $k = 1$, $n_{\text{train}} = 1999$, $B = 1000$. The prediction of the models represents the mean estimated survival probability $\overline{\hat{S}(\tau|X_{pred})}$ and the MSE represents the $\overline{MSE}$ over the 1000 simulation runs. The error bars correspond to $\overline{\hat{S}(\tau|X_{pred})} \pm 1.96 \cdot s\left(\hat{S}\right)$, which provides an approximate 95% confidence interval of the mean prediction. Each subplot corresponds to a different event and censoring proportion, as indicated by the "Event Prop" and "Cens Prop".

- Confidence Interval Coverage of the True Survival Probability: Across all scenarios, the approximate 95% confidence interval of the mean prediction from the DTBC model consistently includes the true survival probability $S(\tau|X_{\text{pred}})$. This suggests that the DTBC model's predictions fluctuate around the true value across simulations. In scenarios with higher event proportions (starting from event-p = 0.14), the W-AFT model's error bars do not include the true survival probability. This indicates that the W-AFT model's predictions are consistently biased away from the true value in these settings. The Weibull assumptions may be violated in the simulation settings, leading to biased predictions that do not align with the true survival probability.

- Systematic Underestimation of the True Survival Probability: A key observation across almost all models is that they tend to systematically underestimate the true survival probability on average. This underestimation is particularly notable in settings with higher event proportions. However, as this phenomenon is not the focus of the present study, it will not be further investigated.

In summary, the analysis of the model performance reveals that both the W-AFT and DTBC models benefit from larger training sample sizes, exhibiting reduced variability and improved prediction accuracy. The DTBC model consistently shows higher variability in its predictions compared to the W-AFT model, possibly due to its non-parametric nature and sensitivity to data complexity. Additionally, the DTBC model's predictions fluctuate around the true survival probability, while the W-AFT model's predictions may exhibit bias in certain scenarios.

These observations highlight the necessity of accurately quantifying the variability associated with model predictions. Therefore, in the next section, we focus on evaluating the performance of the variance estimation methods, we used for the DTBC model. We aim to determine how well these methods capture the true variability of the model's predictions across different simulation settings.

3.2.2 Variance Estimator's Performance

This section presents the findings from the simulation study, focusing on the performance of the Infinitesimal-Jackknife-after-weighted-Bootstrap-unbiased (IJK-AWB-U) variance estimator under various simulation settings. We will evaluate how well the IJK-AWB-U estimator captures the true variability of the DecisionTreeBaggingClassifier's (DTBC) predictions. Following this evaluation, we will compare the IJK-AWB-U estimator's performance with other variance estimators, such as the IJK-AWB, JK-AB-U, and Bootstrap estimator.

Performance of IJK-AWB-U

The Mean Relative Bias (MRB(%)) is calculated as the percentage difference between the mean of the estimated standard deviations ($\overline{\widehat{\sigma}}$) and the empirical standard deviation of the predictions ($s\left(\hat{S}\right)$). A positive MRB(%) indicates that the variance estimator is overestimating the true variability, while a negative MRB(%) indicates underestimation. Table 3.2 presents the MRB(%) of the IJK-AWB-U variance estimator under different simulation settings. The results are shown for both Weibull shape parameters ($k = 1$ and $k = 1.5$), and the different training sample sizes (n_{train}), and varying numbers of bootstrap samples (B). The censoring proportion (cens-p) is fixed at 0.3, and the event proportion (event-p) is set at 0.21.

Key observations from Table 3.2:

- Effect of Training Sample Size (n_{train}):
  - The MRB decreases as the training sample size increases. At $n_{\text{train}} = 499$, the MRB ranges from 7% to 15%, while at $n_{\text{train}} = 1999$, it decreases to values between -3% and 4%.

Table 3.2: MRB(%) for Variance Estimator IJK-AWB-U with Simulation Parameters: cens-p= 0.3 and event-p= 0.21

k	n_{train}	**MRB (%) for IJK-AWB-U**			
		$B = 500$	$B = 1000$	$B = 2000$	$B = 4000$
1	499	13	15	15	15
	999	8	10	11	11
	1999	0	3	4	4
1.5	499	7	9	10	10
	999	0	3	4	4
	1999	-3	0	1	1

  - This trend indicates improved accuracy of the IJK-AWB-U estimator with larger training datasets.

- Effect of Number of Bootstrap Samples (B):
  - The MRB tends to increase slightly with larger B, especially noticeable at smaller n_{train}. For example, with $k = 1.5$ and $n_{\text{train}} = 999$, the MRB increases from 0% at $B = 500$ to 3% at $B = 1000$ and remains at 4% after that for larger B.
  - This suggests diminishing returns in variance estimation accuracy when increasing B beyond a certain point. A moderate number of bootstrap samples (e.g., $B = n_{train} \div 2$) appears sufficient. Therefore, computational resources may be better utilized elsewhere once a reasonable number of bootstrap samples is reached.

- Effect of Weibull Shape Parameter (k):
  - The MRB values are generally lower for $k = 1.5$ compared to $k = 1$, indicating better performance of the variance estimator when the hazard rate is increasing. At $n_{\text{train}} = 999$, the MRB for $k = 1.5$ is 0% at $B = 500$, whereas for $k = 1$, it is 8%.

The patterns observed in Table 3.2 are consistent across different combinations of event proportions and censoring proportions. It is important to note that the MRB does not always remain within ±15% under other settings. In some scenarios with different combinations of event proportions and censoring proportions, the IJK-AWB-U estimator may exhibit higher MRB. To recognize the settings where the estimator performs well and identifying its limitations, we now present Figure 3.2.

The Figure 3.2 illustrates the performance of various variance estimators for the DecisionTreeBaggingClassifier (DTBC) under different simulation settings, specifically focusing on the event and censoring proportions. Each subplot corresponds to a unique combination of event and censoring proportions, indicated by the "Event Prop" and "Cens Prop" labels at the top of each panel.

(The 24 figures located in the appendix under Section 6.0.4 and 6.0.3 follow the same structure and layout as Figure 3.2, except that the bootstrap variance method was not applied there due to computational runtime constraints. These figures contain results for the other combinations of n_{train}, k and B.)

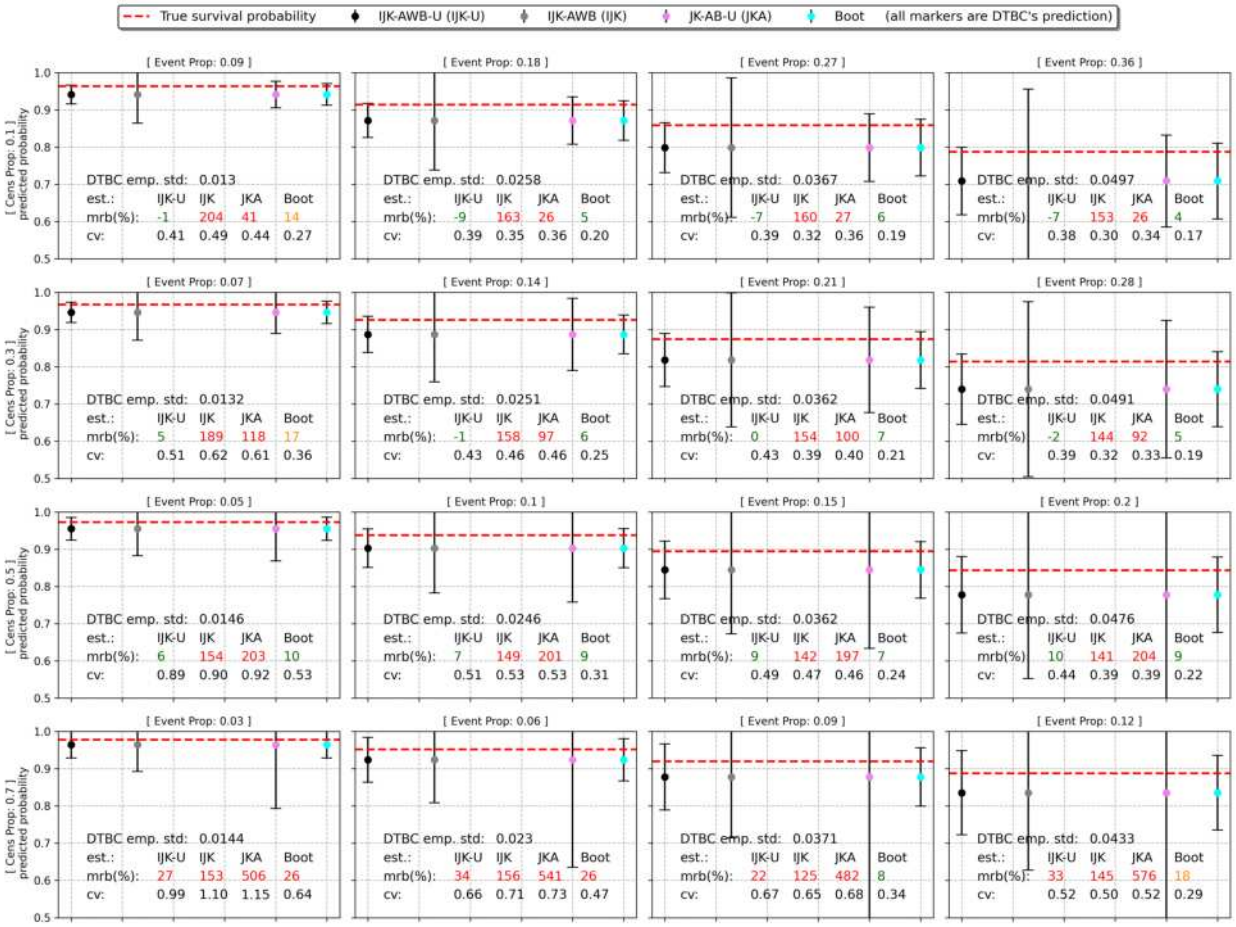


Figure 3.2: Simulation study results for variance estimator's performance with $k = 1$, $n_{\text{train}} = 1999$, $B = 1000$. DTBC's prediction represents the mean estimated survival probability $\overline{\hat{S}(\tau|X_{pred})}$ and the DTBC emp. std stands for $s\left(\overline{\hat{S}}\right)$ over the 1000 simulation runs. The 4 error bars correspond to $\overline{\hat{S}(\tau|X_{pred})} \pm 1.96 \cdot \overline{\hat{\sigma}}$, which provides an approximate 95% confidence interval of the mean prediction. The mean estimated standard deviations ($\overline{\hat{\sigma}}$) for each error bar is based on an estimator from the legend. If $|\text{mrb}(\%)| \leq 10$, it is colored green, if $10 < |\text{mrb}(\%)| \leq 20$, it is colored yellow, otherwise its colored red.

Several patterns regarding the performance of the IJK-AWB-U estimator can be identified from Figure 3.2:

- Effect of Censoring Proportion on MRB(%): The MRB(%) of the IJK-AWB-U estimator is significantly influenced by the censoring proportion. When the censoring proportion is less than 0.7, the MRB ranges between -9% and 10%, indicating acceptable estimation accuracy. However, at a censoring proportion of 0.7, the MRB increases substantially, ranging from 22% to 33%, which suggests an overestimation of the variance. This pattern appears consistent regardless of the event proportion. Observations from the additional figures in the appendix (Sections 6.0.4 and 6.0.3) confirm that for censoring proportions up to 0.3, the absolute MRB is generally less than or equal to 10%. For a censoring proportion of 0.5, the absolute MRB typically ranges between 5% and 25%, and for a censoring proportion of 0.7, it generally lies between 10% and 40%.

These findings imply that the IJK-AWB-U method performs well on average when the censoring proportion is moderate to low.

- Effect of Event and Censoring Proportion on CV: When holding the censoring proportion constant and examining the subplots in each row, we observe that the CV decreases as the event proportion increases. To better illustrate this relationship, we have extracted the relevant values from Figure 3.2 and present them in Table 3.3:

Table 3.3: Impact of Event Proportion on Coefficients of Variation (Simulation Parameters: $k = 1.5$, $n_{\text{train}} = 1999$, $B = 1000$)

Cens-p = 0.5		Cens-p = 0.3		Cens-p = 0.1	
Event-p	**CV**	**Event-p**	**CV**	**Event-p**	**CV**
0.05	0.89	0.07	0.51	0.09	0.41
0.1	0.51	0.14	0.43	0.18	0.39
0.15	0.49	0.21	0.43	0.27	0.39
0.2	0.44	0.28	0.39	0.36	0.38

For example the CV decreases from 0.89 to 0.44, when incresing the event proportion, for a cens propotion of 0.5 . Conversely, when comparing subplots with similar event proportions but different censoring proportions, we notice that the CV increases with higher censoring. For instance, the CV is 0.41 for cens-p = 0.1 and event-p = 0.09, while it increases to 0.51 for cens-p = 0.5 and event-p = 0.1. These patterns we can also observe in the additional figures in the appendix (Sections 6.0.4 and 6.0.3). This indicates that higher event proportions lead to more stable variance estimates (lower CV), whereas higher censoring proportions increase the variability of the variance estimates (higher CV).

In summary, the IJK-AWB-U estimator provides reliable variance estimates under conditions of low to moderate censoring proportions. Its performance, in terms of MRB(%) and CV, is adversely affected by high censoring proportions, where it tends to overestimate the variance and exhibit greater variability. The event proportion has a more pronounced effect on the CV than on the MRB(%), with higher event proportions leading to more stable variance estimates. These observations highlight the importance of considering the censoring and event proportions when applying the IJK-AWB-U estimator in practice.

Comparison of IJK-AWB-U Estimator with Other Variance Estimators

In addition to the IJK-AWB-U estimator, we compared its performance with other variance estimators, namely the IJK-AWB, JK-AB-U, and Bootstrap estimators. Figure 3.2 illustrates the performance of these estimators under various simulation settings.

The key observations from the comparison are as follows:

- **IJK-AWB Estimator:** The Infinitesimal-Jackknife-after-weighted-Bootstrap (IJK-AWB) estimator, without the bias correction applied in the IJK-AWB-U method, exhibits significantly higher MRB(%) compared to the IJK-AWB-U estimator (cf.

Figure 3.2). This indicates that the bias correction in the IJK-AWB-U estimator effectively improves the accuracy of the variance estimates.

Table 3.4: Impact of Bootstrap Sample Size (B) on MRB(%) of IJK-AWB-U, IJK-AWB and JK-AB-U (Simulation Parameters: $k = 1.5$, $n_{\text{train}} = 1999$, cens-p= 0.3, and event-p= 0.21)

	MRB(%) of		
B	**IJK-AWB-U**	**IJK-AWB**	**JK-AB-U**
500	−3	244	166
1,000	0	154	100
2,000	1	93	57
4,000	1	55	31

Specifically, the IJK-AWB estimator tends to overestimate the variance considerably, especially in scenarios with smaller B, as observed in Table 3.4. For instance, at $B = 500$, the MRB for the IJK-AWB estimator is 244%, indicating a substantial overestimation of the variance, whereas the IJK-AWB-U estimator shows an MRB of −3%, closely aligning with the true variability. As B increases, the MRB(%) for the IJK-AWB estimator decreases but remains significantly higher than that of the IJK-AWB-U estimator, suggesting persistent overestimation even with larger bootstrap sample sizes. Observations from the additional figures in the appendix (Sections 6.0.4 and 6.0.3) confirm that the bias correction in the IJK-AWB-U estimator is effective in almost every scenario of our simulation settings.

Given these findings, we conclude that the IJK-AWB estimator lacks reliability due to its tendency to overestimate variance, which can mislead interpretations of model uncertainty. Therefore, the bias-corrected IJK-AWB-U estimator is preferred for providing more accurate and dependable variance estimates of the DTBC model's predictions.

- **JK-AB-U Estimator:** The Jackknife-after-Bootstrap-unbiased (JK-AB-U) estimator exhibits mixed performance. Generally, as shown in Figure 3.2 and supported by Table 3.4, it consistently displays higher MRB(%) values compared to the IJK-AWB-U estimator, indicating substantial overestimation of the variance. This overestimation is particularly evident at lower bootstrap sample sizes (B).

  However, in scenarios with low censoring proportions and larger bootstrap sample sizes, the performance of the JK-AB-U estimator improves and can become comparable to that of the IJK-AWB-U estimator. This is illustrated in Table 3.5, which presents results under different simulation parameters ($k = 1.5$, $n_{\text{train}} = 1999$, `cens-p`= 0.1, and `event-p`= 0.18). In Table 3.5, as B increases, the MRB of the JK-AB-U estimator decreases significantly, reaching 1% at $B = 4000$, which is comparable to the MRB of −8% for the IJK-AWB-U estimator. This improved performance at low censoring levels is likely because the effect of censoring is minimal in these scenarios. It is important to note that the JK-AB-U estimator is misspecified for our simulation settings, as it does not account for censoring. Therefore, its performance decrease in scenarios with higher censoring proportions where the impact of censoring becomes more pronounced.

Table 3.5: Impact of Bootstrap Sample Size (B) on MRB(%) of IJK-AWB-U and JK-AB-U (Simulation Parameters: $k = 1.5$, $n_{\text{train}} = 1999$, `cens-p`= 0.1, and `event-p`= 0.18)

B	**MRB(%) of IJK-AWB-U**	**MRB(%) of JK-AB-U**
500	−14	53
1,000	−9	26
2,000	−8	10
4,000	−8	1

Overall, despite occasional comparable performance under specific conditions (e.g., low censoring proportion and large B), the JK-AB-U estimator generally provides less accurate variance estimates compared to the IJK-AWB-U estimator. The consistent overestimation observed in most scenarios (as seen in Figure 3.2 and the additional figures in the appendix Sections 6.0.4 and 6.0.3) indicates that the JK-AB-U estimator is less reliable for accurate variance estimation in the context of our simulation settings. Consequently the IJK-AWB-U estimator remains the preferred choice for variance estimation of the DTBC model's predictions.

- **Bootstrap Estimator:** Due to computational runtime constraints, the Bootstrap estimator was only evaluated under the simulation setting corresponding to Figure 3.2. Therefore, we have results for the Bootstrap estimator only from this figure. Both Table 3.6 and Table 3.7 were generated from Figure 3.2 to better illustrate these performance comparisons.

Table 3.6: Performance Comparison at Low to Moderate Censoring Proportions for IJK-AWB-U and Bootstrap Estimators ($k = 1.5$, $n_{\text{train}} = 1999$, $B = 1000$)

Cens-p	**Event-p**	**IJK-AWB-U MRB(%)**	**Bootstrap MRB(%)**	**IJK-AWB-U CV**	**Bootstrap CV**
0.1	0.09	−1	14	0.41	0.27
0.1	0.18	−9	5	0.39	0.20
0.1	0.27	−7	6	0.39	0.19
0.1	0.36	−7	4	0.38	0.17
0.3	0.07	5	17	0.51	0.36
0.3	0.14	−1	6	0.43	0.25
0.3	0.21	0	7	0.43	0.21
0.3	0.28	−2	5	0.39	0.19
0.5	0.05	6	10	0.89	0.53
0.5	0.10	7	9	0.51	0.31
0.5	0.15	9	7	0.49	0.24
0.5	0.20	10	9	0.44	0.22

Analyzing the performance of the Bootstrap estimator in Table 3.6, we observe that it provides variance estimates with relatively low MRB(%) and CV across the simulated scenarios. At low to moderate censoring proportions (cens-p ≤ 0.5), the Bootstrap estimator shows good performance, MRB(%) values ranging from 5% to 17%, indicating

acceptable bias in variance estimation. The CV values are lower than those of the IJK-AWB-U estimator, suggesting more stable variance estimates. Additionally, they show decreasing CV values with increasing event proportions, as we previously observed with the IJK-AWB-U estimator.

At higher censoring proportions (`cens-p` = 0.7), both estimators exhibit increased MRB(%) and CV values. Table 3.7 presents the performance metrics in these settings.

Table 3.7: Performance Comparison at High Censoring Proportions for IJK-AWB-U and Bootstrap Estimators ($k = 1.5$, $n_{\text{train}} = 1999$, $B = 1000$)

Cens-p	**Event-p**	**IJK-AWB-U MRB(%)**	**Bootstrap MRB(%)**	**IJK-AWB-U CV**	**Bootstrap CV**
0.7	0.03	27	26	0.99	0.64
0.7	0.06	34	26	0.66	0.47
0.7	0.09	22	8	0.67	0.34
0.7	0.12	33	18	0.52	0.29

In these high censoring scenarios, the Bootstrap estimator consistently achieves lower MRB(%) than the IJK-AWB-U estimator. However, the Bootstrap estimator also faces challenges in accurately capturing the true variance at high censoring levels. The CV values for the Bootstrap estimator are lower, indicating more stable variance estimates. Here too, we observe decreasing CV values with increasing event proportions, similar to the pattern seen with the IJK-AWB-U estimator.

However, it is important to note that the computational cost of the Bootstrap estimator is significantly higher due to the need for resampling and refitting the DTBC model multiple times. This may limit its practicality for larger datasets or more complex models. The IJK-AWB-U estimator, despite having higher CV values, offers a balance between computational efficiency and acceptable variance estimation accuracy. Therefore, in practice, the choice between the Bootstrap and IJK-AWB-U estimators should consider both the statistical performance and the computational resources available.

In summary, our simulation study demonstrated that the IJK-AWB-U estimator provides reliable and accurate variance estimates for the DTBC model's prediction, especially under low to moderate censoring proportions and higher training sample sizes. It effectively corrects bias present in the IJK-AWB estimator and offers a good balance between accuracy and computational efficiency compared to the Bootstrap estimator, which, despite its stability, is computationally intensive. The JK-AB-U estimator was generally less reliable due to its consistent overestimation of variance. Given these findings, the IJK-AWB-U estimator emerges as the preferred method for variance estimation in our context.

In the next chapter, we will apply the variance estimation methods discussed in this chapter to the TxReg dataset to assess their practical utility in estimating the variance of predictions made by a DTBC on real-world data.

4 Application on TxReg Dataset

In the earlier chapters, we delved into various methods for estimating the variance of the Decision Tree Bagging Classifier's (DTBC) prediction, evaluating their effectiveness through extensive simulation studies. Building on these insights, we now shift our focus to applying these variance estimation techniques to a real-world dataset to assess their practical applicability.

This chapter examines the use of the IJK-AWB-U, IJK-AWB, JK-AB-U, and Bootstrap variance estimators on the TxReg dataset, which contains retrospective and anonymized data from the German Transplant Registry. These data have been supplied by the Transplant Registry Agency, represented by the Gesundheitsforen Leipzig GmbH.

The research project within which this thesis was conducted is funded by the Federal Ministry of Education and Research (project 13FH019KX1). As part of this project, the registry data underwent comprehensive feature engineering and data cleaning processes. These efforts resulted in a refined dataset $\mathbf{X}_{clean}$ devoid of missing values and comprising only contextually relevant features. The results presented in this thesis are the responsibility of the author.

4.1 Description of the Dataset

The $\mathbf{X}_{clean}$ dataset utilized in this study comprises 17,016 observations across 21 distinct features, of which 19 are used for prediction and 2 represent the `time` and `event` variables. Table 4.1 provides a detailed overview of the dataset's features, categorized by their respective data types.

Table 4.1: Features of the TxReg Dataset Categorized by Data Type

Data Type	**Features**
`float64` (×10)	time, donor_age_years, donor_height_cm, donor_weight_kg, donor_creatinin_umol_per_l, recipient_age_years, recipient_height_cm, recipient_weight_kg, recipient_dialysis_years, transplant_cold_ischemia_time_min
`object` (×1)	destination
`bool` (×10)	event, donor_sex, recipient_sex, donor_diabetes, donor_hypertension, donor_smoking, donor_hcv, recipient_bloodtransfusion, recipient_hcv, recipient_pra

The dataset encompasses a diverse range of features that capture both donor and recipient characteristics, as well as transplantation-specific details. Additionally, the dataset was filtered to include only observations with `time` > 0. Furthermore, the ''Local'' category in

the `destination` feature was removed due to its low frequency (14 instances), resulting in `destination` having only three categories. Now the dataset comprises 15,786 observations.

In our classification problem, we consider a time horizon of three years. Specifically, the objective is to determine whether a patient has not experienced the event of interest within this three-year period. The distribution of events and censored observations in the dataset after the cutoff time $\tau = 3$ years is as follows:

- Proportion of Censored Observations: 19.63%
  represents the fraction of samples that are censored before time τ
- Proportion of Events: 18.69%
  denotes the proportion of samples in which the event of interest is observed before time τ

These proportions indicate that a moderate portion of the data is censored at a time horizon of three years.

4.2 DTBC Model

To effectively predict patient outcomes within a three-year time horizon, we employed a Decision Tree Bagging Classifier (DTBC).

Hyperparameter Evaluation

The performance of the DTBC model is highly dependent on the selection of optimal hyperparameters. To systematically identify the best combination of hyperparameters, we implemented a cross-validation approach encapsulated within the `evaluate_params` function. This function performs the following steps:

1. **Cross-Validation Setup:** Utilizing K-Fold cross-validation with $K = 10$ folds, the dataset is partitioned into training and validation subsets. This ensures that the model is evaluated on diverse data segments, enhancing the reliability of performance metrics.
2. **Model Training and Validation:** For each fold, the DTBC model is trained on the training subset with a specific set of hyperparameters. The Kaplan-Meier estimator is fitted on the training data to account for censored observations, and Inverse Probability of Censoring Weights (IPCW) are computed for both training and validation sets to adjust for censoring bias.
3. **Feature Encoding and Alignment:** Categorical variables are transformed into dummy variables to facilitate model training. To maintain consistency across folds, the validation subset is reindexed to match the training subset's feature space, filling any missing columns with zeros.
4. **Performance Evaluation:** The trained model predicts the survival probabilities at $\tau = 3$ years for the validation subset. The IPCW Mean Squared Error (MSE) is then calculated to assess the model's predictive accuracy, accounting for censored data.
5. **Aggregation of Results:** The IPCW MSE scores from all folds are averaged to obtain a mean performance metric for the given hyperparameter set.

Hyperparameter Grid and Optimization

To explore the hyperparameter space comprehensively, we defined a parameter grid encompassing various configurations:

- `B`: Number of decision trees in the ensemble ($\frac{n}{2} \approx 8000$).
- `max_depth`: Maximum depth of each decision tree (e.g., [5, 6, 7, 8, 9, 10]).
- `min_samples_split`: Minimum number of samples required to split an internal node (e.g., [10, 20, 30, 40, 50]).
- `max_features`: Number of features to consider when looking for the best split (e.g., ['sqrt', 'log2']).

Given the extensive search space, a grid search strategy was employed in conjunction with parallel computing to expedite the hyperparameter tuning process. Each combination of hyperparameters was evaluated using the `evaluate_params` function across all folds, and the mean IPCW MSE was recorded.

Selection of Optimal Hyperparameters

After evaluating all hyperparameter combinations, the configuration yielding the lowest mean IPCW MSE was identified as follows:

- `'B'`: 8000
- `'max_depth'`: 10
- `'min_samples_split'`: 40
- `'max_features'`: `'sqrt'`

This combination achieved an IPCW MSE of 0.1438. These optimal hyperparameters were then used to train the final DTBC model on the entire training dataset.

Implementation Details

The hyperparameter tuning process was carried out using Python, leveraging libraries such as `pandas` for data manipulation, `lifelines` for survival analysis, and `scikit-learn` for machine learning utilities. The custom `DecisionTreeBaggingClassifier` class is implemented as described in Section 3.1.4.

4.3 Variance Estimates

To illustrate the capabilities of the variance estimators applied to the DTBC model's predictions, we selected three representative patients from the dataset whose three-year survival probabilities were predicted. The average patient was identified using the Gower distance metric, which quantifies the similarity between instances.[1] Following the prediction of survival probabilities for all patients in the dataset, we identified the patients with the lowest and

[1] [Gow71]

highest predicted survival probabilities. These selections provide a comprehensive view of the variance estimators' performance across different risk profiles. Table 4.2 summarizes the covariates of these three patients along with their corresponding survival probabilities.

Table 4.2: Covariates of Selected Patients

Covariate	**Survival Probability (%)**		
	Low	**Mean**	**High**
`time`	145.0	1826.25	1319.0
`event`	1	0	0
`donor_age_years`	76.0	57.0	18.0
`donor_sex`	male	male	male
`donor_height_cm`	157.0	175.0	175.0
`donor_weight_kg`	68.0	78.0	65.0
`donor_creatinin_umol_per_l`	70.9	25.6	54.8
`donor_diabetes`	True	False	True
`donor_hypertension`	True	False	False
`donor_smoking`	False	False	False
`donor_hcv`	False	False	False
`recipient_age_years`	72.70	52.77	49.51
`recipient_sex`	male	male	female
`recipient_height_cm`	178.0	173.0	175.0
`recipient_weight_kg`	117.0	84.0	61.0
`recipient_bloodtransfusion`	True	False	False
`recipient_dialysis_years`	4.12	4.88	8.28
`recipient_hcv`	False	False	False
`recipient_pra`	False	False	False
`transplant_cold_ischemia_time_min`	538	829	565
`destination`	Regional	Regional	Regional
Survival Probability $\hat{S}\left(1095 \mid X_{patient(i)}\right)$	40.83 %	89.44 %	95.03 %

The `time` variable is measured in days. The patient with the lowest survival probability experienced an event after approximately 0.4 years. The average patient was censored after approximately 5 years, and the patient with the highest survival probability was censored after approximately 3.6 years. We subsequently applied the following variance estimators to these three predictions:

- *Infinitesimal-Jackknife-after-weighted-Bootstrap (IJK-AWB)*
- *Infinitesimal-Jackknife-after-weighted-Bootstrap-unbiased (IJK-AWB-U)*
- *Jackknife-after-Bootstrap-unbiased (JK-AB-U)*
- *Bootstrap* (with $B_1 = 200$)

The results of the variance estimators are presented in Figure 4.1. This figure illustrates the variance estimates for each of the three selected patients across the different variance estimation methods.

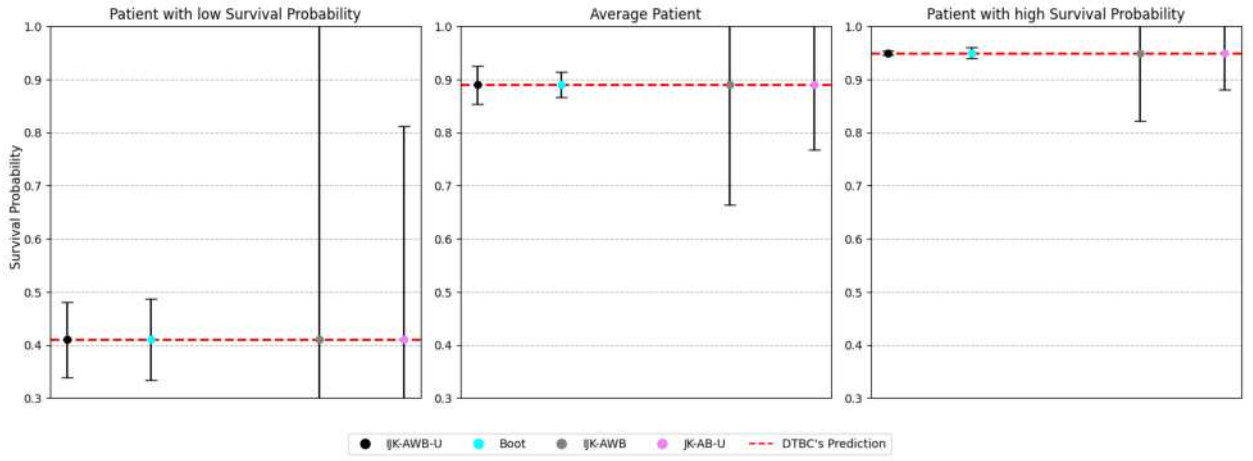


Figure 4.1: Variance Estimates on predicted Survival Probabilities at $\tau = 3$ years of 3 Patients from the TxReg Dataset. The 4 error bars correspond to $\hat{S}\left(1095 \mid X_{patient(i)}\right) \pm 1.96 \cdot \hat{\sigma}$, which provides an approximate 95% confidence interval of the prediction. The estimated standard deviations ($\hat{\sigma}$) for each error bar is based on an estimator from the legend.

The figure reveals a clear trend: all four estimators exhibit the widest confidence intervals for the patient with the lowest survival probability (left panel), indicating the highest uncertainty. As survival probability increases (middle and right panels), the confidence intervals become progressively narrower for each method, suggesting increased confidence in the predictions. This pattern is consistent across all estimators, reflecting the DTBC model's greater stability in estimating higher survival probabilities.

In previous chapters, we identified the *Bootstrap* variance estimator as a "gold standard" due to its stability and robustness in simulated settings. Here, the *IJK-AWB-U* estimator produces variance estimates closely aligned with those of the *Bootstrap* method, which suggests that *IJK-AWB-U* performs reliably on this real dataset as well. This similarity supports the choice of *IJK-AWB-U* as a computationally efficient yet accurate alternative for variance estimation in survival predictions.

Conversely, the *IJK-AWB* estimator, which lacks bias correction, consistently shows the widest intervals, particularly for the patient with low survival probability, indicating a tendency to overestimate the variance. The *JK-AB-U* estimator performs slightly better than *IJK-AWB* but still overestimates variance compared to the *Bootstrap* and *IJK-AWB-U* methods.

In summary, Figure 4.1 supports the insights gained from the simulation study. The *IJK-AWB-U* estimator provides narrower confidence intervals that closely align with the *Bootstrap* estimator, which was identified as the most stable approach in the simulated settings. This alignment suggests that the *IJK-AWB-U* estimator may offer a reliable approximation of variance in practical applications. In contrast, the *IJK-AWB* and *JK-AB-U* estimators consistently produce wider confidence intervals, particularly for lower survival probabilities, indicating a tendency to overestimate variance. This overestimation underscores the limitations of these methods for practical use in variance estimation within the DTBC model with survival data.

5 Conclusion

The present study addressed the question, "How can variance estimation methods for IPC-weighted classification models be developed to provide reliable estimates in the context of censored data?" A comprehensive analysis of existing methods for both weighted and unweighted classification models was conducted, alongside the development and evaluation of a novel approach. Specifically, a new method, the *Infinitesimal-Jackknife-after-weighted-Bootstrap-unbiased* (IJK-AWB-U), was developed to provide unbiased variance estimates for bagged learners utilizing IPC-weighted resampling.

The results indicate that the IJK-AWB-U estimator provides robust and efficient variance estimation, especially under conditions of moderate to low censoring rates and with larger training sample sizes. A simulation study, designed within the ADEMP framework, demonstrated that this method outperformed the gold standard, the nonparametric Bootstrap, in terms of computational efficiency while maintaining high accuracy. This positions the IJK-AWB-U estimator as a valuable tool for predictive modeling in real-world scenarios, particularly when working with censored data in medical research and survival analysis.

Moreover, the practical application to the TxReg dataset validated the estimator's reliability, showing that the IJK-AWB-U estimator produced confidence intervals that closely align with those generated by the Bootstrap estimator, widely regarded as the gold standard in variance estimation. This close alignment underscores the practical utility of the IJK-AWB-U estimator, especially as traditional estimators such as the Jackknife-after-Bootstrap showed significantly higher bias under IPC-weighted resampling, leading to overestimated variance and potential inaccuracies in predictive intervals. By focusing on bagged learners based on decision trees, this study demonstrated that IJK-AWB-U can be effectively adapted to work within specific machine learning frameworks, such as decision tree ensembles, while delivering unbiased results.

This work contributes methodologically by adapting existing variance estimation techniques to censored data frameworks. Specifically, the IJK-AWB-U estimator builds upon Wager's Infinitesimal Jackknife approach for unweighted bagged learners, extending it to address key challenges associated with IPC-weighted resampling. Moreover, by implementing an effective bias correction that adjusts for the bias introduced by finite bootstrap samples, the IJK-AWB-U estimator provides more accurate variance estimates. This adaptation makes the IJK-AWB-U estimator a robust choice for censored data applications. The extensive simulation study provides insight into the estimator's performance across different censoring rates, event proportions, and sample sizes. The results demonstrate that the estimator maintains stable bias and coverage properties under moderate to low degrees of censoring, suggesting broad applicability and resilience of the method across various levels of censoring. However, under conditions of high censoring, the estimator's performance diminishes, indicating that its reliability may be limited in extreme censorship contexts. This

highlights the importance of selecting appropriate methods based on the degree of censoring in the data. Additionally, by addressing the inadequacies of the Jackknife-after-Bootstrap under IPC-weighting, this study clarifies the limitations of conventional methods that do not incorporate weights in their estimations, demonstrating that these methods fail to provide reliable results in weighted contexts.

Limitations and Future Directions

While this thesis makes significant advancements, several limitations must be acknowledged. First, the study focuses exclusively on bagged learners based on decision trees, which, while highly relevant, may limit generalizability to other machine learning architectures, such as gradient-boosting models or neural networks. Additionally, the simulation study was based on specific data-generating mechanisms, which may not encompass all real-world scenarios. While extensive, the simulation scope does not fully capture all variability types encountered in diverse applications of censored data. Future work could expand on this by testing the IJK-AWB-U estimator across a broader range of data types and model frameworks. Furthermore, as shown in this study, the Jackknife-after-Bootstrap method, although traditionally effective under unweighted resampling, performed poorly in IPC-weighted scenarios, underscoring the need for further research into extending existing methods for IPC-weighted applications.

Building on these findings, future research could pursue several promising directions. First, expanding the application of the IJK-AWB-U estimator to diverse datasets and more complex model architectures will be essential to validate its versatility and generalizability. Additionally, exploring hybrid methods that combine the strengths of the IJK-AWB-U estimator with other machine learning frameworks may yield improvements in predictive accuracy. Finally, applying the estimator in various fields, including finance, engineering, and epidemiology, would provide further insights into its robustness and adaptability beyond medical research contexts.

In summary, this thesis presents a significant advancement in variance estimation for IPC-weighted classification models, addressing the complex challenges posed by censored data. By developing and validating the IJK-AWB-U estimator, the study provides a new tool that enhances predictive accuracy and reliability, paving the way for more sophisticated and tailored approaches to uncertainty estimation in censored data analysis.

6 Appendix

6.0.1 Repository

All experiments and figures presented in this thesis were generated using the codebase provided in the repository https://github.com/rehan-b/Masterarbeit__Butt. This repository contains all scripts and configurations required to reproduce the analyses and visualizations, ensuring transparency and reproducibility of the results.

6.0.2 Parameters for Data Generation in Simulation Study

The following parameters were used for the Data Generating Mechanism (cf. Section 3.1.2) in the Simulation Study.

```
params_data = [

# Cens_prop = 0.1 //  Event Proportion = 0.09
{ 'shape_weibull': 1,          ## ( = k )
      'scale_weibull_base': 24_300,
      'rate_censoring': 0.003,
      'b_bloodp': -0.405, 'b_diab': -0.4, 'b_age': -0.05,
      'b_bmi': -0.01, 'b_kreat': -0.2,
      'n': n, 'seed': seed, 'tau': 37},
# Cens_prop = 0.1 //  Event Proportion = 0.18
{ 'shape_weibull': 1,          ## ( = k )
      'scale_weibull_base': 10803.76159628643   ,
      'rate_censoring':0.003170578469623819    ,
      'b_bloodp': -0.405, 'b_diab': -0.4, 'b_age': -0.05,
      'b_bmi': -0.01, 'b_kreat': -0.2,
      'n': n, 'seed': seed, 'tau': 37},
# Cens_prop = 0.1 //  Event Proportion = 0.27
{ 'shape_weibull': 1,          ## ( = k )
      'scale_weibull_base': 6539.41883092019  ,
      'rate_censoring':0.0033904243453215187,
      'b_bloodp': -0.405, 'b_diab': -0.4, 'b_age': -0.05,
      'b_bmi': -0.01, 'b_kreat': -0.2,
      'n': n, 'seed': seed, 'tau': 37},
# Cens_prop = 0.1 //  Event Proportion = 0.36
{ 'shape_weibull': 1,          ## ( = k )
      'scale_weibull_base':  4453.164150258696    ,
      'rate_censoring': 0.003624326851330594   ,
      'b_bloodp': -0.405, 'b_diab': -0.4, 'b_age': -0.05,
      'b_bmi': -0.01, 'b_kreat': -0.2,
      'n': n, 'seed': seed, 'tau': 37},

# Cens_prop = 0.3 //  Event Proportion = 0.07
{ 'shape_weibull': 1,          ## ( = k )
      'scale_weibull_base':   28000   ,
      'rate_censoring': 0.01,
      'b_bloodp': -0.405, 'b_diab': -0.4, 'b_age': -0.05,
      'b_bmi': -0.01, 'b_kreat': -0.2,
      'n': n, 'seed': seed, 'tau': 37},
# Cens_prop = 0.3 //  Event Proportion = 0.14
{ 'shape_weibull': 1,          ## ( = k )
      'scale_weibull_base':  12463.811039838654   ,
      'rate_censoring': 0.010725364504143705      ,
      'b_bloodp': -0.405, 'b_diab': -0.4, 'b_age': -0.05,
      'b_bmi': -0.01, 'b_kreat': -0.2,
```

```
      'n': n, 'seed': seed, 'tau': 37},
# Cens_prop = 0.3 //  Event Proportion = 0.21
{ 'shape_weibull': 1,          ## ( = k )
      'scale_weibull_base':7500     ,
      'rate_censoring':  0.011600103696876245    ,
      'b_bloodp': -0.405, 'b_diab': -0.4, 'b_age': -0.05,
      'b_bmi': -0.01, 'b_kreat': -0.2,
      'n': n, 'seed': seed, 'tau': 37},
# Cens_prop = 0.3 //  Event Proportion = 0.28
{ 'shape_weibull': 1,          ## ( = k )
      'scale_weibull_base':  5122.0241237382925    ,
      'rate_censoring':  0.012470507897824007      ,
      'b_bloodp': -0.405, 'b_diab': -0.4, 'b_age': -0.05,
      'b_bmi': -0.01, 'b_kreat': -0.2,
      'n': n, 'seed': seed, 'tau': 37},

# Cens_prop = 0.5 //  Event Proportion = 0.05
{ 'shape_weibull': 1,          ## ( = k )
      'scale_weibull_base': 34000     ,
      'rate_censoring':  0.019578490533008537    ,
      'b_bloodp': -0.405, 'b_diab': -0.4, 'b_age': -0.05,
      'b_bmi': -0.01, 'b_kreat': -0.2,
      'n': n, 'seed': seed, 'tau': 37},
# Cens_prop = 0.5 //  Event Proportion = 0.10
{ 'shape_weibull': 1,          ## ( = k )
      'scale_weibull_base': 15800      ,
      'rate_censoring':  0.02052170406791234   ,
      'b_bloodp': -0.405, 'b_diab': -0.4, 'b_age': -0.05,
      'b_bmi': -0.01, 'b_kreat': -0.2,
      'n': n, 'seed': seed, 'tau': 37},
# Cens_prop = 0.5 //  Event Proportion = 0.15
{ 'shape_weibull': 1,          ## ( = k )
      'scale_weibull_base': 9600      ,
      'rate_censoring':  0.0218      ,
      'b_bloodp': -0.405, 'b_diab': -0.4, 'b_age': -0.05,
      'b_bmi': -0.01, 'b_kreat': -0.2,
      'n': n, 'seed': seed, 'tau': 37},
# Cens_prop = 0.5 //  Event Proportion = 0.20
{ 'shape_weibull': 1,          ## ( = k )
      'scale_weibull_base': 6600       ,
      'rate_censoring':  0.022901136686777616      ,
      'b_bloodp': -0.405, 'b_diab': -0.4, 'b_age': -0.05,
      'b_bmi': -0.01, 'b_kreat': -0.2,
      'n': n, 'seed': seed, 'tau': 37},

# Cens_prop = 0.7 //  Event Proportion = 0.03
{ 'shape_weibull': 1,          ## ( = k )
      'scale_weibull_base': 45_000       ,
      'rate_censoring':  0.034   ,
      'b_bloodp': -0.405, 'b_diab': -0.4, 'b_age': -0.05,
      'b_bmi': -0.01, 'b_kreat': -0.2,
      'n': n, 'seed': seed, 'tau': 37},
# Cens_prop = 0.7 //  Event Proportion = 0.06
{ 'shape_weibull': 1,          ## ( = k )
      'scale_weibull_base': 21700       ,
      'rate_censoring':  0.0352   ,
      'b_bloodp': -0.405, 'b_diab': -0.4, 'b_age': -0.05,
      'b_bmi': -0.01, 'b_kreat': -0.2,
      'n': n, 'seed': seed, 'tau': 37},
# Cens_prop = 0.7 //  Event Proportion = 0.09
{ 'shape_weibull': 1,          ## ( = k )
      'scale_weibull_base': 13_000       ,
      'rate_censoring':  0.0375   ,
      'b_bloodp': -0.405, 'b_diab': -0.4, 'b_age': -0.05,
      'b_bmi': -0.01, 'b_kreat': -0.2,
      'n': n, 'seed': seed, 'tau': 37},
# Cens_prop = 0.7 //  Event Proportion = 0.12
{ 'shape_weibull': 1,          ## ( = k )
      'scale_weibull_base': 9115.851814783131        ,
      'rate_censoring':  0.04021055606963396   ,
      'b_bloodp': -0.405, 'b_diab': -0.4, 'b_age': -0.05,
      'b_bmi': -0.01, 'b_kreat': -0.2,
```

```
      'n': n, 'seed': seed, 'tau': 37}
                ]
```

Listing 6.1: Parameter for Data-Generating Mechanism for $k = 1$

```
params_data = [

# Cens_prop = 0.1 //  Event Proportion = 0.09
{ 'shape_weibull': 1.5,   ##(( = k )
        'scale_weibull_base': 12239.657909989573 ,
        'rate_censoring':0.002923945373663359 ,
        'b_bloodp': -0.405, 'b_diab': -0.4, 'b_age': -0.05,
        'b_bmi': -0.01, 'b_kreat': -0.2,
        'n': n, 'seed': seed, 'tau': 37},
# Cens_prop = 0.1 //  Event Proportion = 0.18
{ 'shape_weibull': 1.5,   ##(( = k )
        'scale_weibull_base': 6764.6566929711325 ,
        'rate_censoring': 0.0031267247333730632,
        'b_bloodp': -0.405, 'b_diab': -0.4, 'b_age': -0.05,
        'b_bmi': -0.01, 'b_kreat': -0.2,
        'n': n, 'seed': seed, 'tau': 37},
# Cens_prop = 0.1 //  Event Proportion = 0.27
{ 'shape_weibull': 1.5,   ##(( = k )
            'scale_weibull_base': 4750.499036902161   ,
            'rate_censoring':0.003341895652382912   ,
            'b_bloodp': -0.405, 'b_diab': -0.4, 'b_age': -0.05,
            'b_bmi': -0.01, 'b_kreat': -0.2,
            'n': n, 'seed': seed, 'tau': 37},
# Cens_prop = 0.1 //  Event Proportion = 0.36
{ 'shape_weibull': 1.5,   ##(( = k )
            'scale_weibull_base': 3519.924999170495  ,
            'rate_censoring':  0.0036209661533116422 ,
            'b_bloodp': -0.405, 'b_diab': -0.4, 'b_age': -0.05,
            'b_bmi': -0.01, 'b_kreat': -0.2,
            'n': n, 'seed': seed, 'tau': 37},

# Cens_prop = 0.3 //  Event Proportion = 0.07
{ 'shape_weibull': 1.5,   ##(( = k )
            'scale_weibull_base':     12980.954805020172  ,
            'rate_censoring':  0.009892476005579862  ,
            'b_bloodp': -0.405, 'b_diab': -0.4, 'b_age': -0.05,
            'b_bmi': -0.01, 'b_kreat': -0.2,
            'n': n, 'seed': seed, 'tau': 37},
# Cens_prop = 0.3 //  Event Proportion = 0.14
{ 'shape_weibull': 1.5,   ##(( = k )
            'scale_weibull_base':   7479.611749700075    ,
            'rate_censoring': 0.010427842997795981,
            'b_bloodp': -0.405, 'b_diab': -0.4, 'b_age': -0.05,
            'b_bmi': -0.01, 'b_kreat': -0.2,
            'n': n, 'seed': seed, 'tau': 37},
# Cens_prop = 0.3 //  Event Proportion = 0.21
{ 'shape_weibull': 1.5,   ##(( = k )
            'scale_weibull_base':  5156.811483486331    ,
            'rate_censoring':     0.011388821997114692  ,
            'b_bloodp': -0.405, 'b_diab': -0.4, 'b_age': -0.05,
            'b_bmi': -0.01, 'b_kreat': -0.2,
            'n': n, 'seed': seed, 'tau': 37},
# Cens_prop = 0.3 //  Event Proportion = 0.28
{ 'shape_weibull': 1.5,   ##(( = k )
            'scale_weibull_base':3880.8399775438843      ,
            'rate_censoring':  0.011920788360226362   ,
            'b_bloodp': -0.405, 'b_diab': -0.4, 'b_age': -0.05,
            'b_bmi': -0.01, 'b_kreat': -0.2,
            'n': n, 'seed': seed, 'tau': 37},

# Cens_prop = 0.5 //  Event Proportion = 0.05
{ 'shape_weibull': 1.5,   ##(( = k )
            'scale_weibull_base':  14705.860131739864     ,
            'rate_censoring':     0.019500697591904738   ,
            'b_bloodp': -0.405, 'b_diab': -0.4, 'b_age': -0.05,
            'b_bmi': -0.01, 'b_kreat': -0.2,
            'n': n, 'seed': seed, 'tau': 37},
```

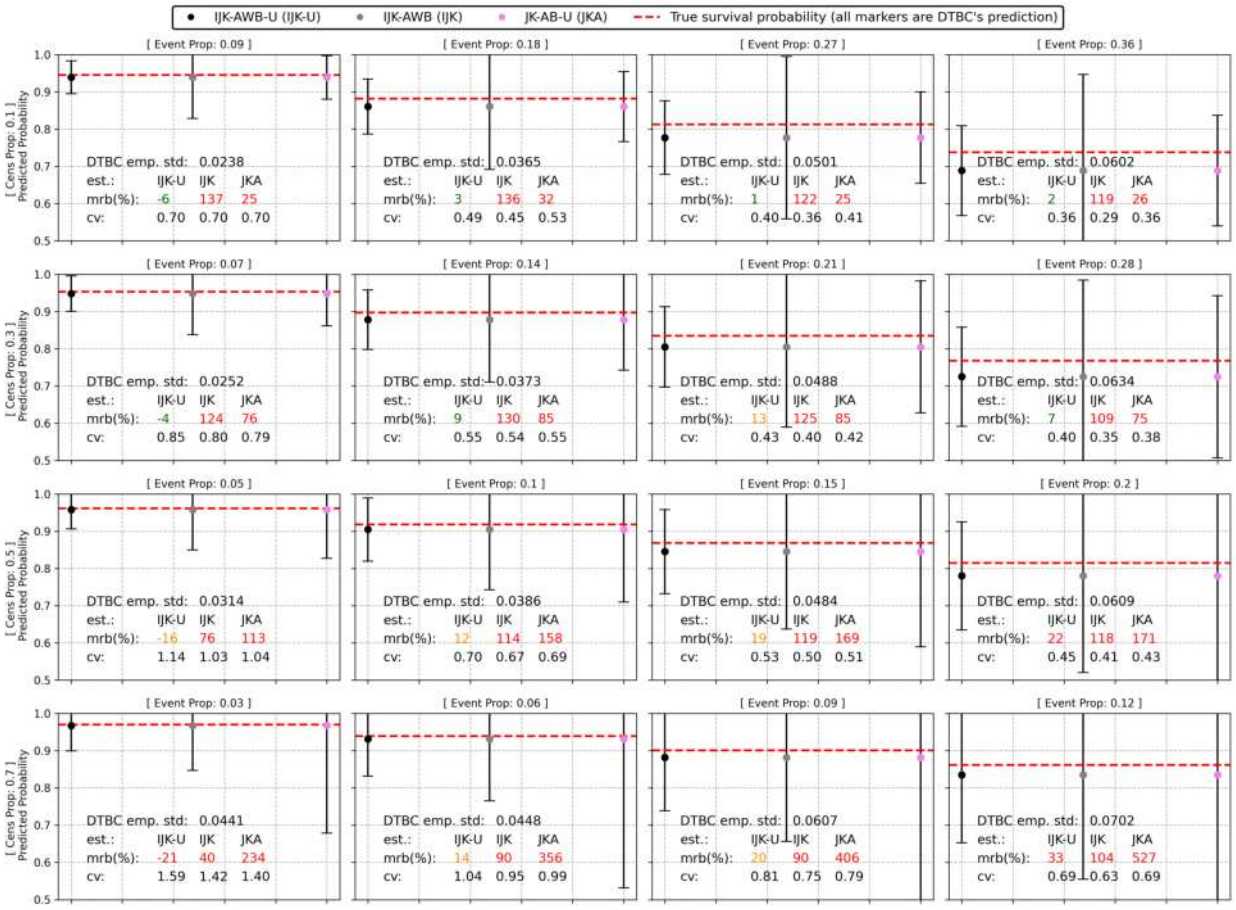
```
# Cens_prop = 0.5 //  Event Proportion = 0.10
{ 'shape_weibull': 1.5,   ##(( = k )
            'scale_weibull_base': 8374.984580837609      ,
            'rate_censoring':    0.020387722883706005  ,
            'b_bloodp': -0.405, 'b_diab': -0.4, 'b_age': -0.05,
            'b_bmi': -0.01, 'b_kreat': -0.2,
            'n': n, 'seed': seed, 'tau': 37},
# Cens_prop = 0.5 //  Event Proportion = 0.15
{ 'shape_weibull': 1.5,   ##(( = k )
            'scale_weibull_base':    5840.913861634944    ,
            'rate_censoring':   0.021592256830888657  ,
            'b_bloodp': -0.405, 'b_diab': -0.4, 'b_age': -0.05,
            'b_bmi': -0.01, 'b_kreat': -0.2,
            'n': n, 'seed': seed, 'tau': 37},
# Cens_prop = 0.5 //  Event Proportion = 0.20
{ 'shape_weibull': 1.5 ,
            'scale_weibull_base':  4400.762312906189     ,
            'rate_censoring':     0.022856524563802574   ,
            'b_bloodp': -0.405, 'b_diab': -0.4, 'b_age': -0.05,
            'b_bmi': -0.01, 'b_kreat': -0.2,
            'n': n, 'seed': seed, 'tau': 37},

# Cens_prop = 0.7 //  Event Proportion = 0.03
{ 'shape_weibull': 1.5,   ##(( = k )
            'scale_weibull_base':    17169.304714916914     ,
            'rate_censoring':     0.03414274145819428   ,
            'b_bloodp': -0.405, 'b_diab': -0.4, 'b_age': -0.05,
            'b_bmi': -0.01, 'b_kreat': -0.2,
            'n': n, 'seed': seed, 'tau': 37},
# Cens_prop = 0.7 //  Event Proportion = 0.06
{ 'shape_weibull': 1.5,   ##(( = k )
            'scale_weibull_base':   10028.241813497492      ,
            'rate_censoring':   0.03561801193145946  ,
            'b_bloodp': -0.405, 'b_diab': -0.4, 'b_age': -0.05,
            'b_bmi': -0.01, 'b_kreat': -0.2,
            'n': n, 'seed': seed, 'tau': 37},
# Cens_prop = 0.7 //  Event Proportion = 0.09
{ 'shape_weibull': 1.5,   ##(( = k )
            'scale_weibull_base':   7090.0587356224605      ,
            'rate_censoring':   0.036824097764675705  ,
            'b_bloodp': -0.405, 'b_diab': -0.4, 'b_age': -0.05,
            'b_bmi': -0.01, 'b_kreat': -0.2,
            'n': n, 'seed': seed, 'tau': 37},
# Cens_prop = 0.7 //  Event Proportion = 0.12
{ 'shape_weibull': 1.5,   ##(( = k )
            'scale_weibull_base':   5597.308204063027      ,
            'rate_censoring':  0.038465201478012315   ,
            'b_bloodp': -0.405, 'b_diab': -0.4, 'b_age': -0.05,
            'b_bmi': -0.01, 'b_kreat': -0.2,
            'n': n, 'seed': seed, 'tau': 37}
               ]
```

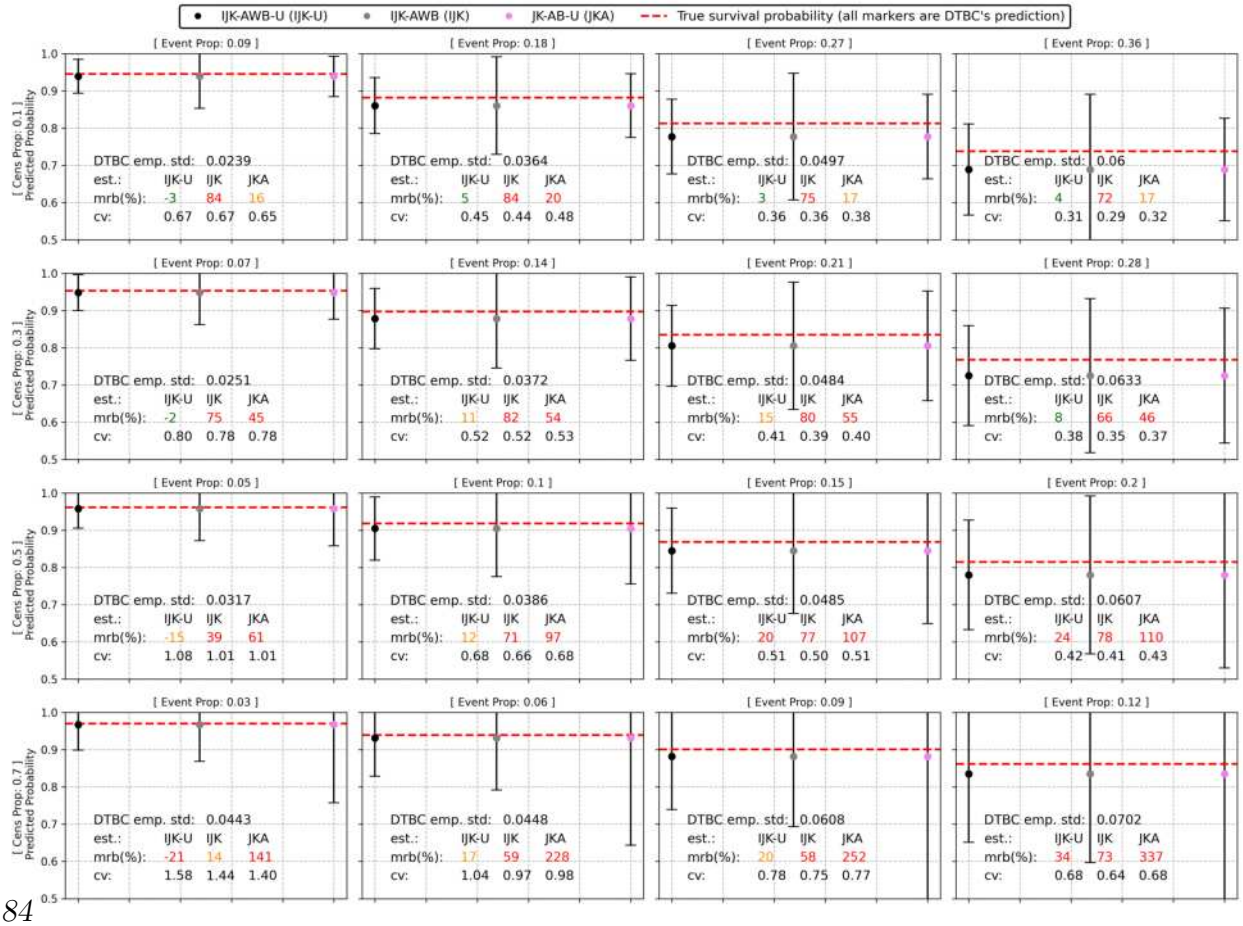
Listing 6.2: Parameter for Data-Generating Mechanism for $k = 1.5$

6.0.3 Simulation Study Results for Variance Estimator's Performance ($k = 1$)

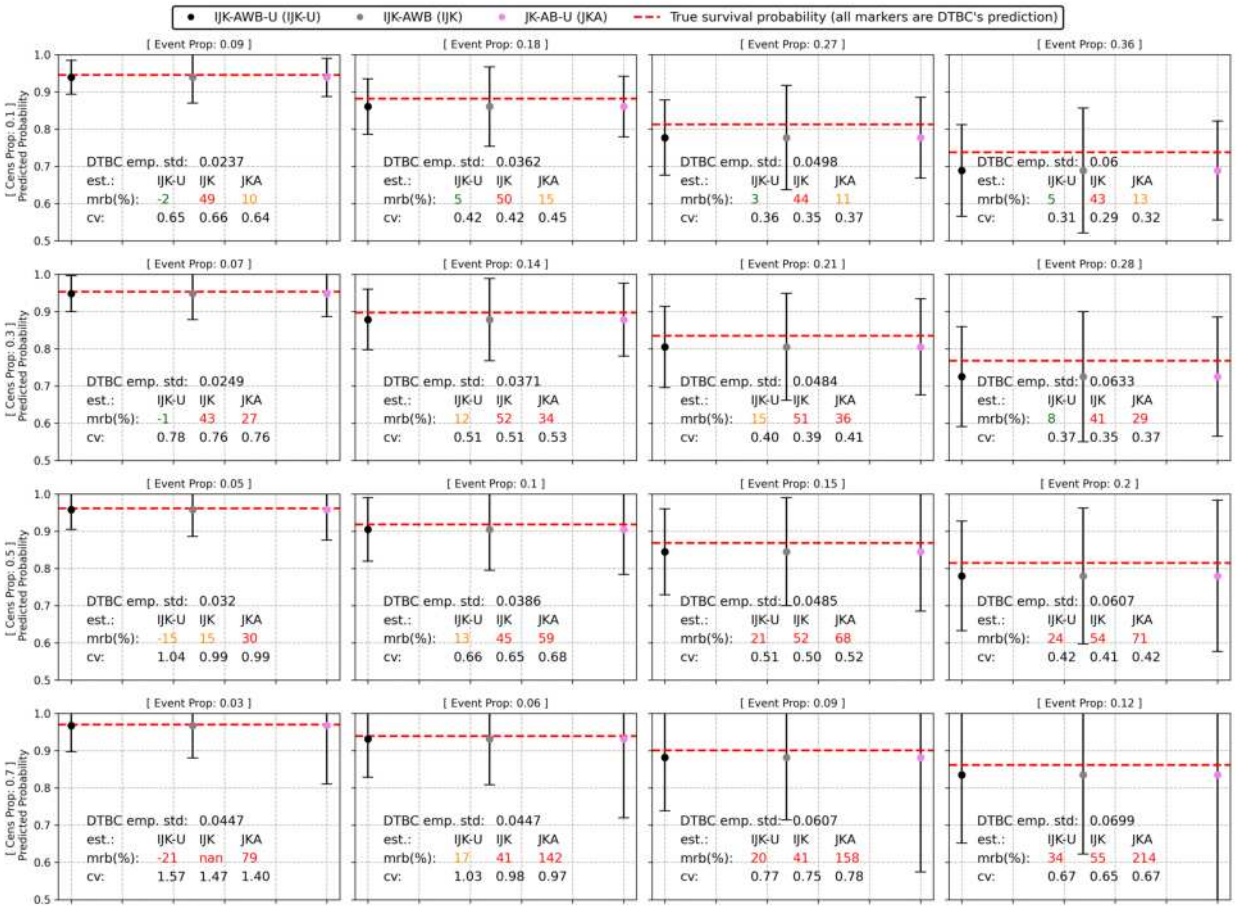
The explanations of these figures are provided in Figure 3.2.



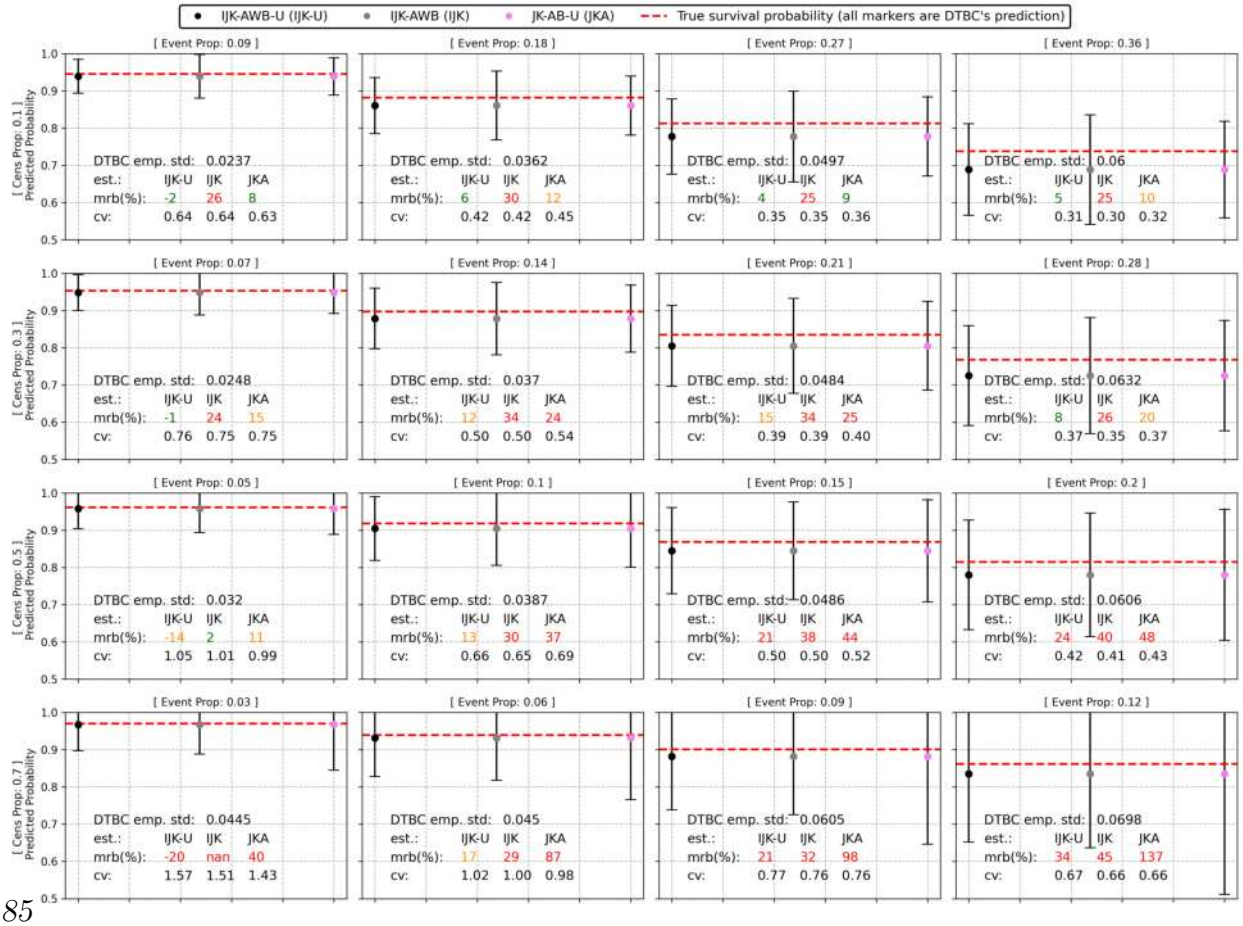
(a) Sim Results for: $k = 1$, $n_{train} = 499$, $B = 500$



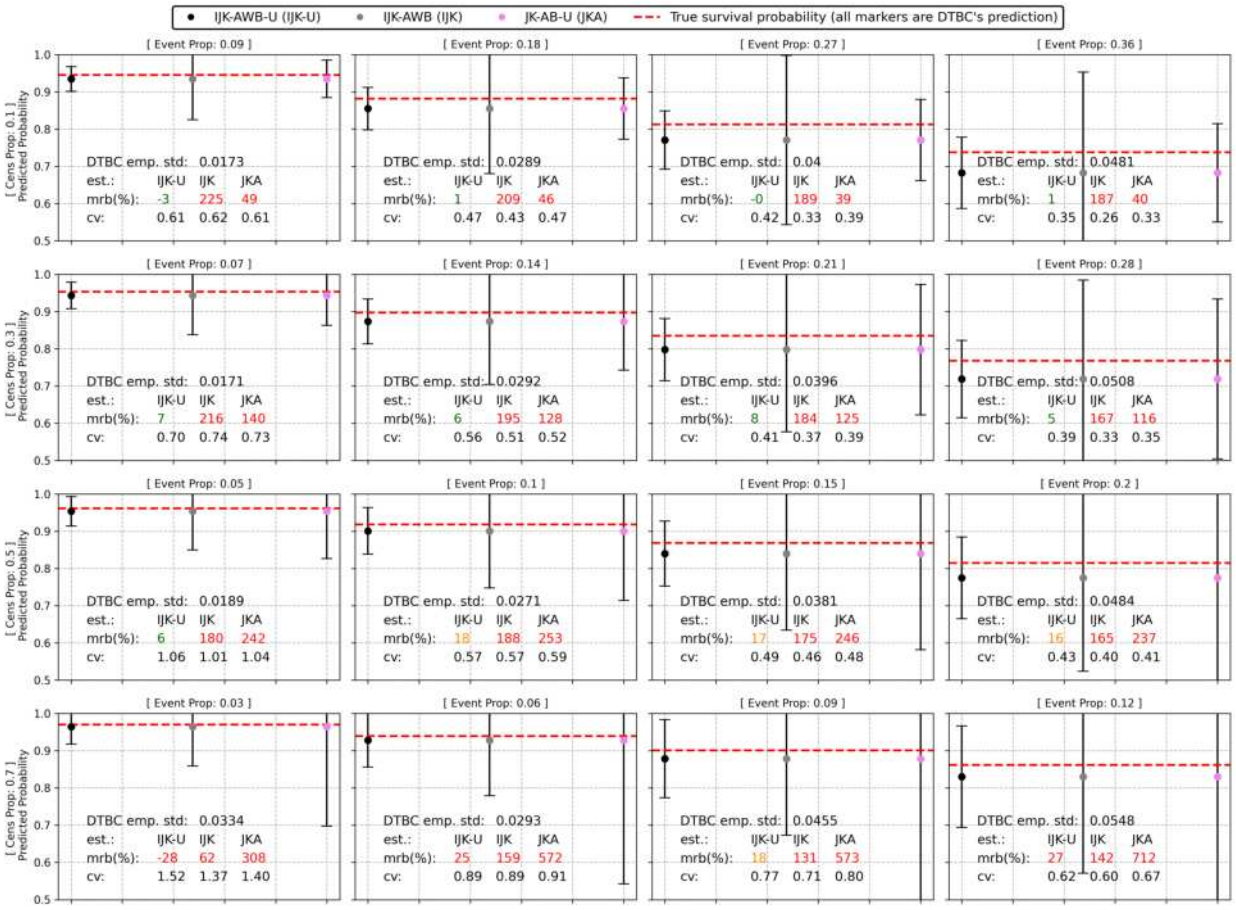
(b) Sim-Results for: $k = 1$, $n_{train} = 499$, $B = 1000$



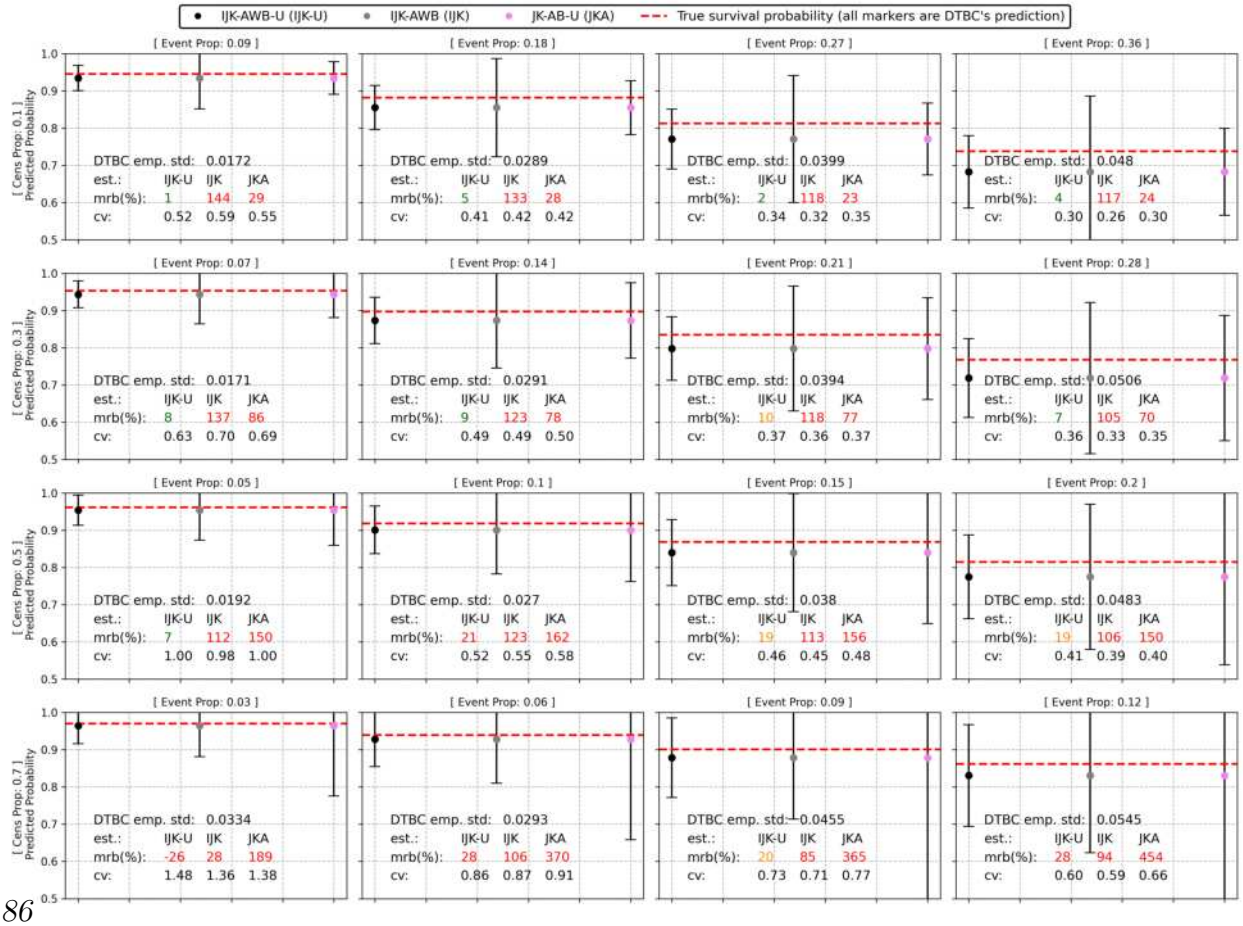
(a) Sim Results for: $k = 1$, $n_{train} = 499$, $B = 2000$



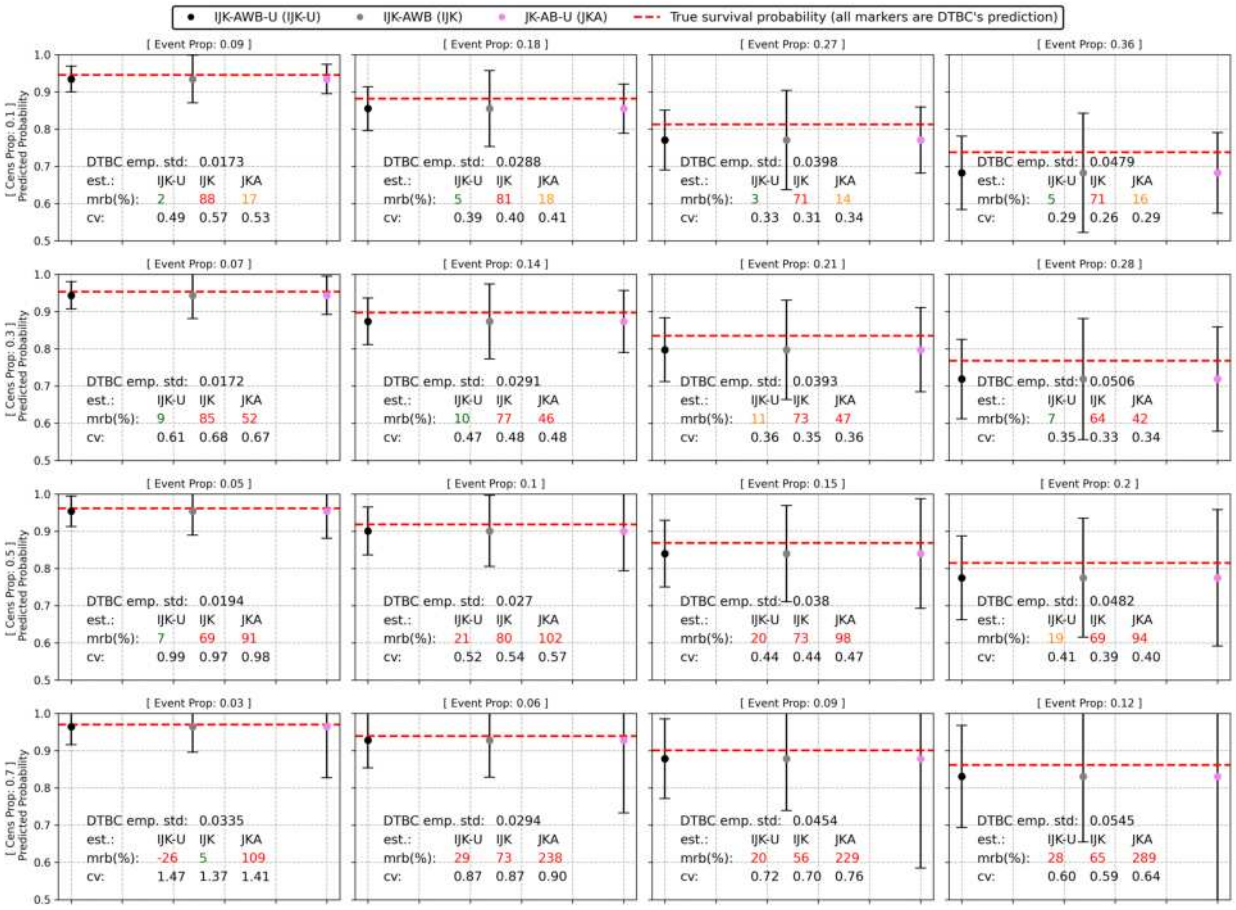
(b) Sim-Results for: $k = 1$, $n_{train} = 499$, $B = 4000$



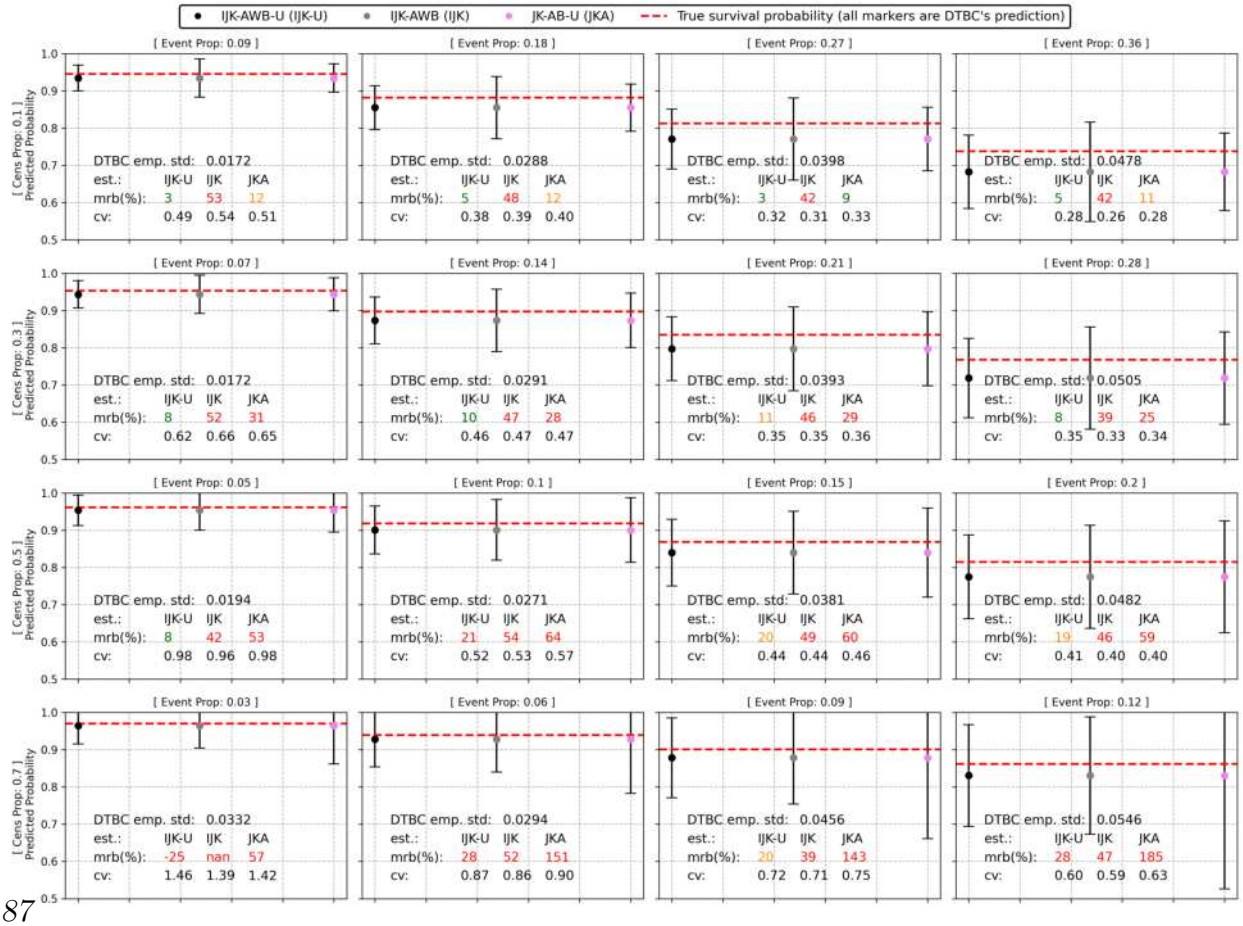
(a) Sim Results for: $k = 1$, $n_{train} = 999$, $B = 500$



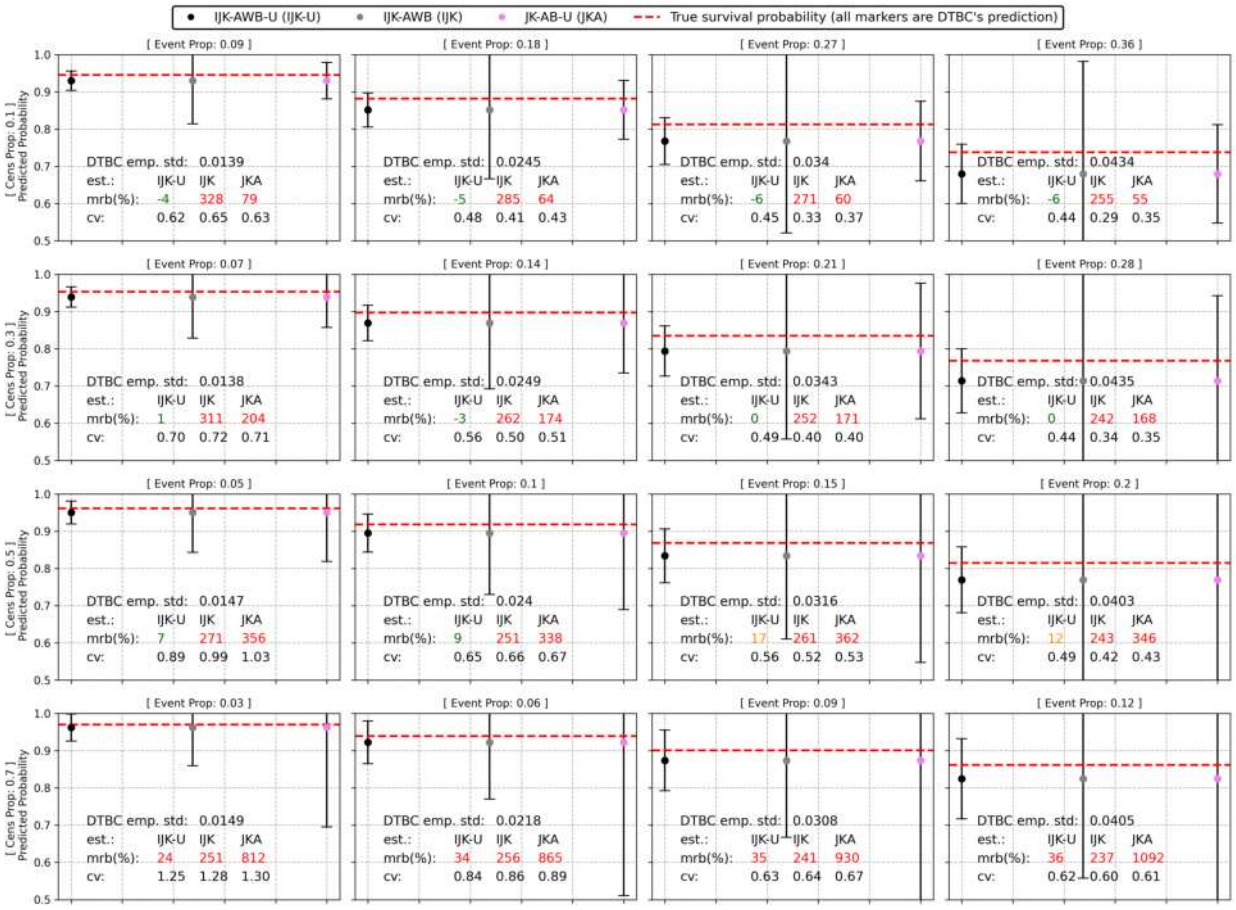
(b) Sim-Results for: $k = 1$, $n_{train} = 999$, $B = 1000$



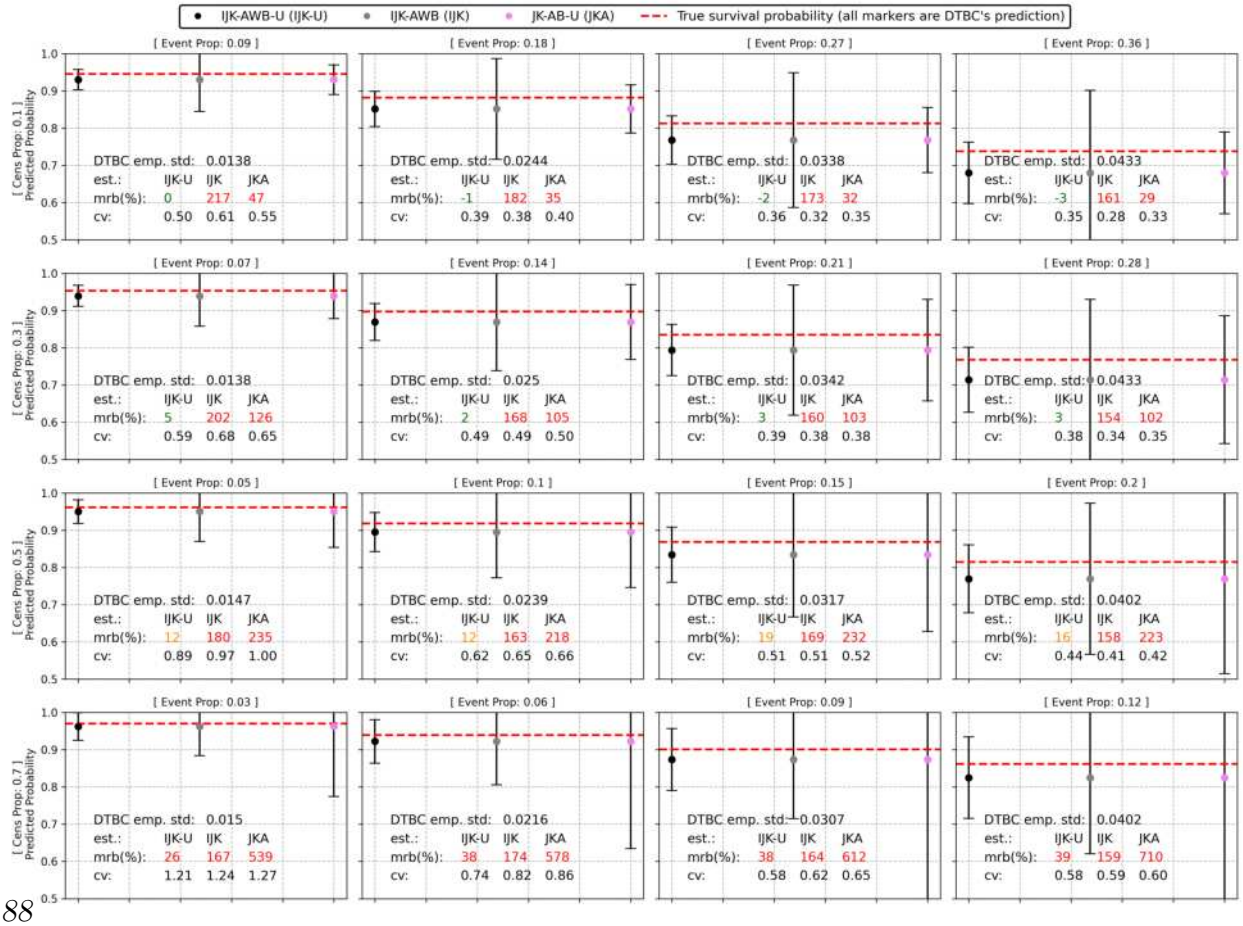
(a) Sim Results for: $k = 1$, $n_{train} = 999$, $B = 2000$



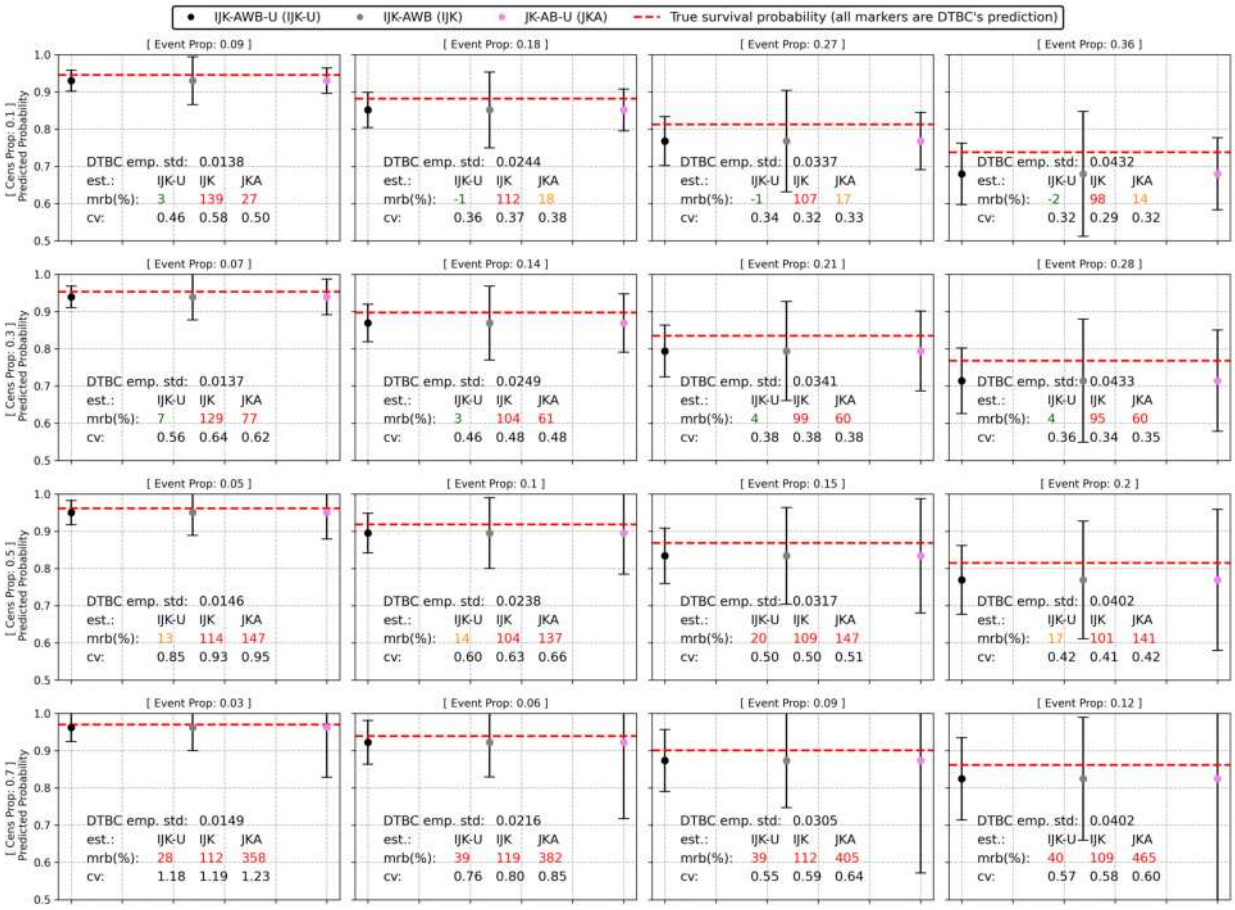
(b) Sim-Results for: $k = 1$, $n_{train} = 999$, $B = 4000$



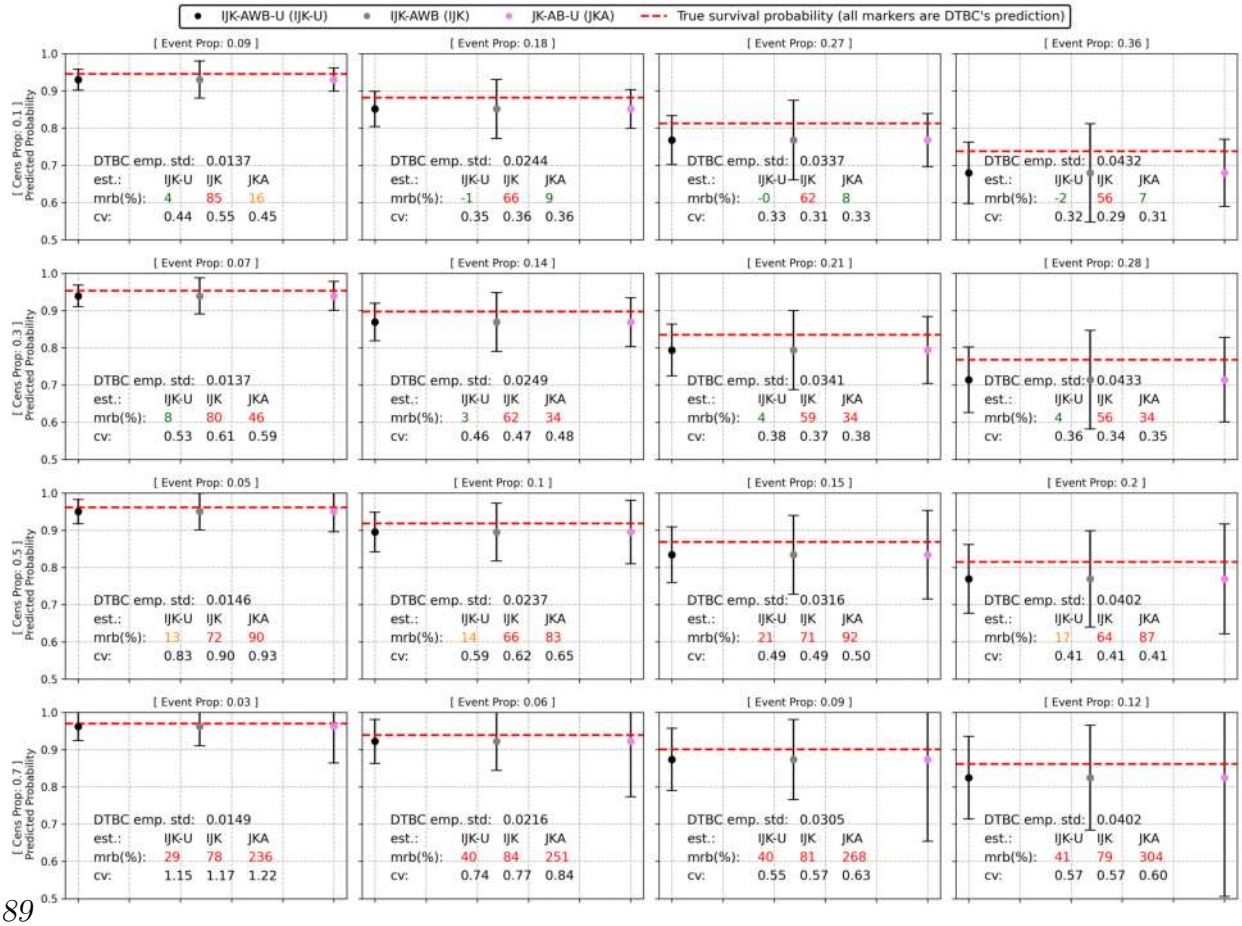
(a) Sim Results for: $k = 1$, $n_{train} = 1999$, $B = 500$



(b) Sim-Results for: $k = 1$, $n_{train} = 1999$, $B = 1000$



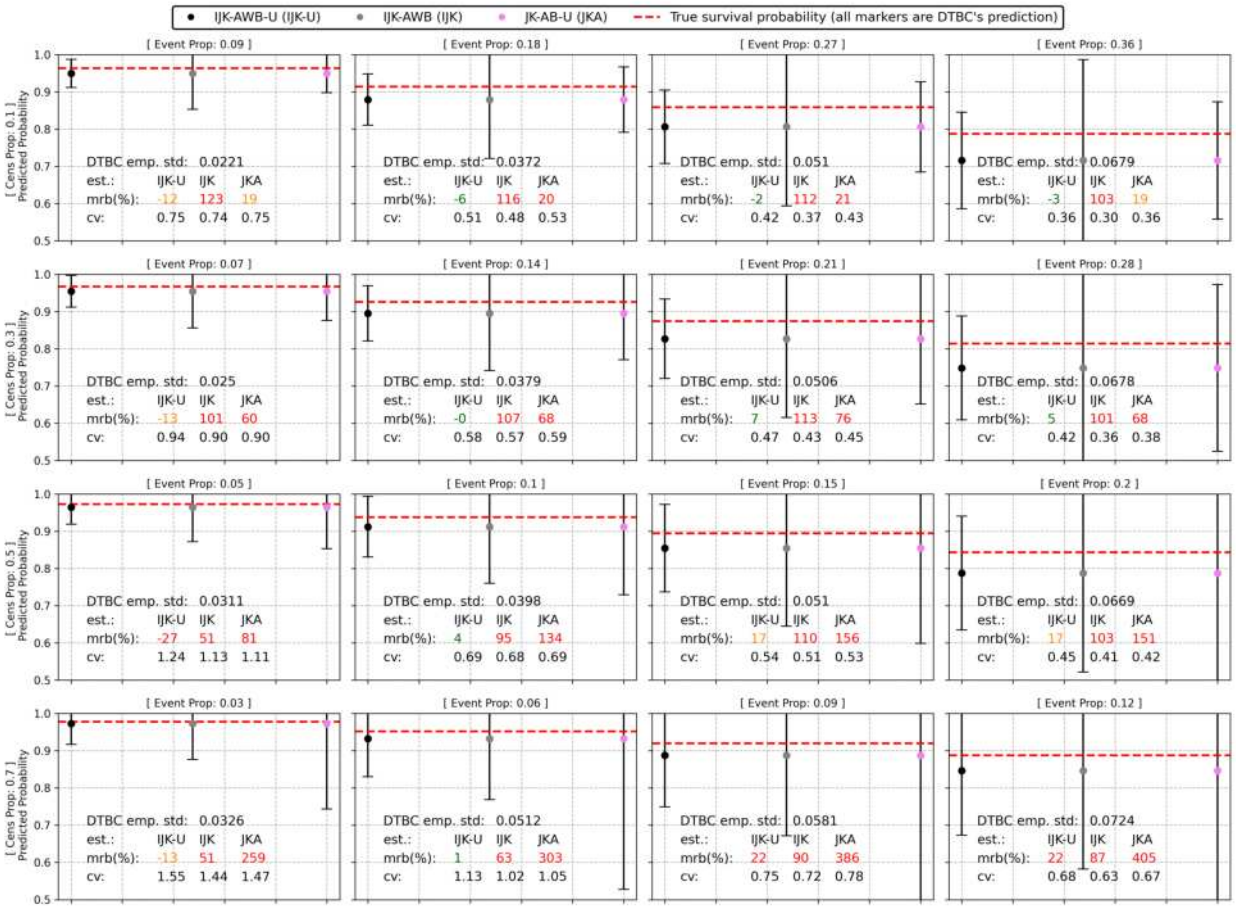
(a) Sim Results for: $k = 1$, $n_{train} = 1999$, $B = 2000$



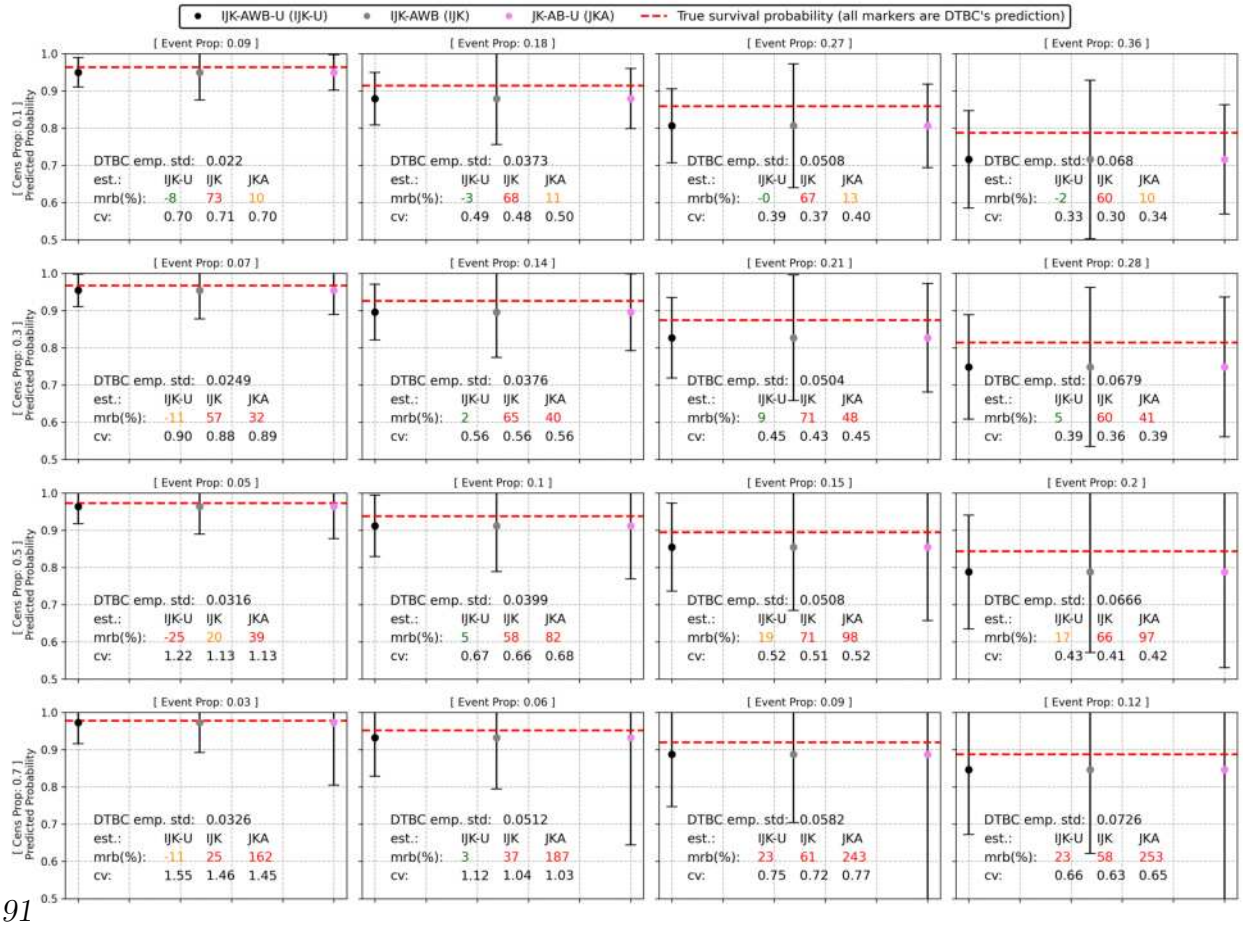
(b) Sim-Results for: $k = 1$, $n_{train} = 1999$, $B = 4000$

6.0.4 Simulation Study Results for Variance Estimator's Performance ($k = 1.5$)

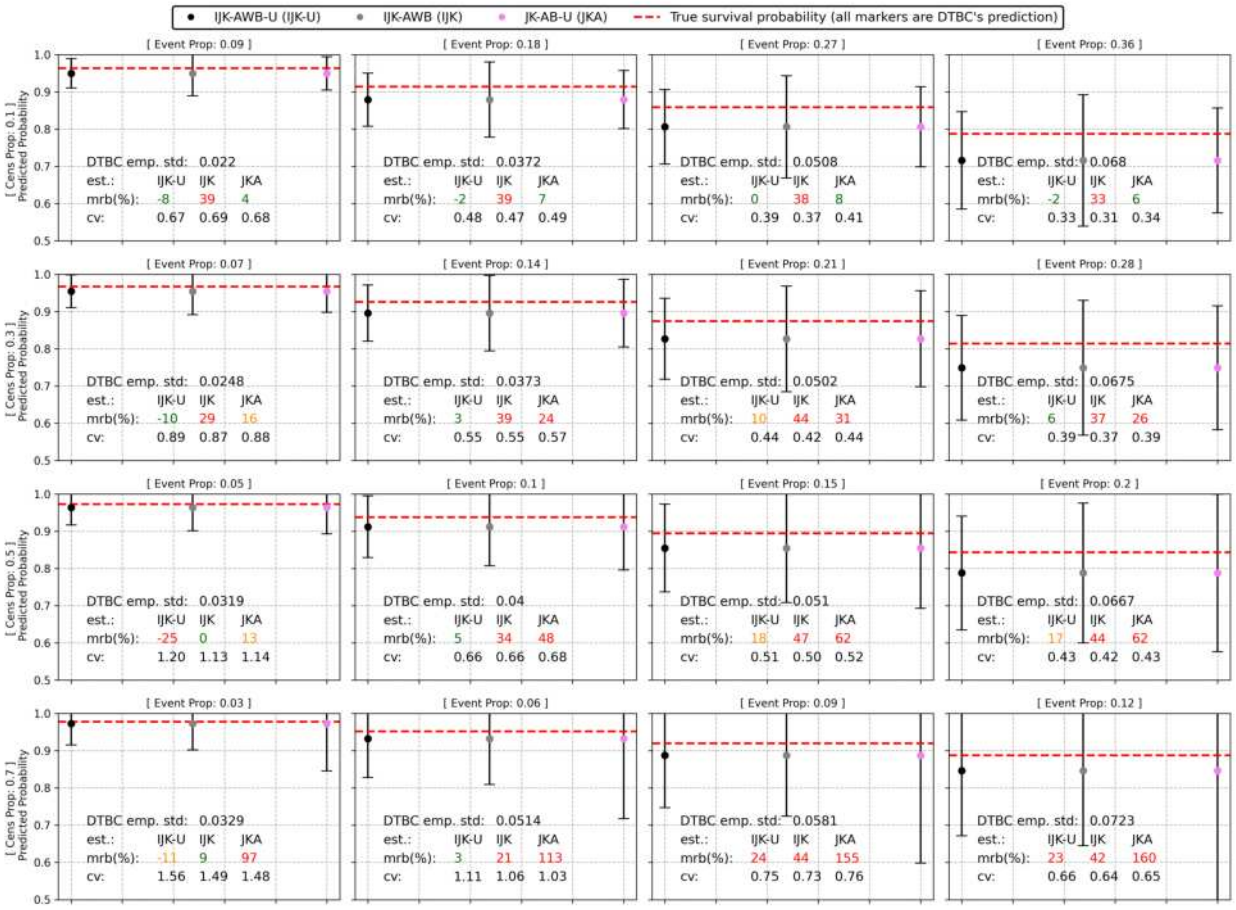
The explanations of these figures are provided in Figure 3.2.



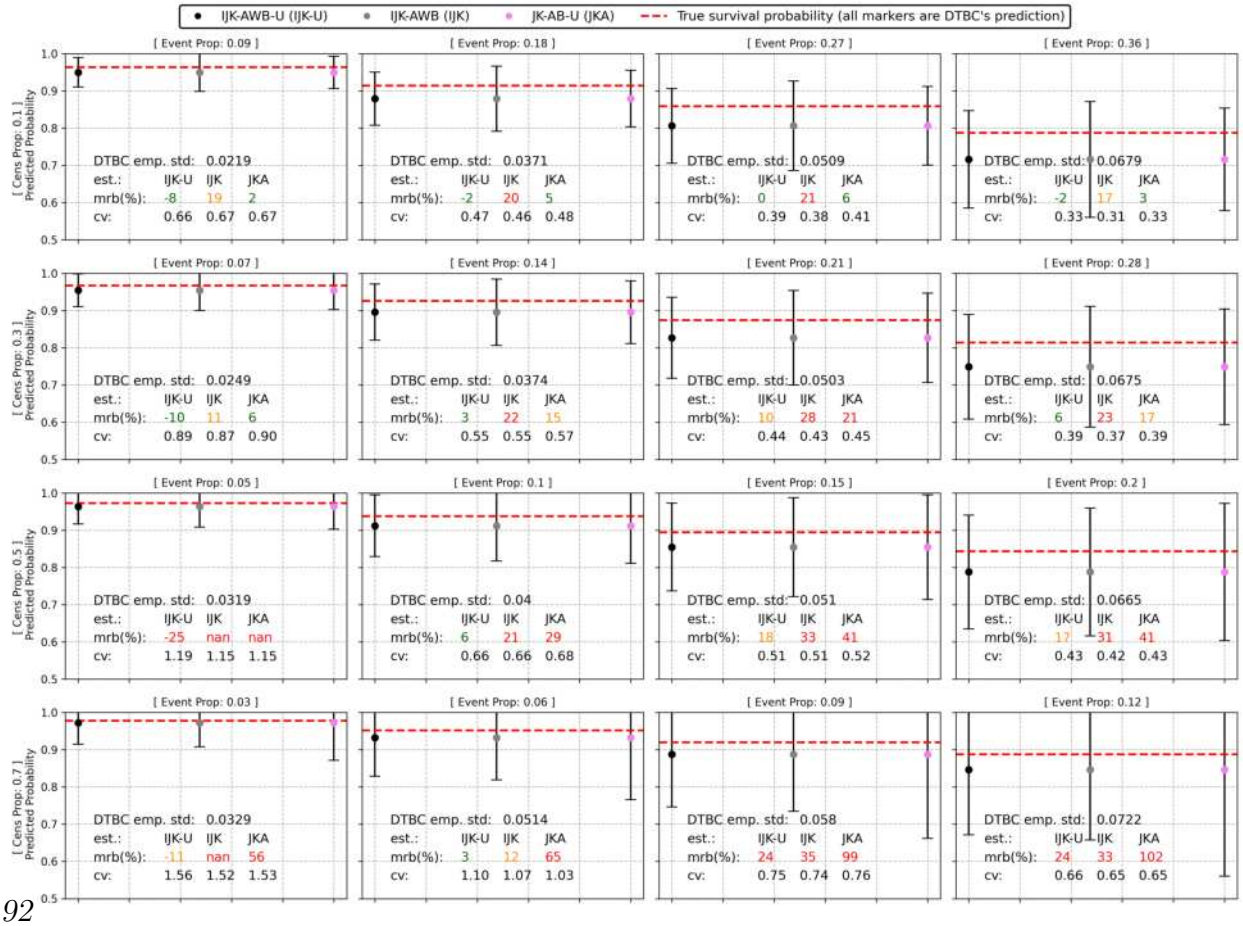
(a) Sim Results for: $k = 1.5$, $n_{train} = 499$, $B = 500$



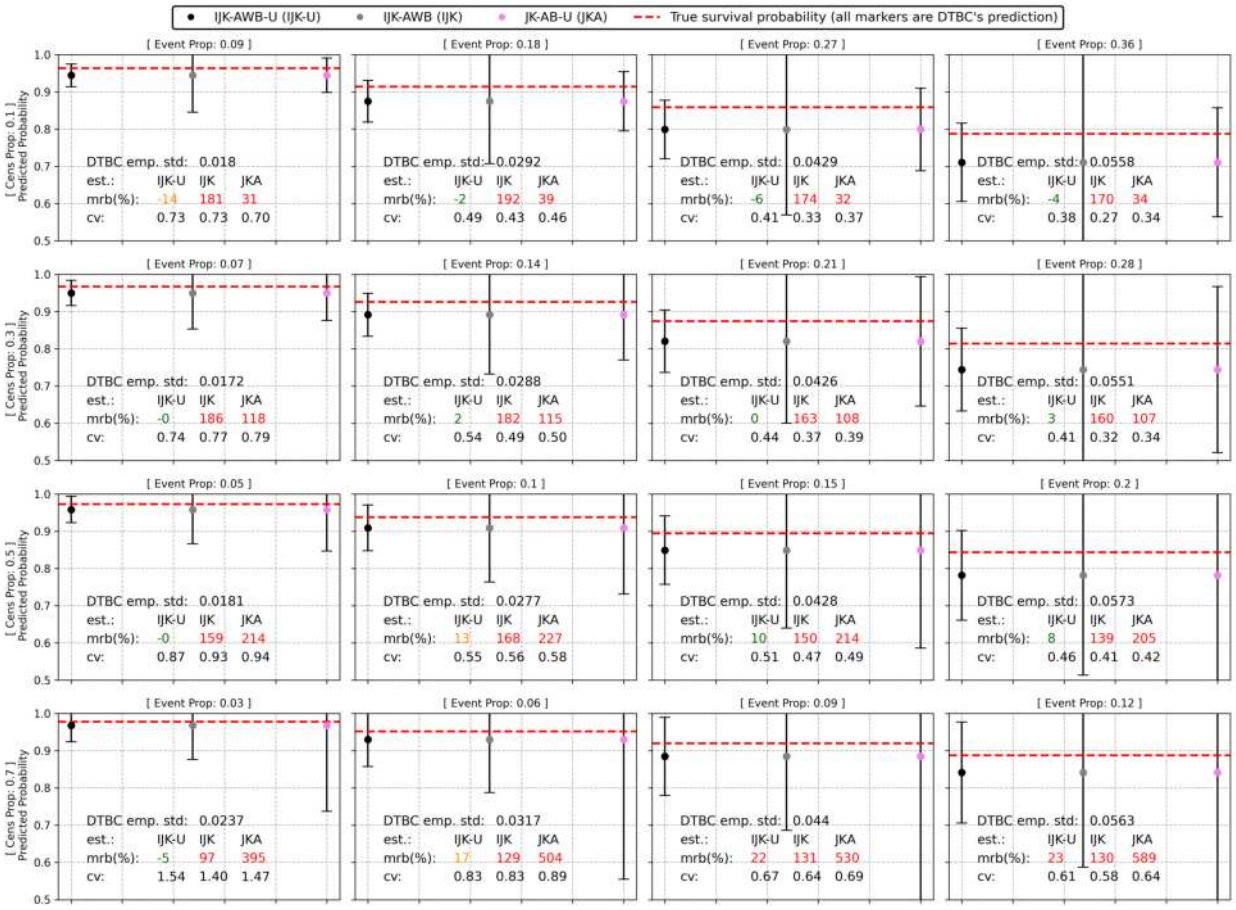
(b) Sim-Results for: $k = 1.5$, $n_{train} = 499$, $B = 1000$



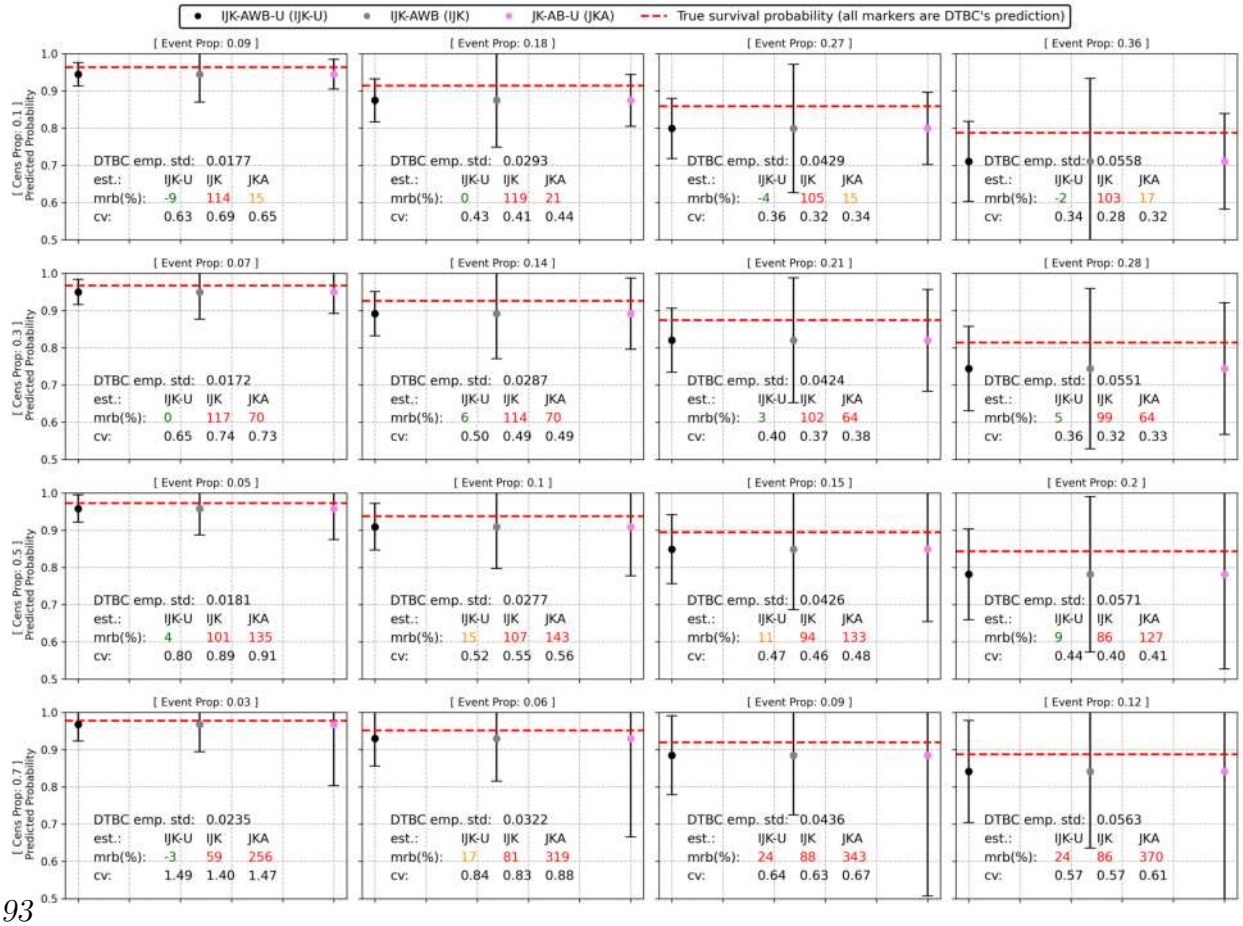
(a) Sim Results for: $k = 1.5$, $n_{train} = 499$, $B = 2000$



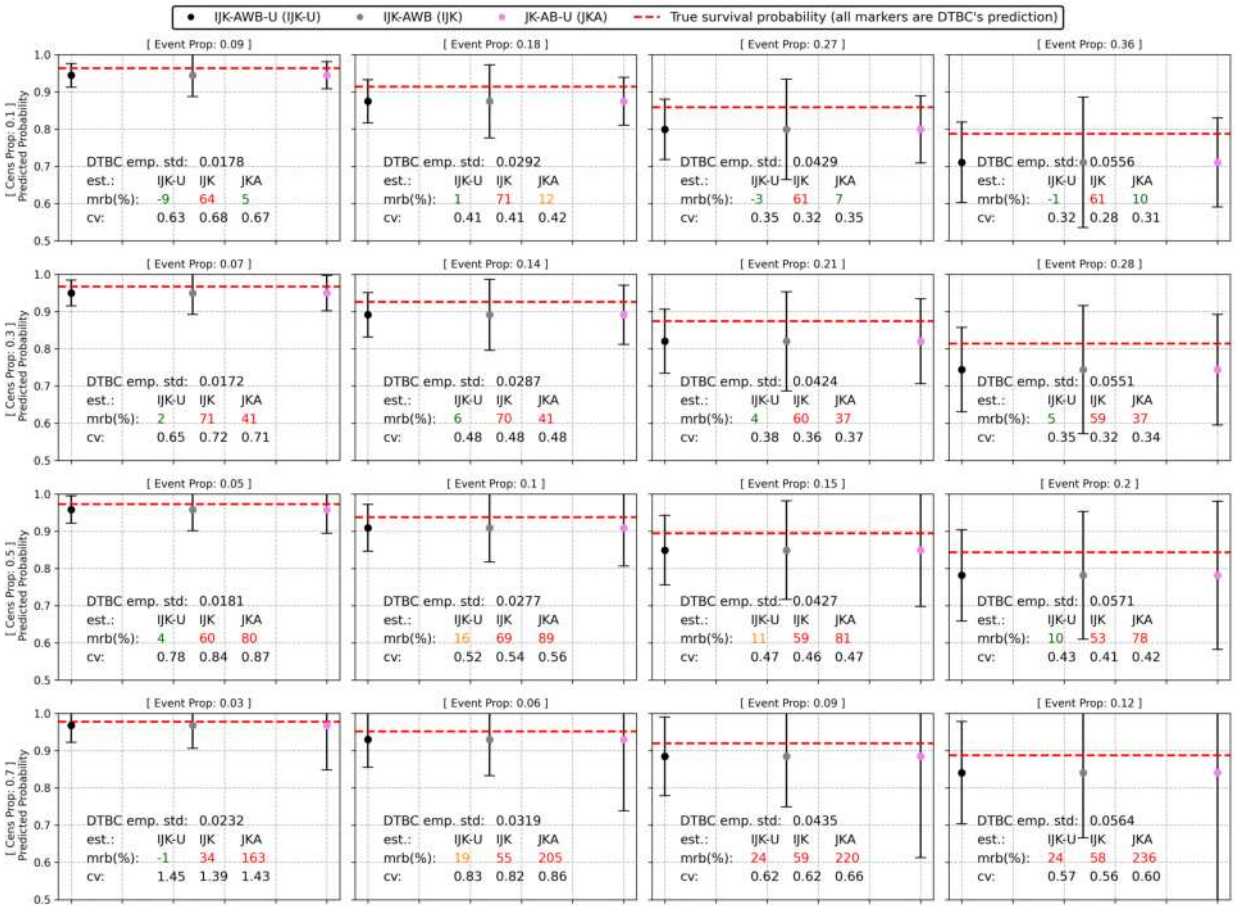
(b) Sim-Results for: $k = 1.5$, $n_{train} = 499$, $B = 4000$



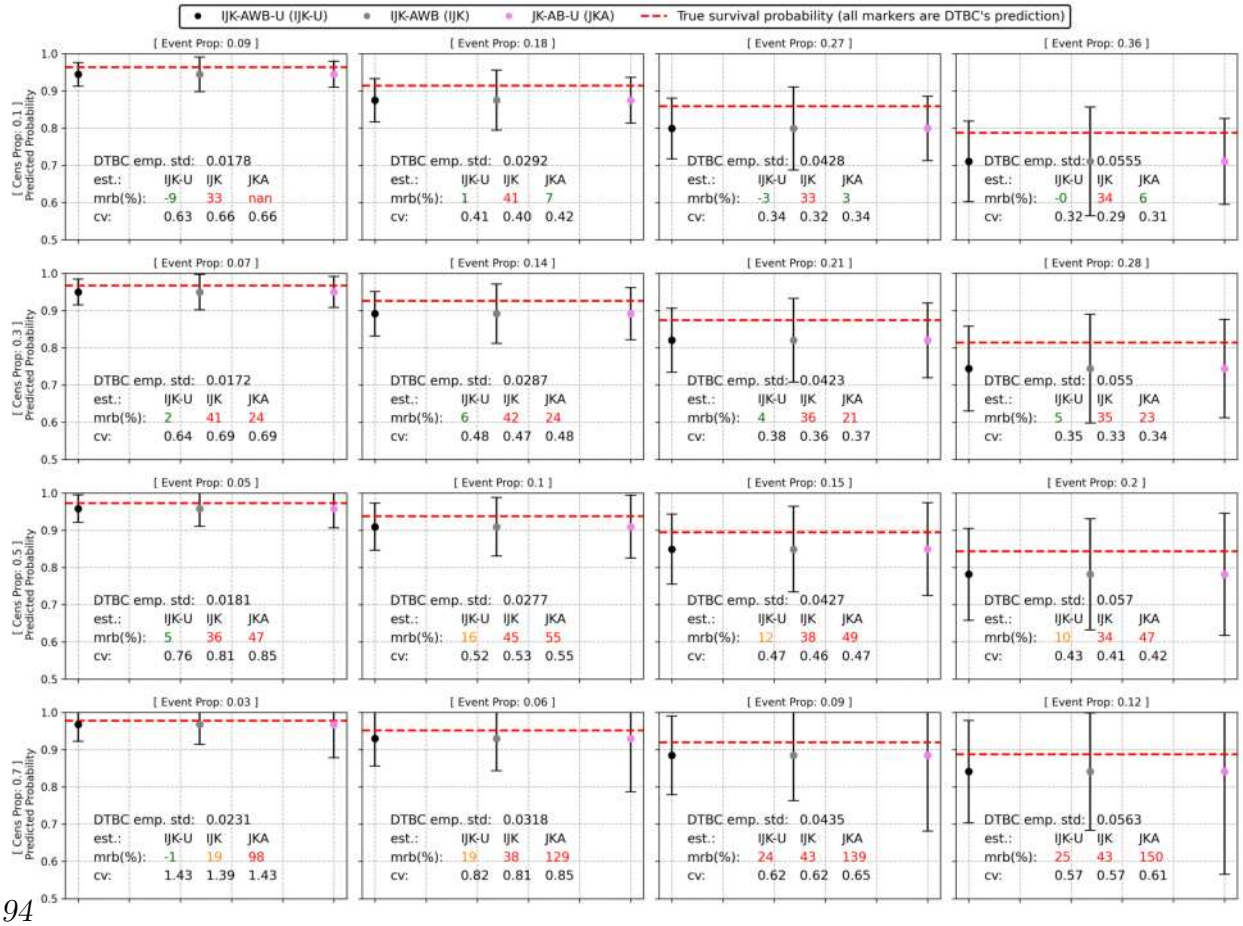
(a) Sim Results for: $k = 1.5$, $n_{train} = 999$, $B = 500$



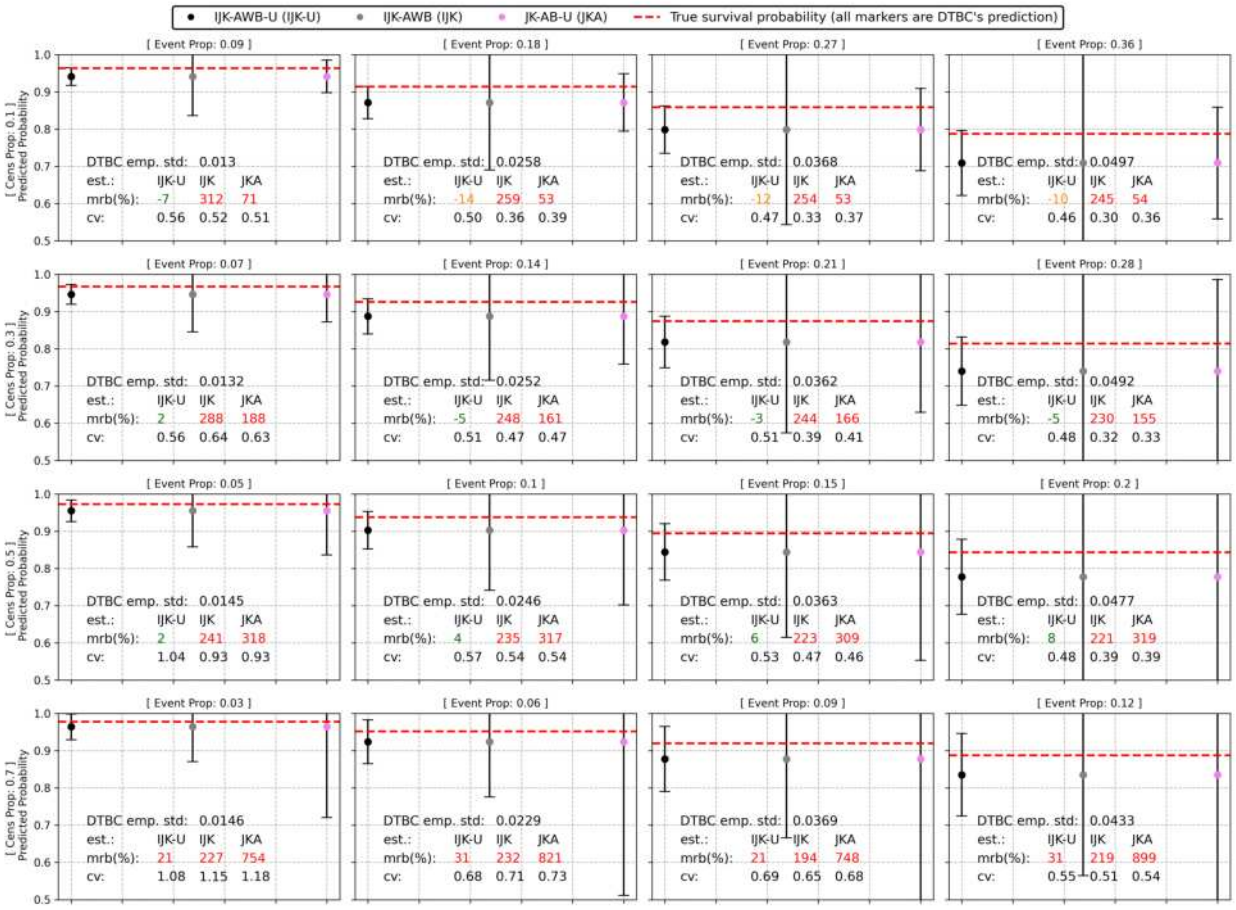
(b) Sim-Results for: $k = 1.5$, $n_{train} = 999$, $B = 1000$



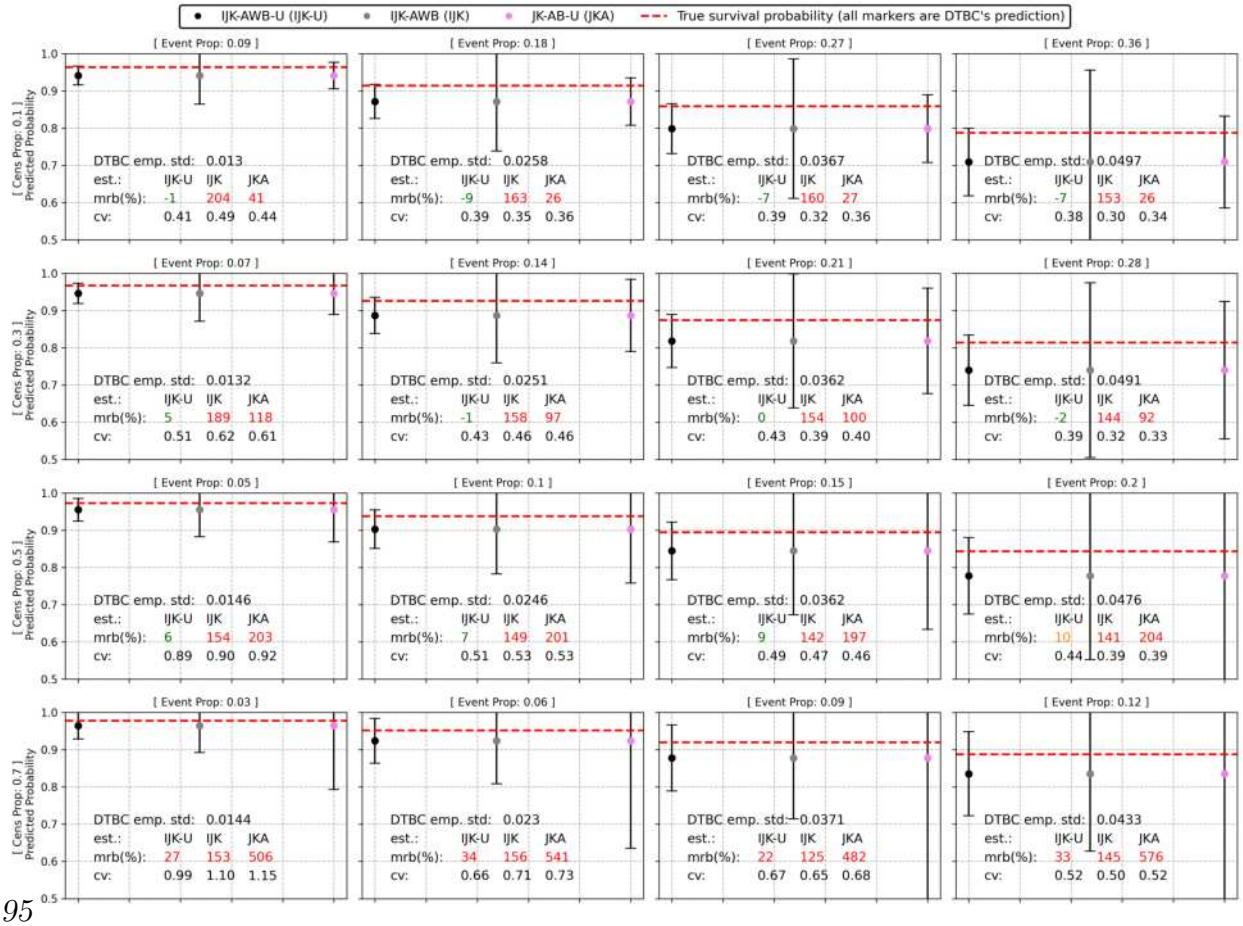
(a) Sim Results for: $k = 1.5$, $n_{train} = 999$, $B = 2000$



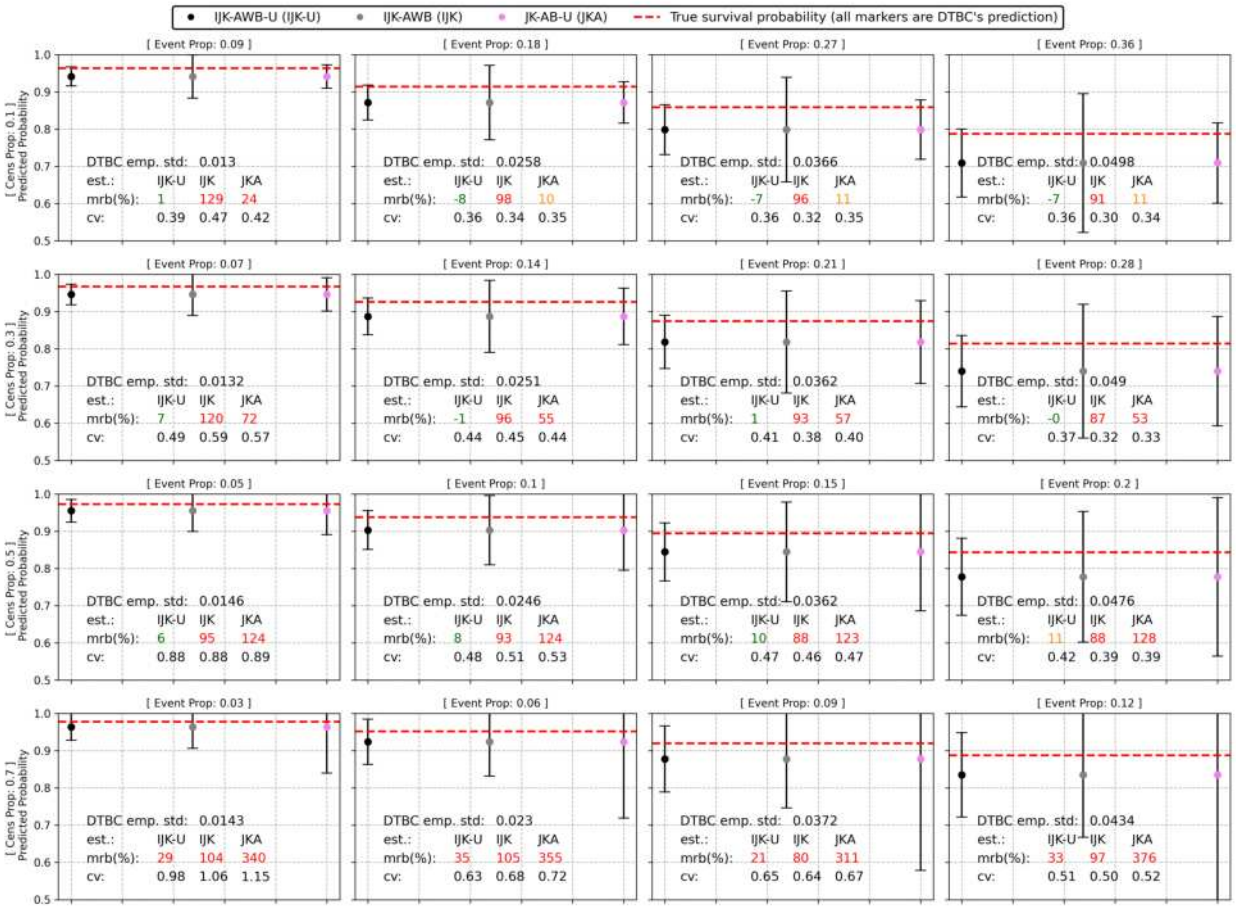
(b) Sim-Results for: $k = 1.5$, $n_{train} = 999$, $B = 4000$



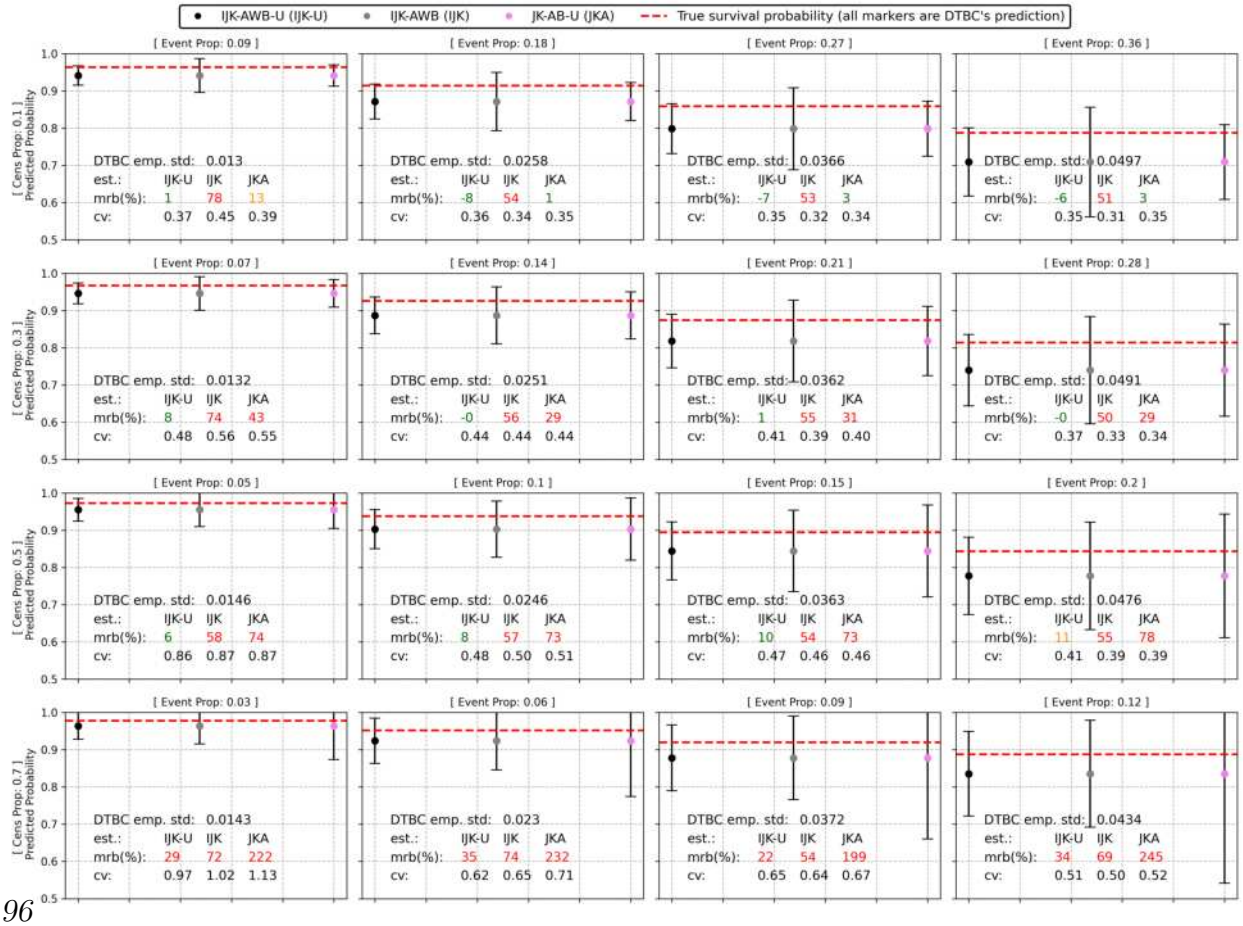
(a) Sim Results for: $k = 1.5$, $n_{train} = 1999$, $B = 500$



(b) Sim-Results for: $k = 1.5$, $n_{train} = 1999$, $B = 1000$



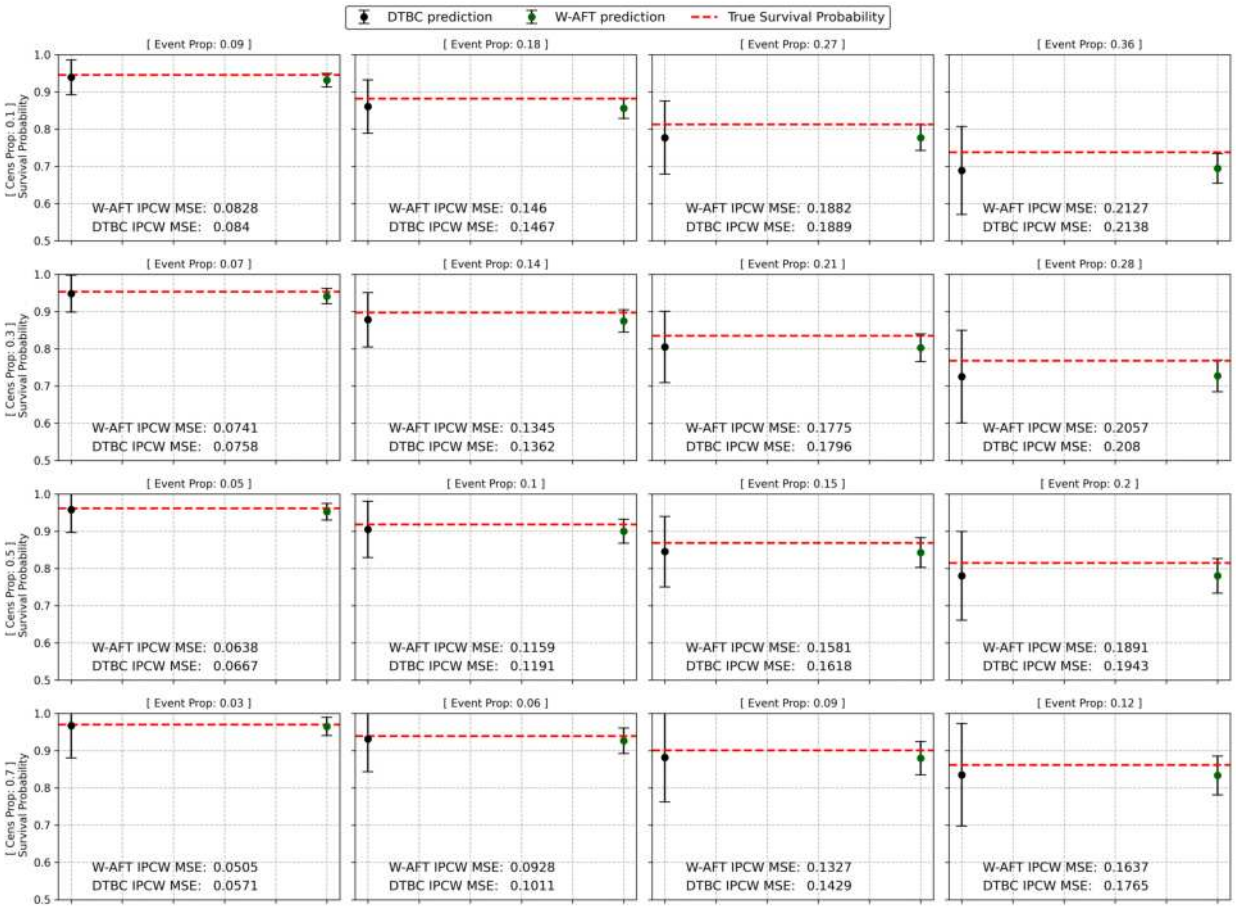
(a) Sim Results for: $k = 1.5$, $n_{train} = 1999$, $B = 2000$



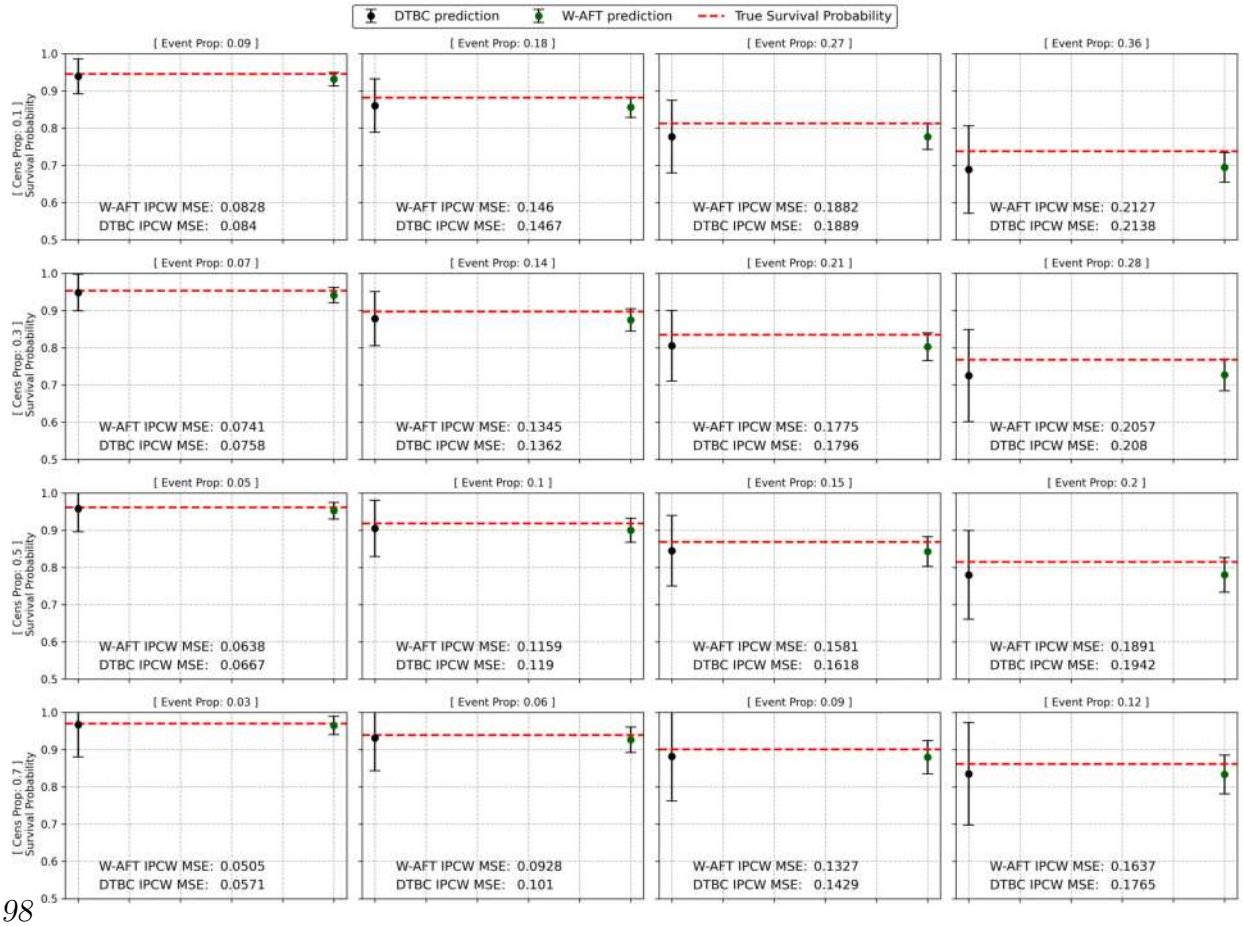
(b) Sim-Results for: $k = 1.5$, $n_{train} = 1999$, $B = 4000$

6.0.5 Simulation Study Results for Model's Performance ($k = 1$)

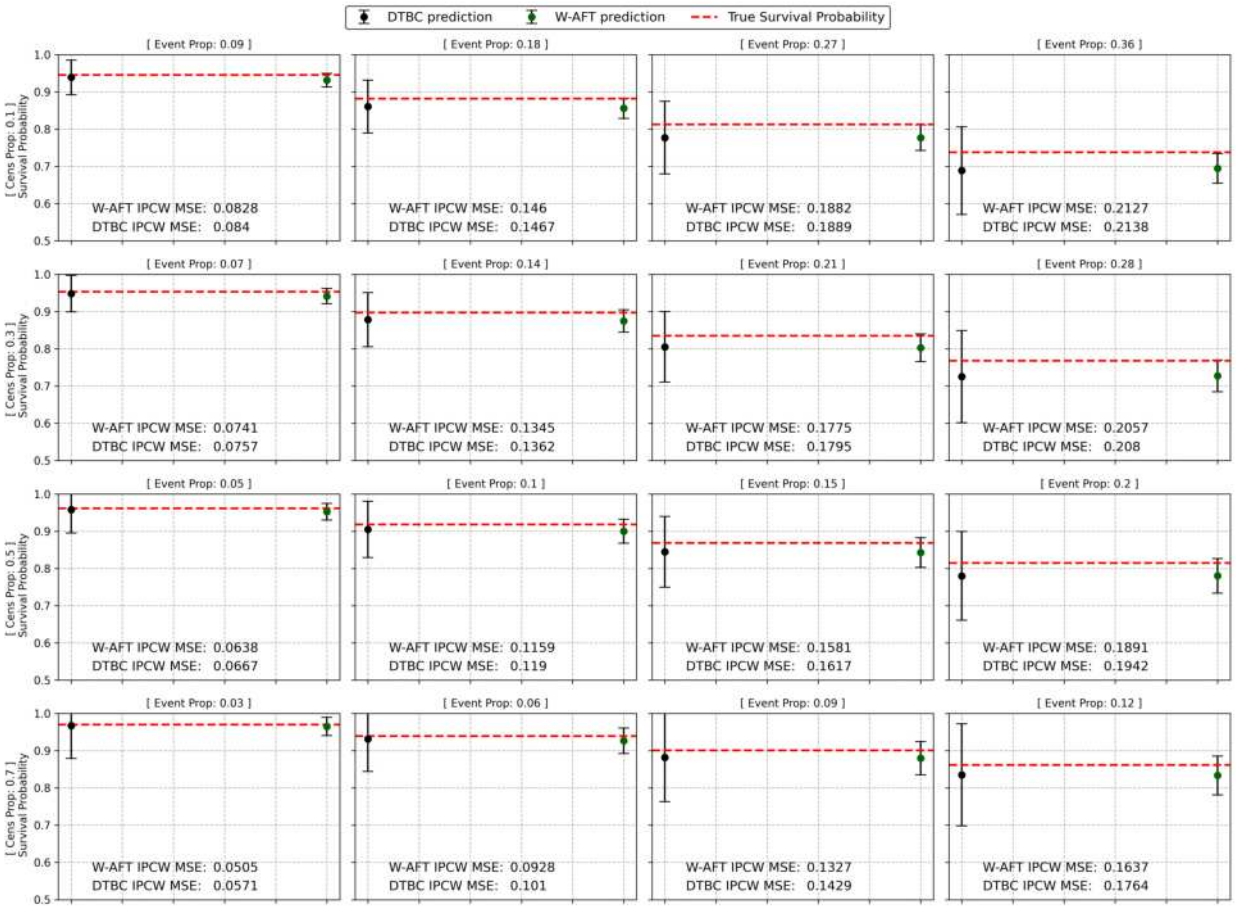
The explanations of these figures are provided in Figure 6.13.



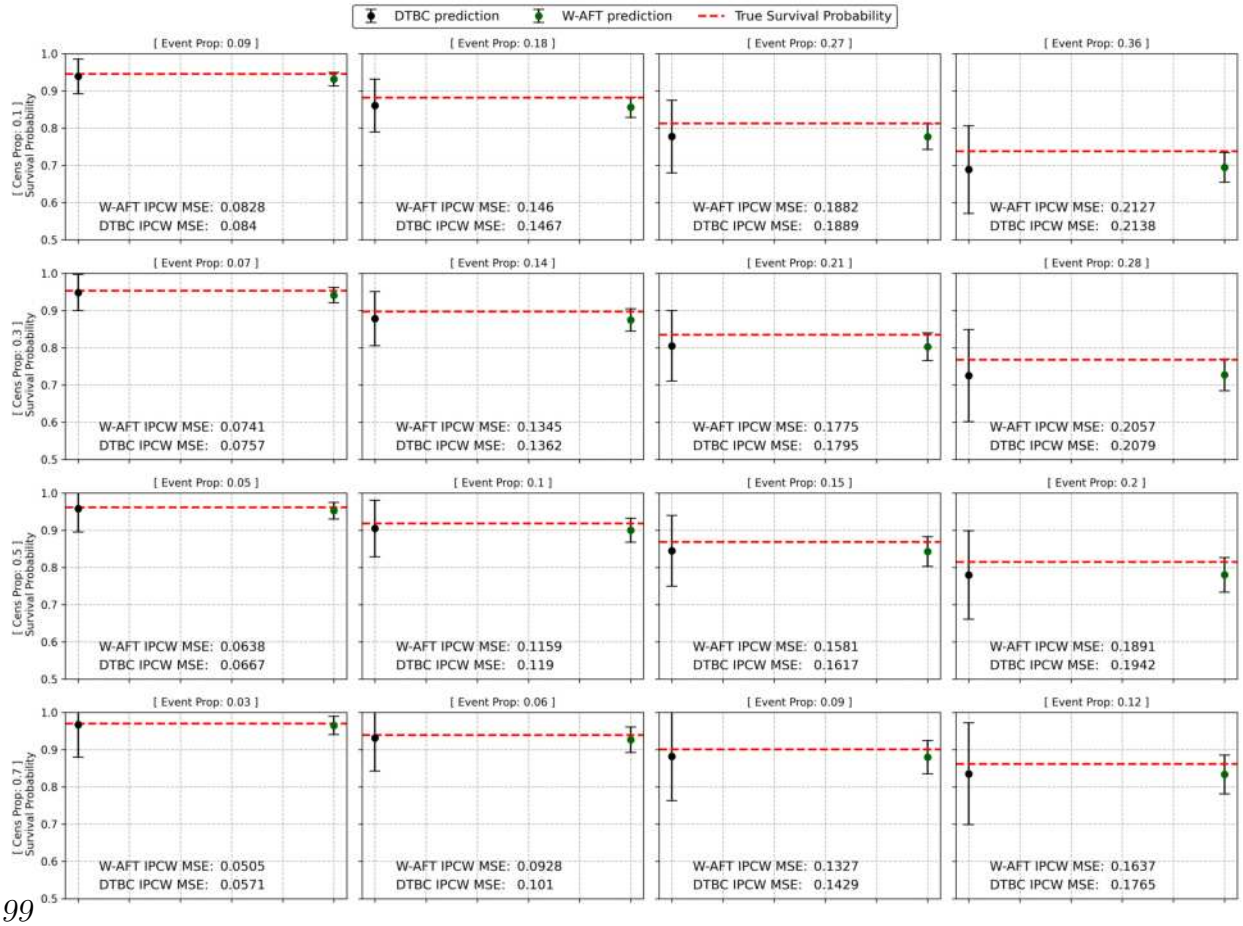
(a) Sim Results for: $k = 1$, $n_{train} = 499$, $B = 500$



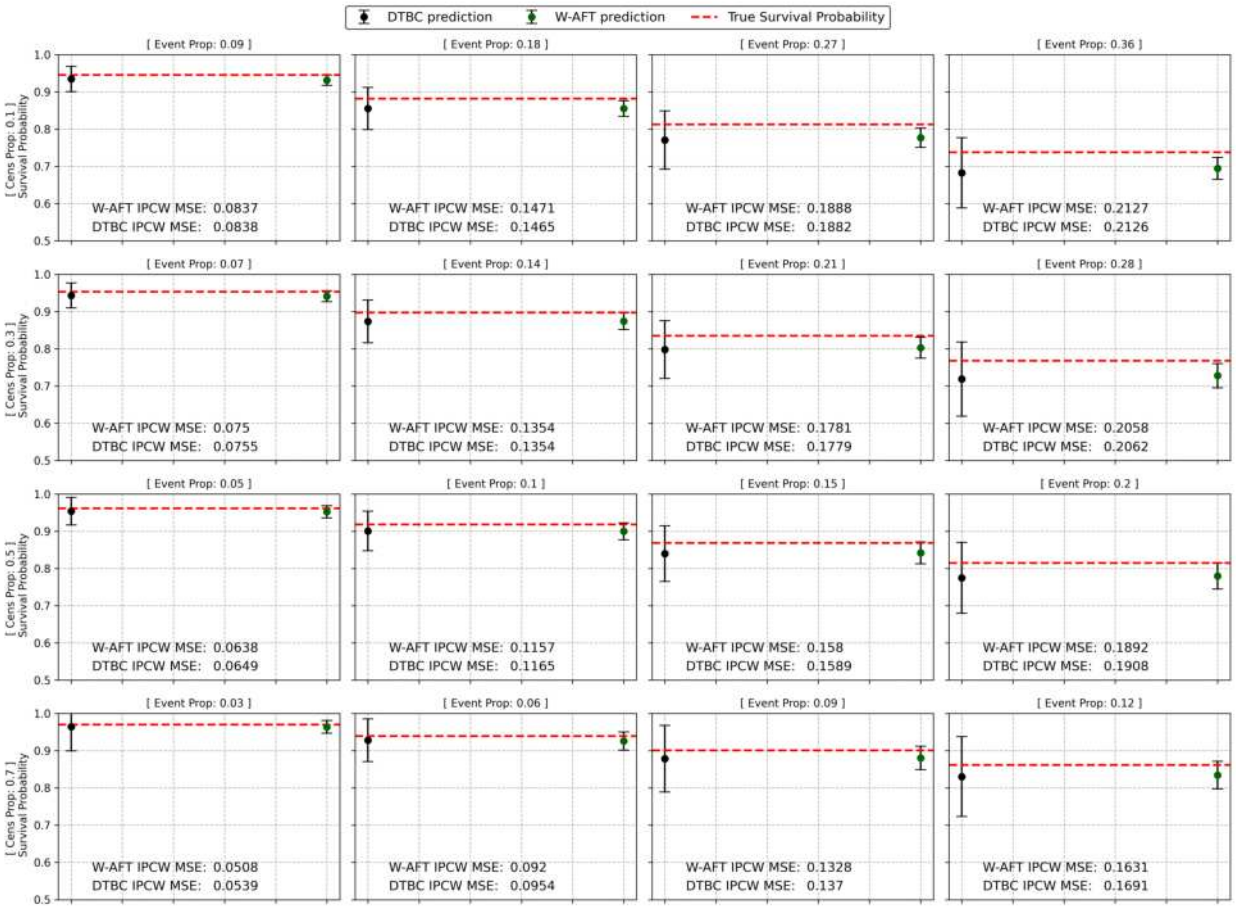
(b) Sim-Results for: $k = 1$, $n_{train} = 499$, $B = 1000$



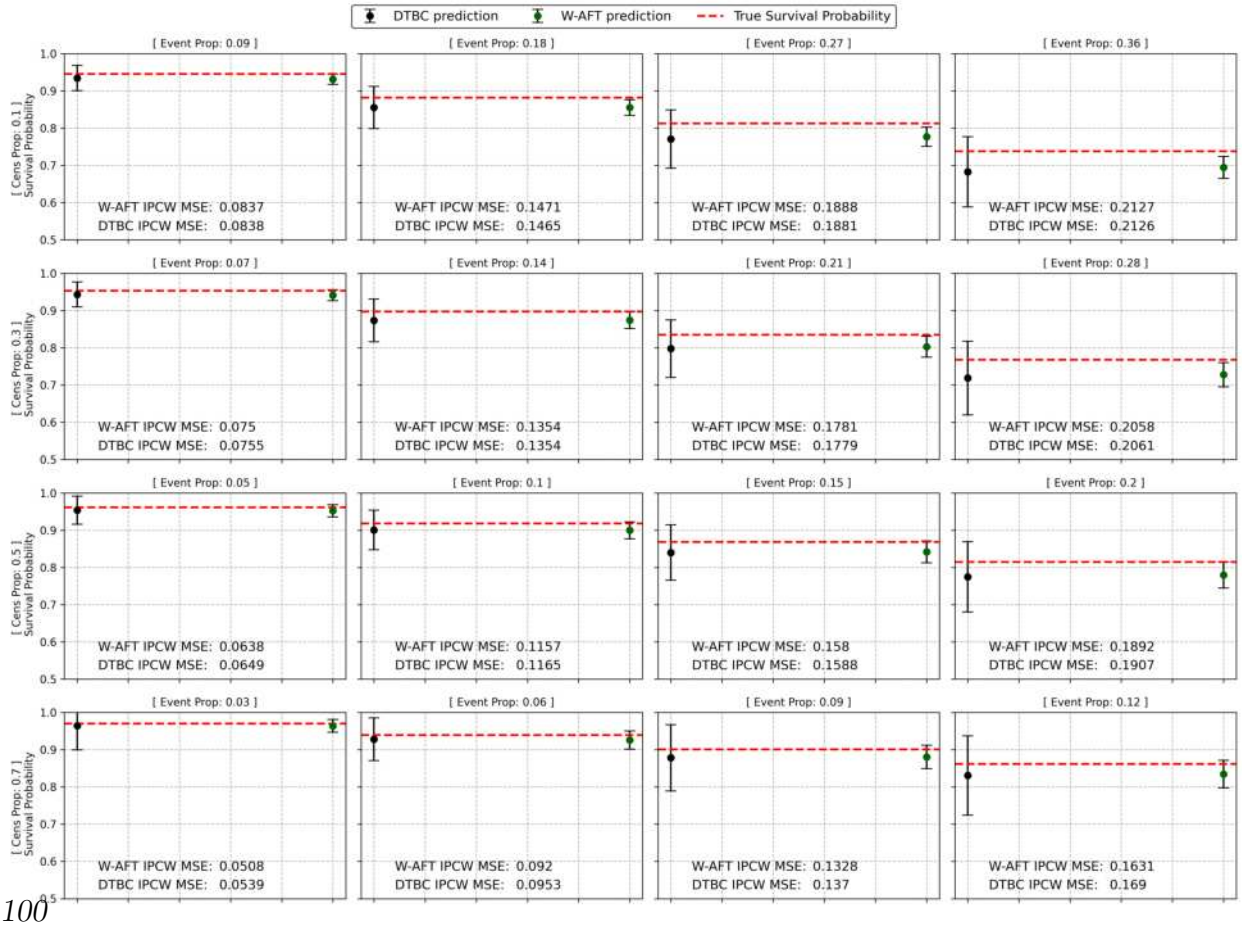
(a) Sim Results for: $k = 1$, $n_{train} = 499$, $B = 2000$



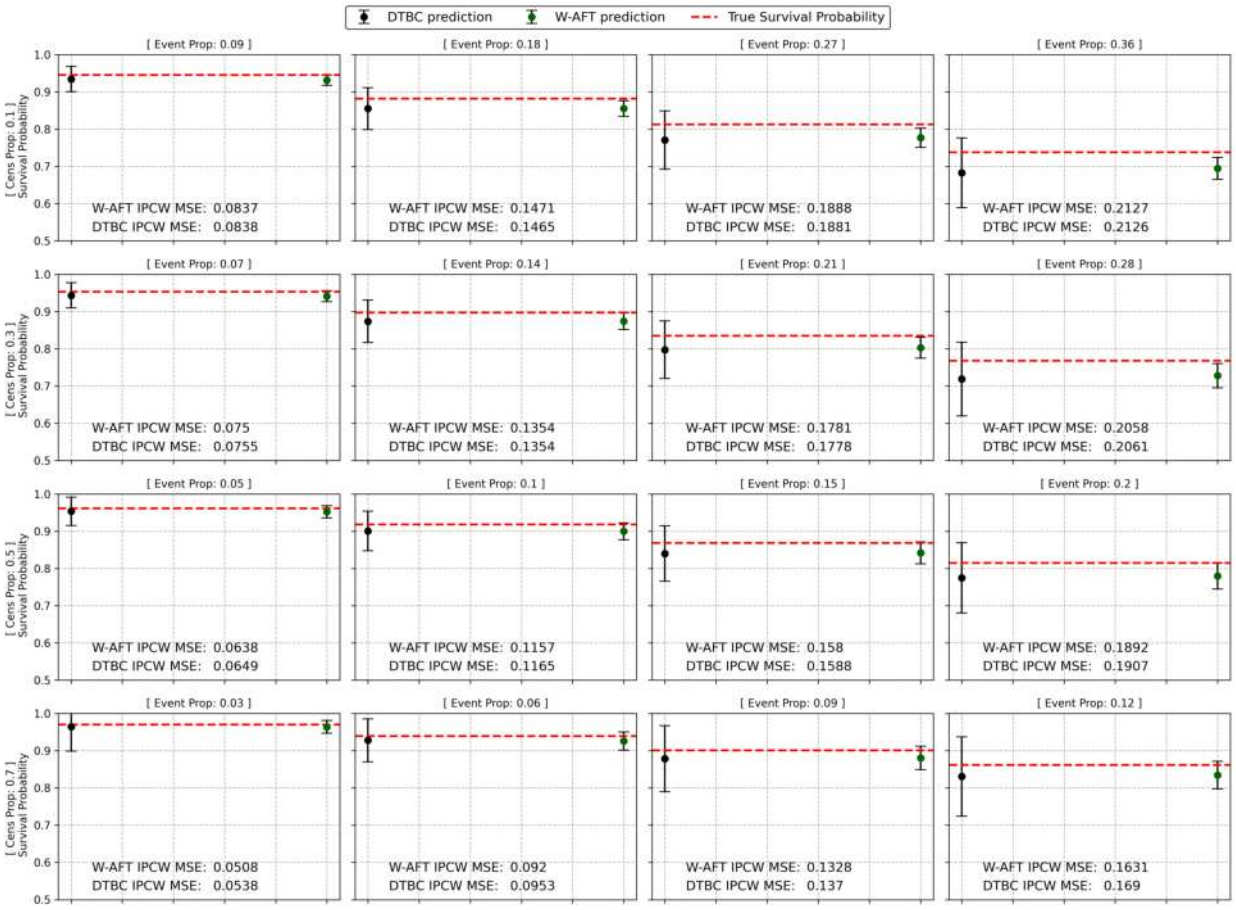
(b) Sim-Results for: $k = 1$, $n_{train} = 499$, $B = 4000$



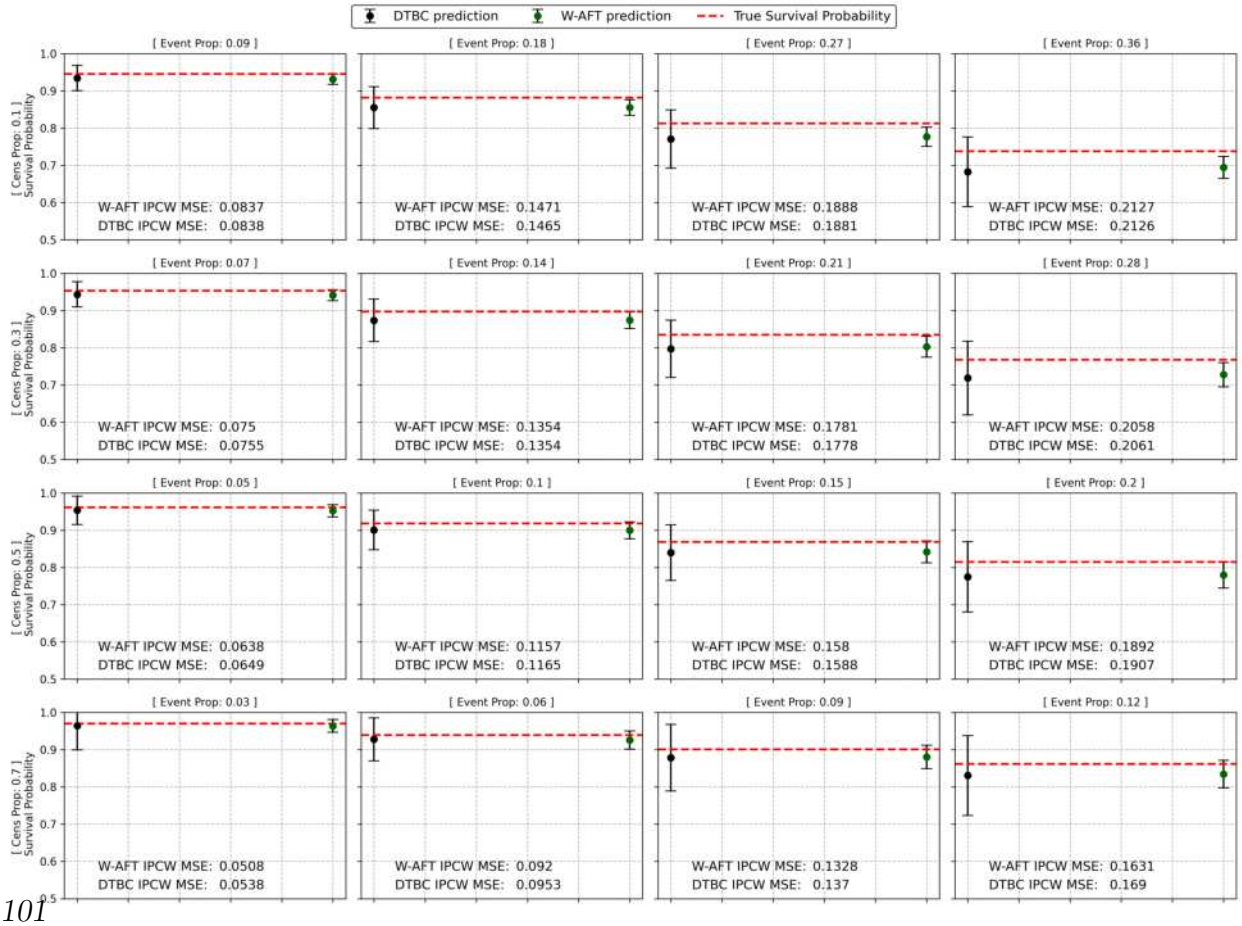
(a) Sim Results for: $k = 1$, $n_{train} = 999$, $B = 500$



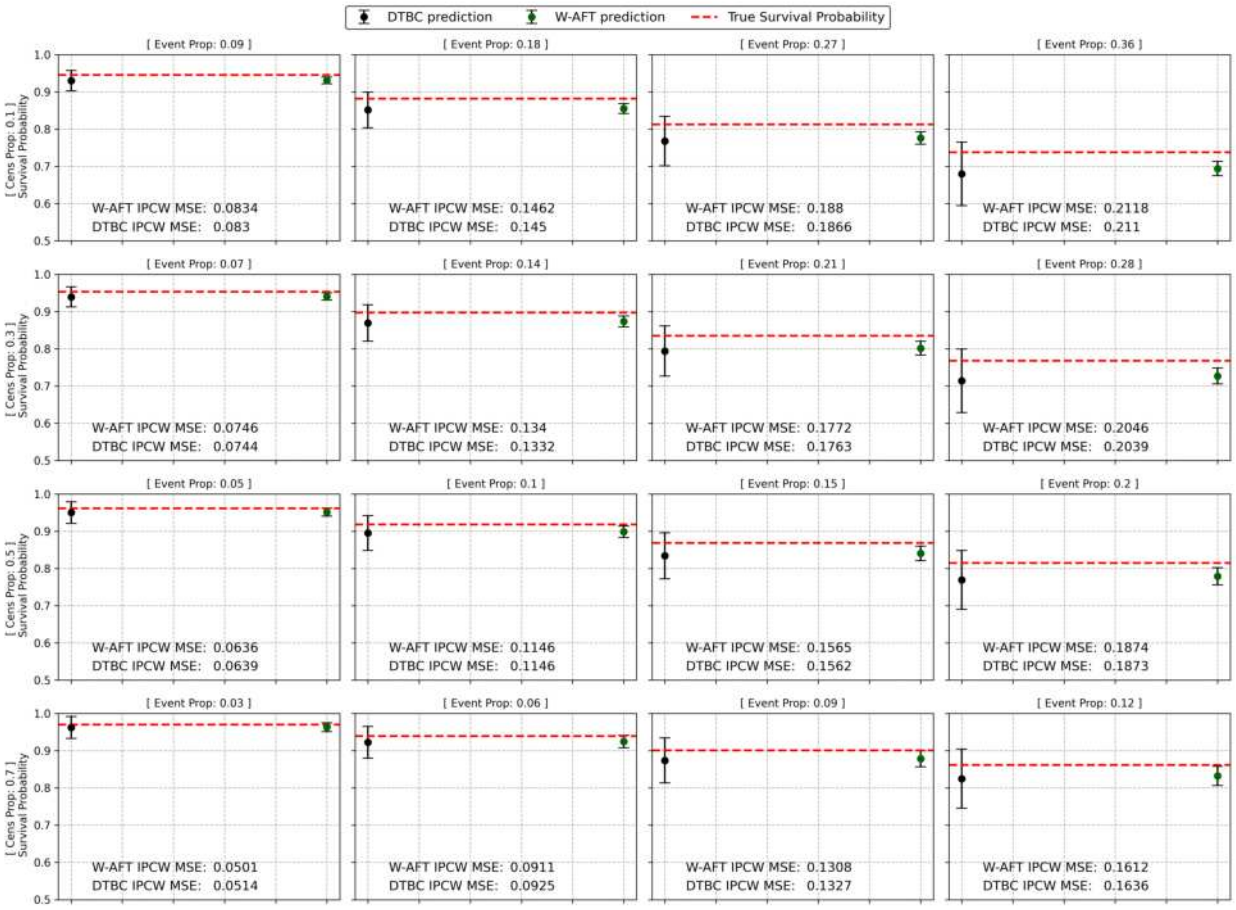
(b) Sim-Results for: $k = 1$, $n_{train} = 999$, $B = 1000$



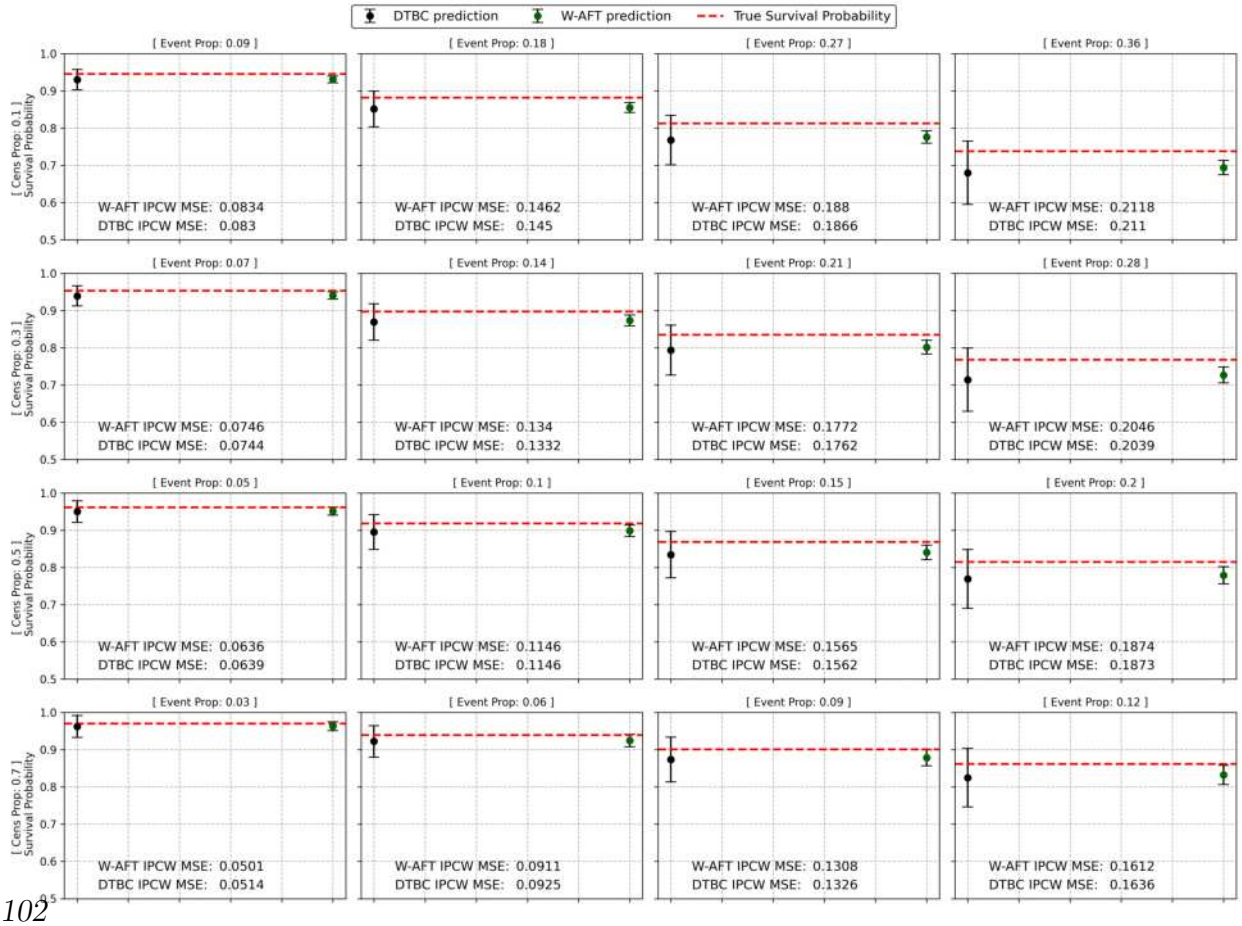
(a) Sim Results for: $k = 1$, $n_{train} = 999$, $B = 2000$



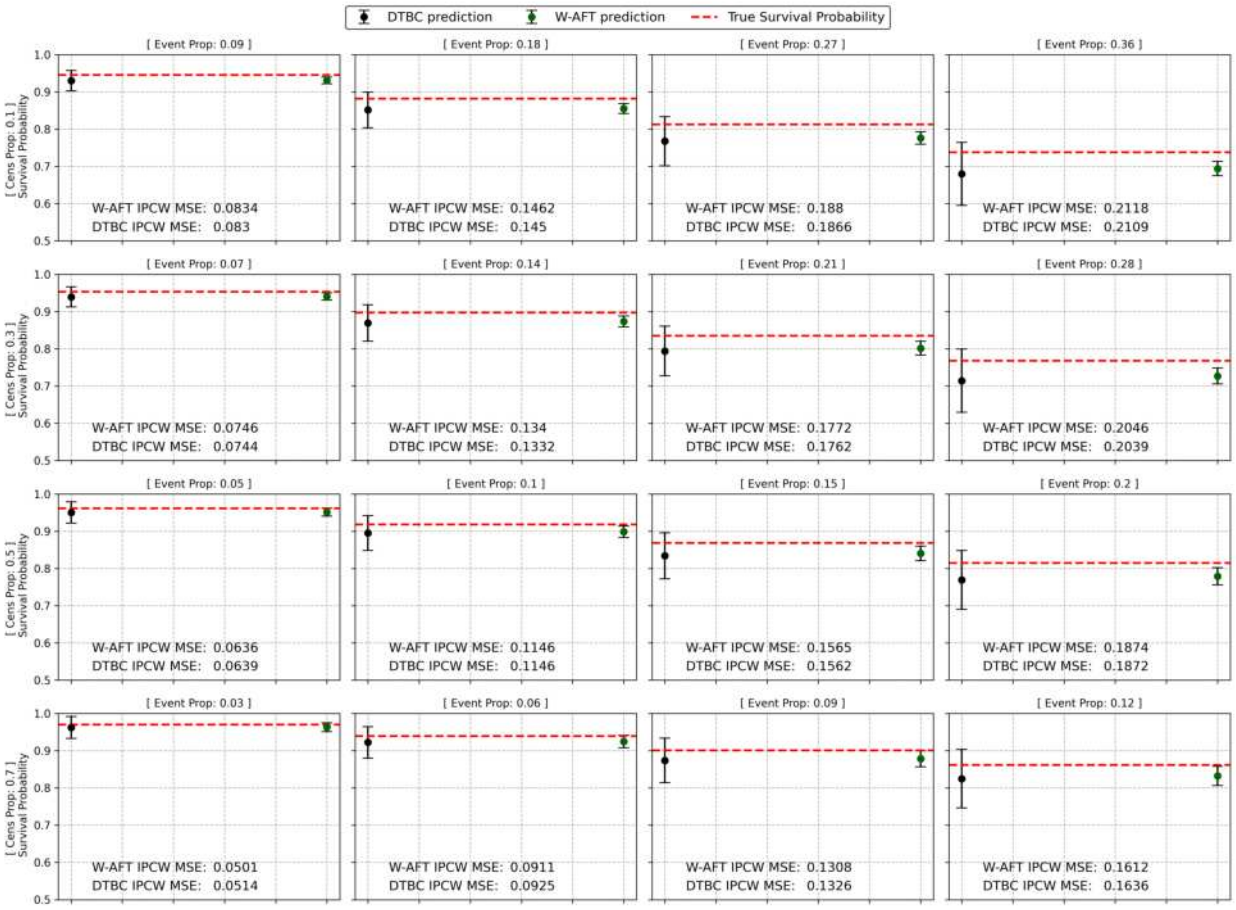
(b) Sim-Results for: $k = 1$, $n_{train} = 999$, $B = 4000$



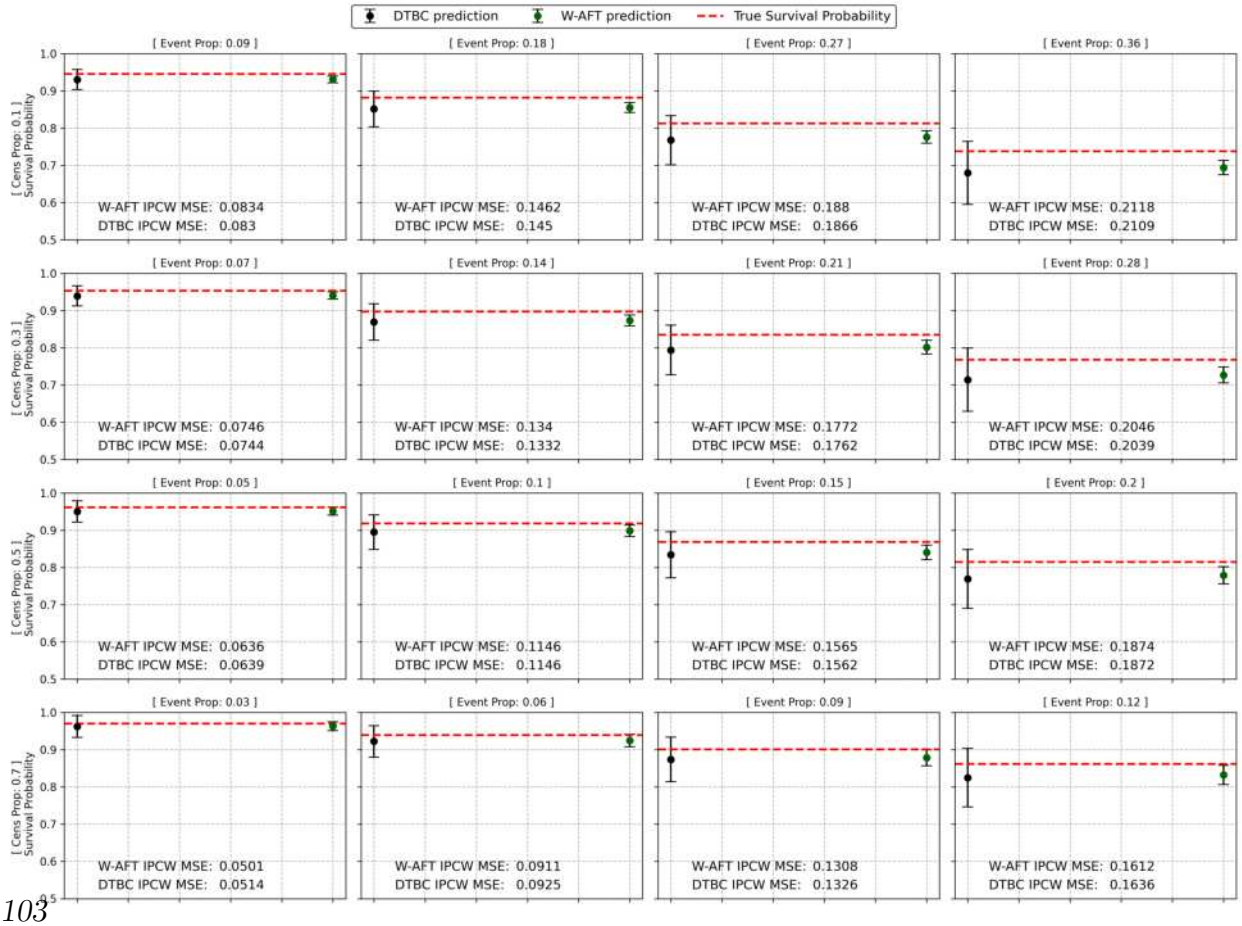
(a) Sim Results for: $k = 1$, $n_{train} = 1999$, $B = 500$



(b) Sim-Results for: $k = 1$, $n_{train} = 1999$, $B = 1000$



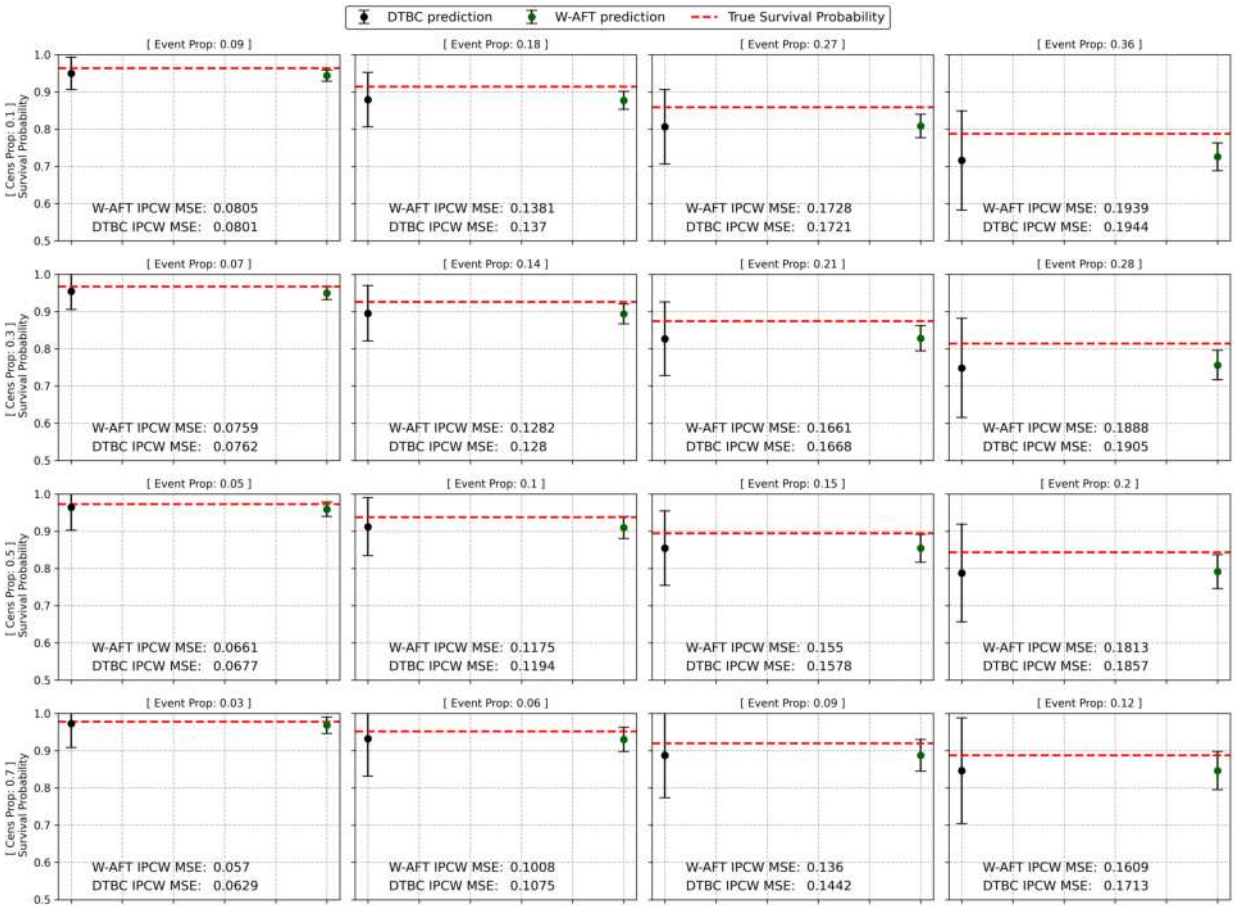
(a) Sim Results for: $k = 1$, $n_{train} = 1999$, $B = 2000$



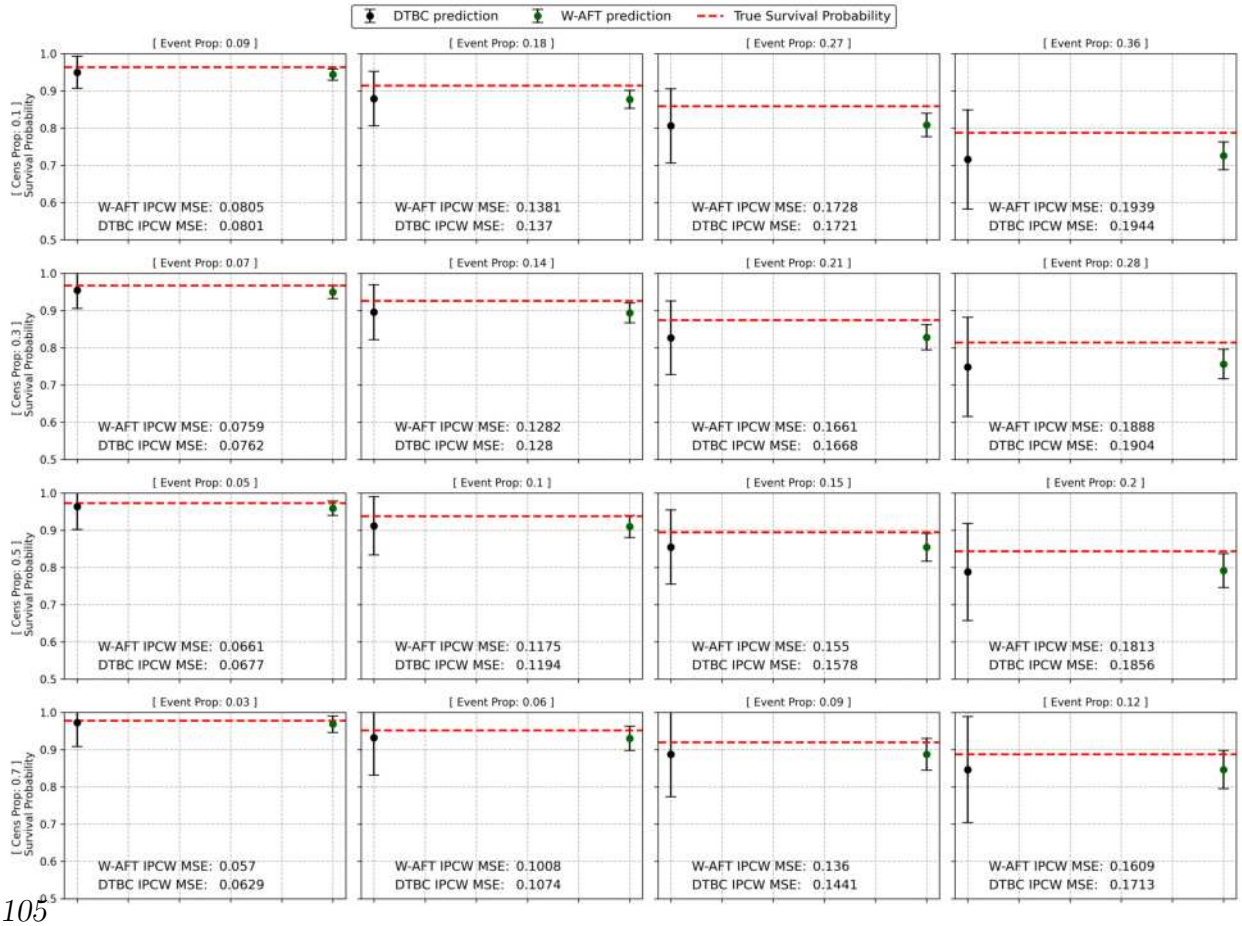
(b) Sim-Results for: $k = 1$, $n_{train} = 1999$, $B = 4000$

6.0.6 Simulation Study Results for Model's Performance ($k = 1.5$)

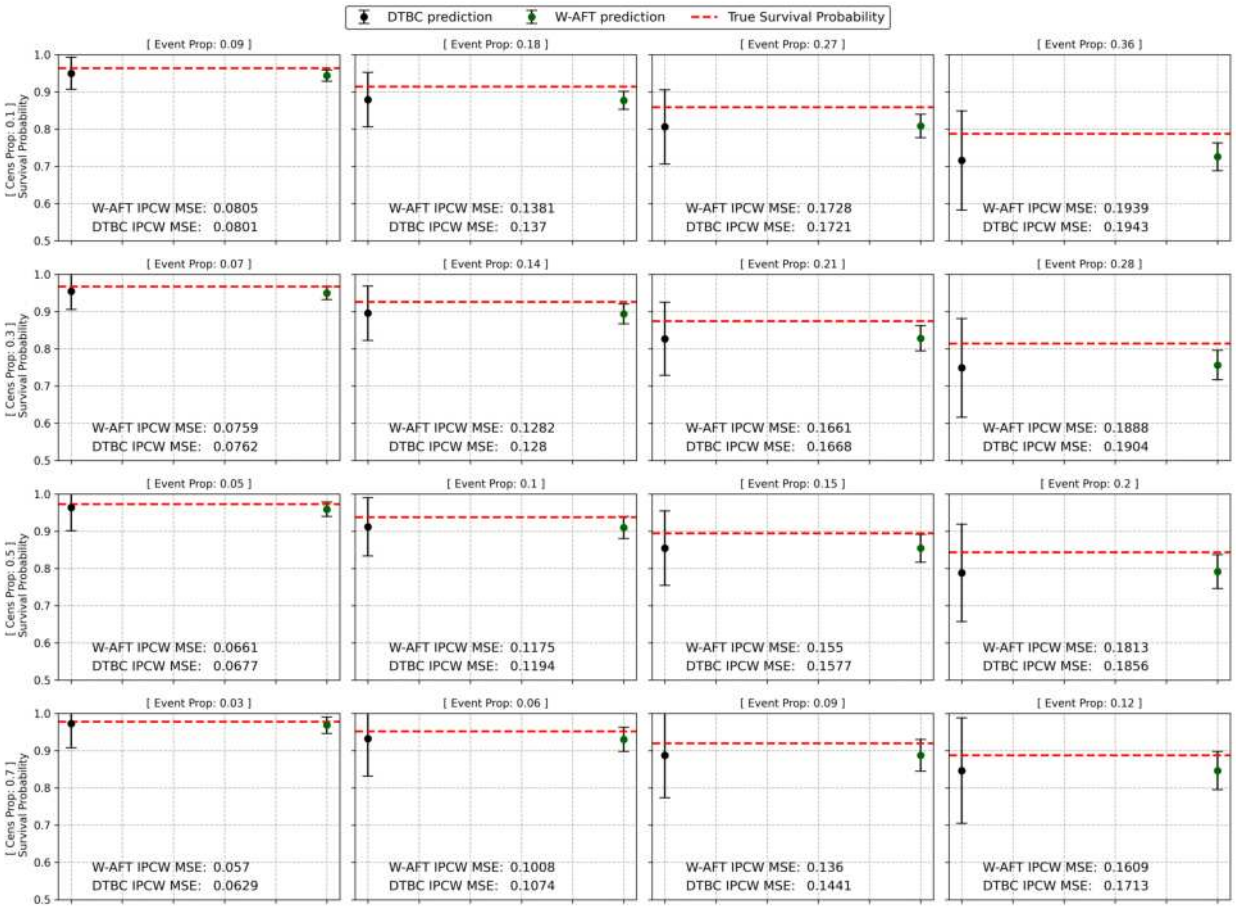
The explanations of these figures are provided in Figure 6.13.



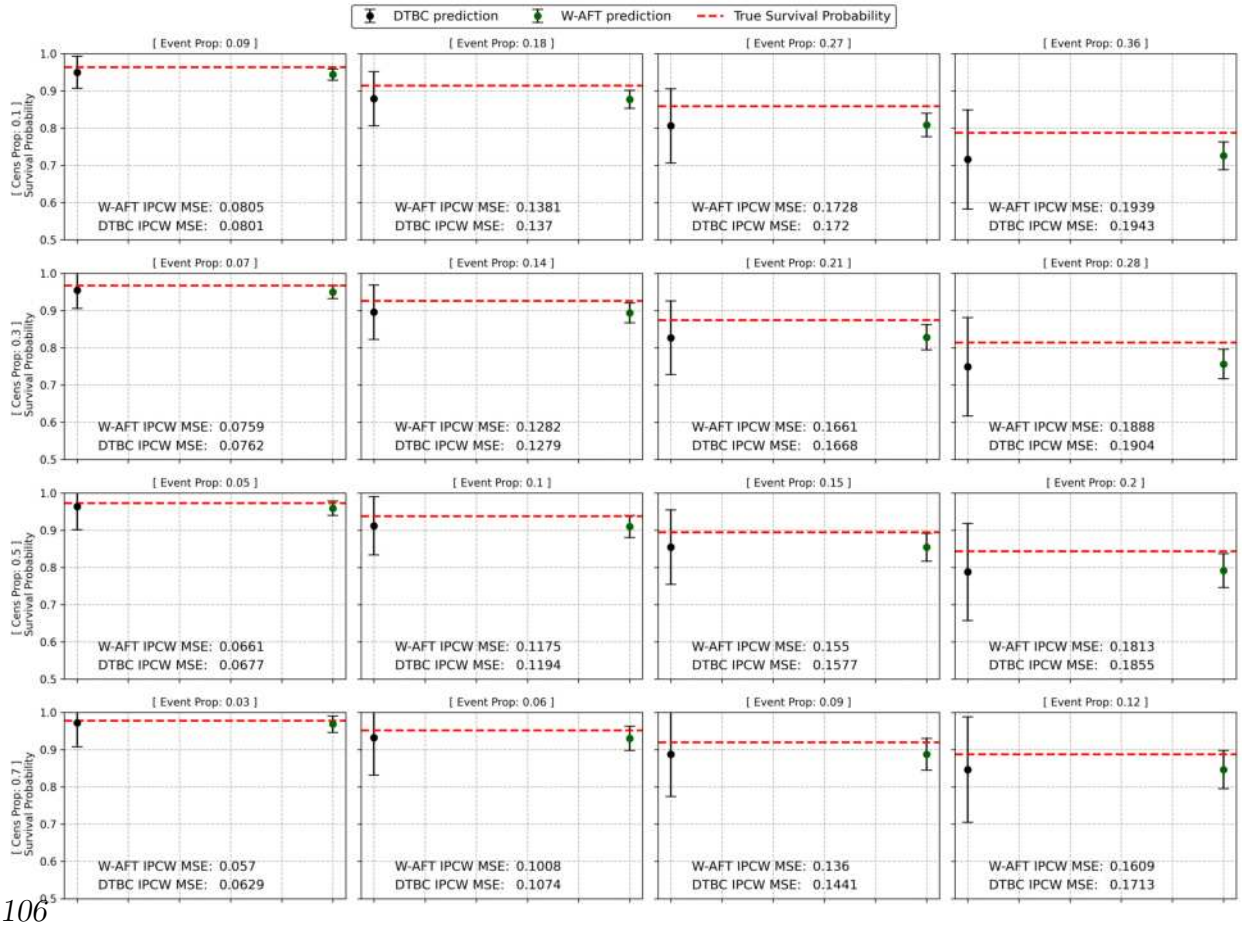
(a) Sim Results for: $k = 1.5$, $n_{train} = 499$, $B = 500$



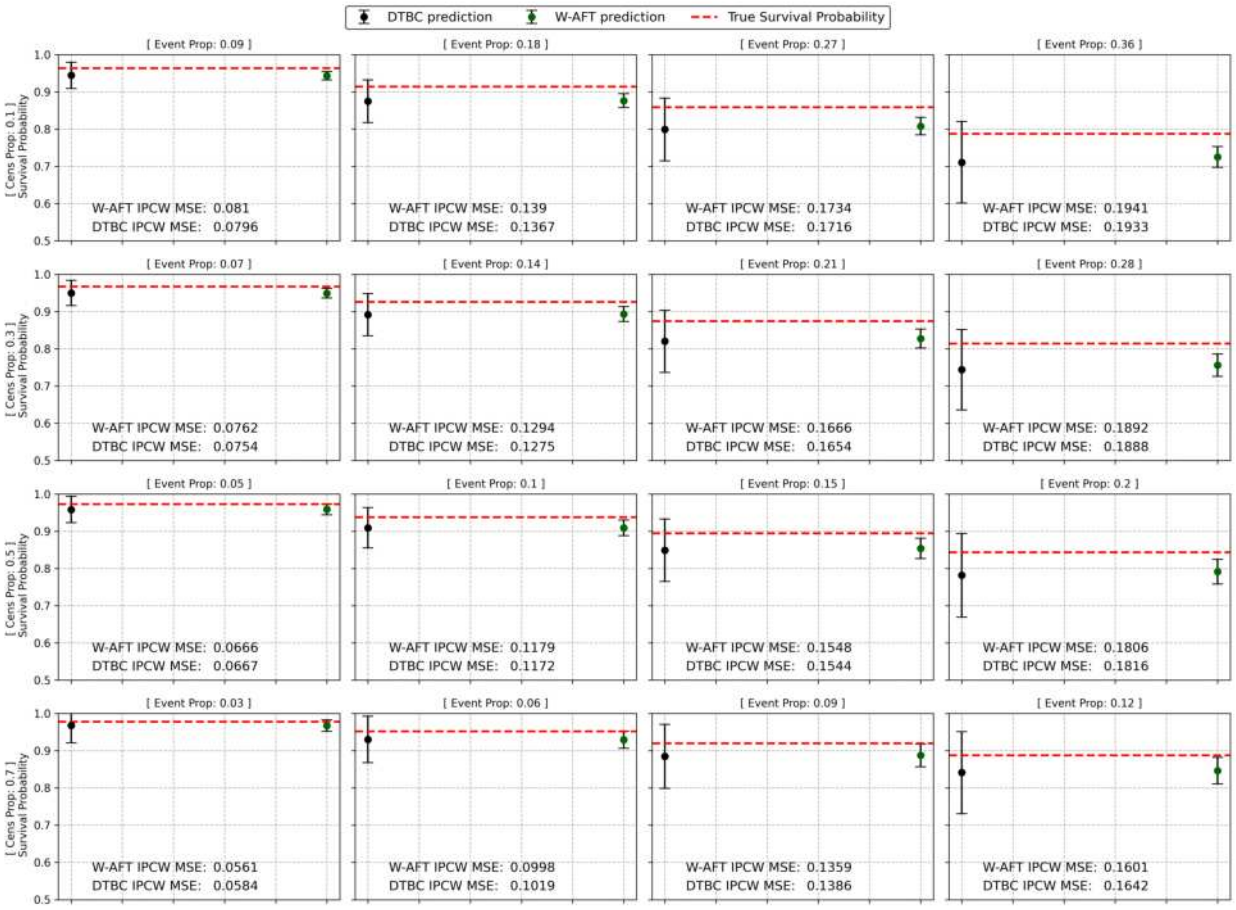
(b) Sim-Results for: $k = 1.5$, $n_{train} = 499$, $B = 1000$



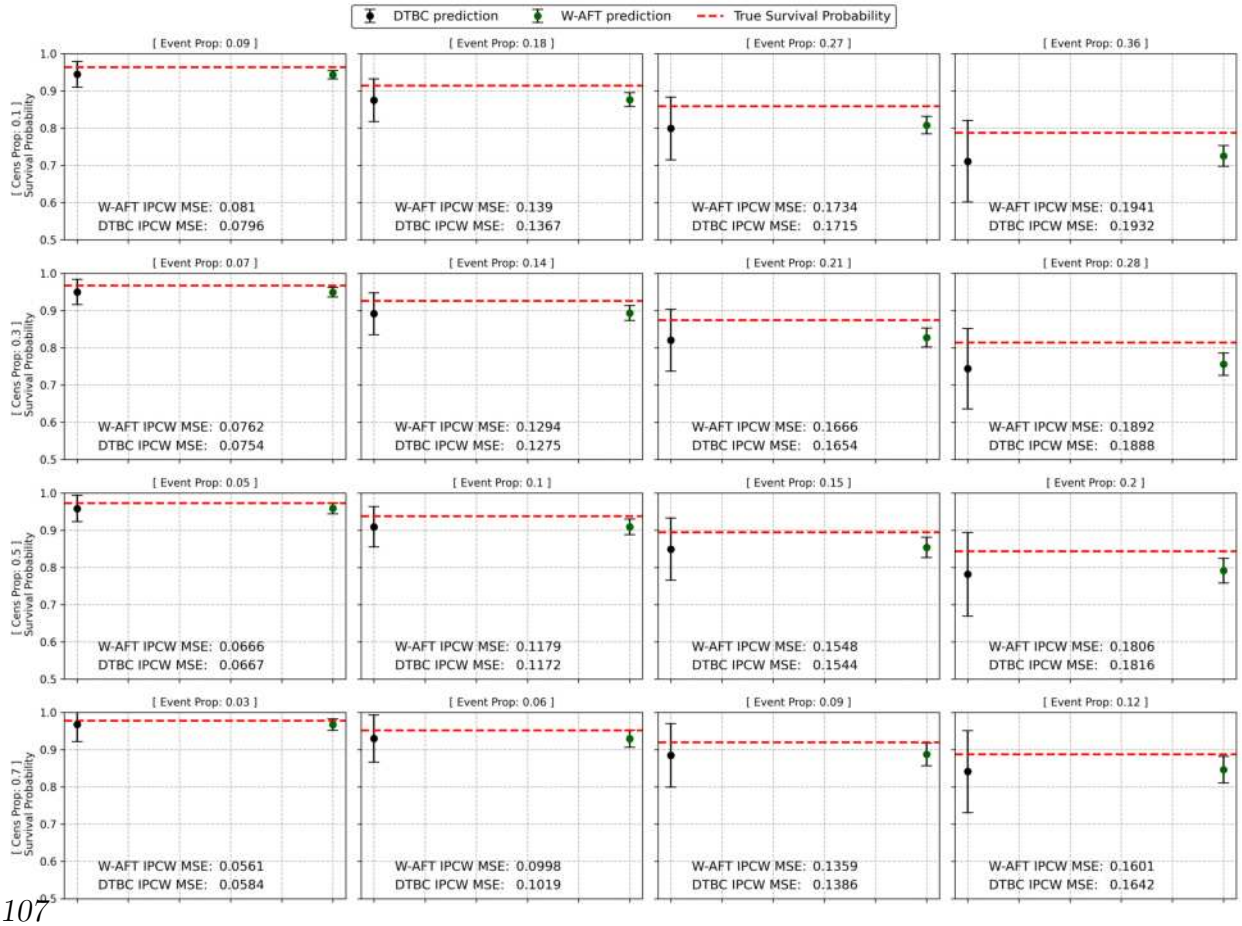
(a) Sim Results for: $k = 1.5$, $n_{train} = 499$, $B = 2000$



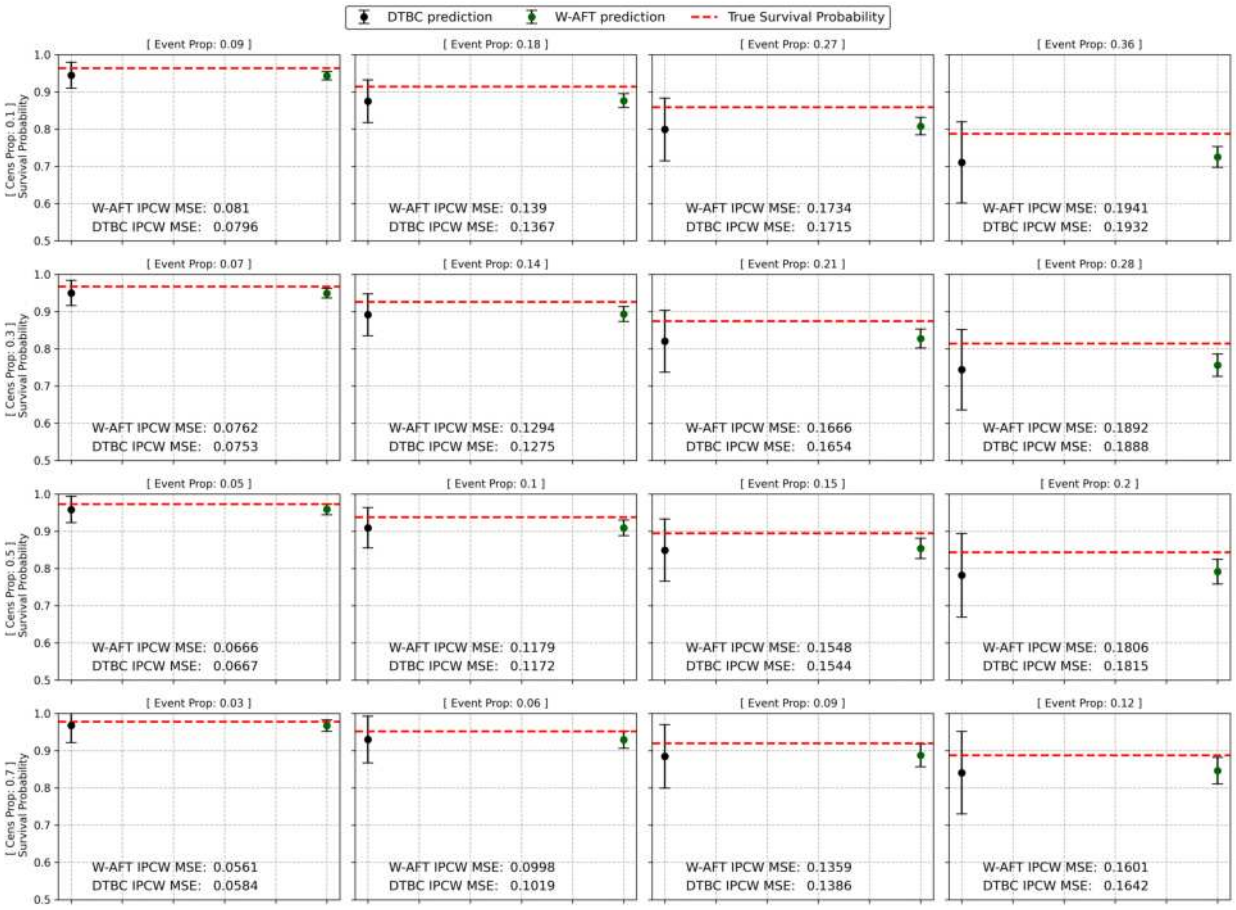
(b) Sim-Results for: $k = 1.5$, $n_{train} = 499$, $B = 4000$



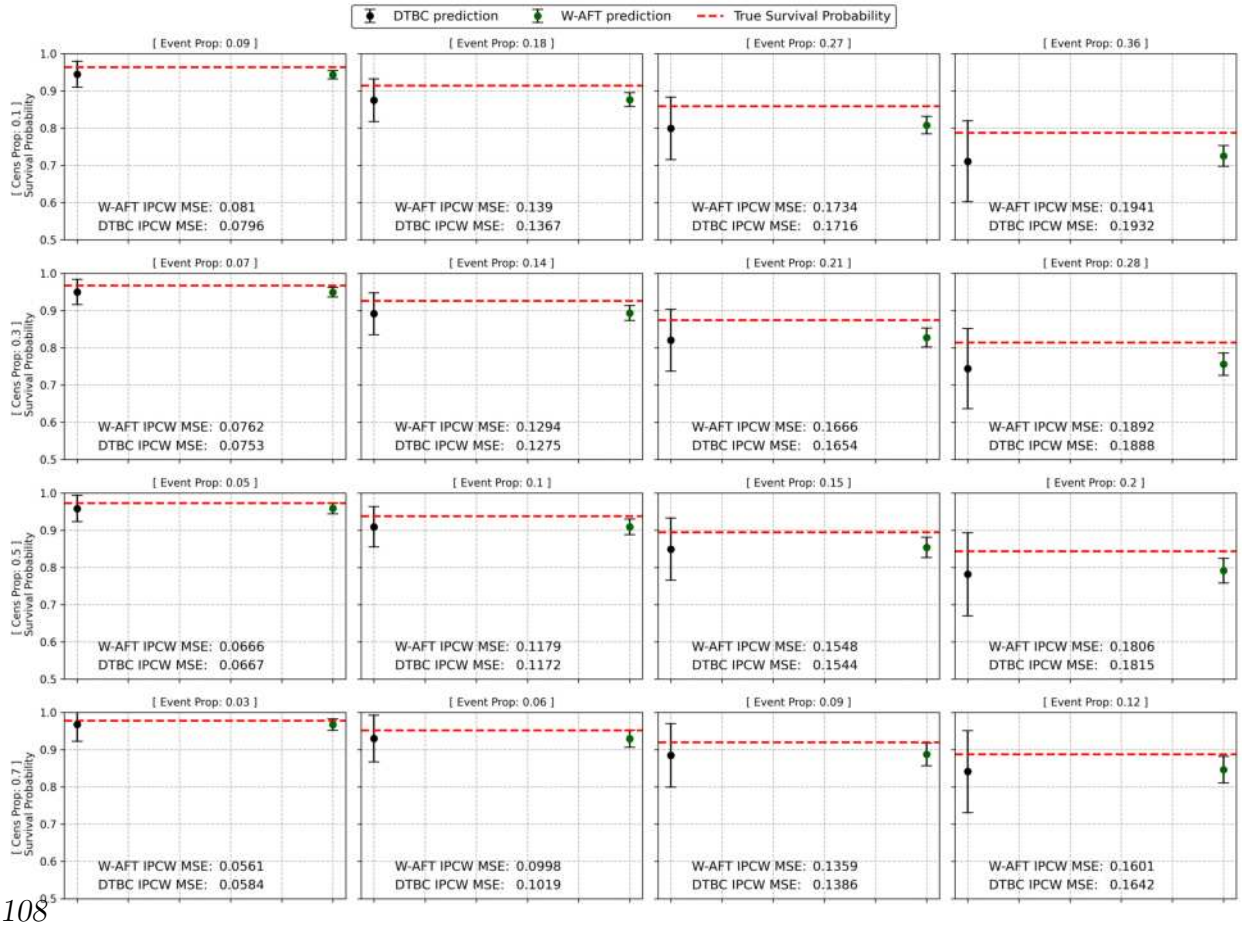
(a) Sim Results for: $k = 1.5$, $n_{train} = 999$, $B = 500$



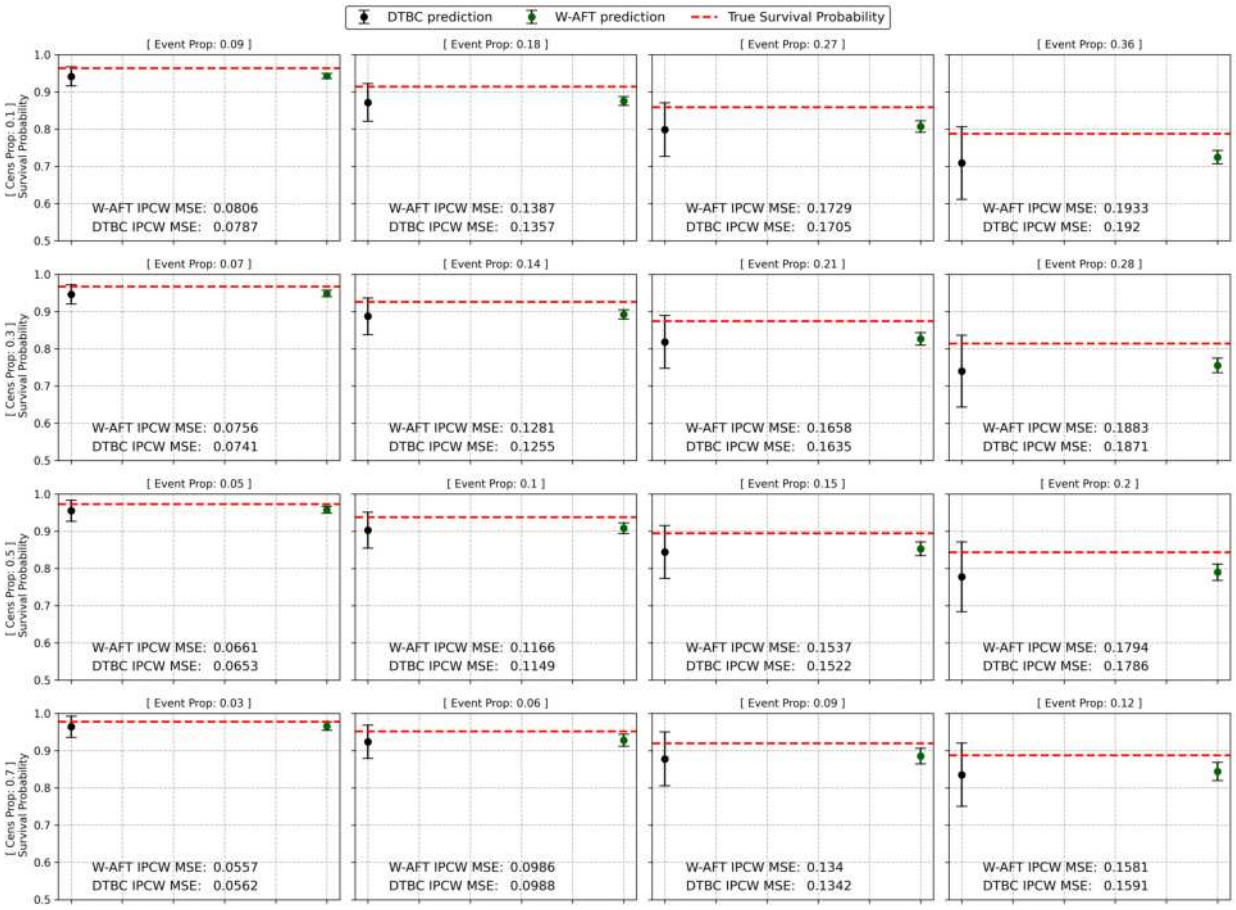
(b) Sim-Results for: $k = 1.5$, $n_{train} = 999$, $B = 1000$



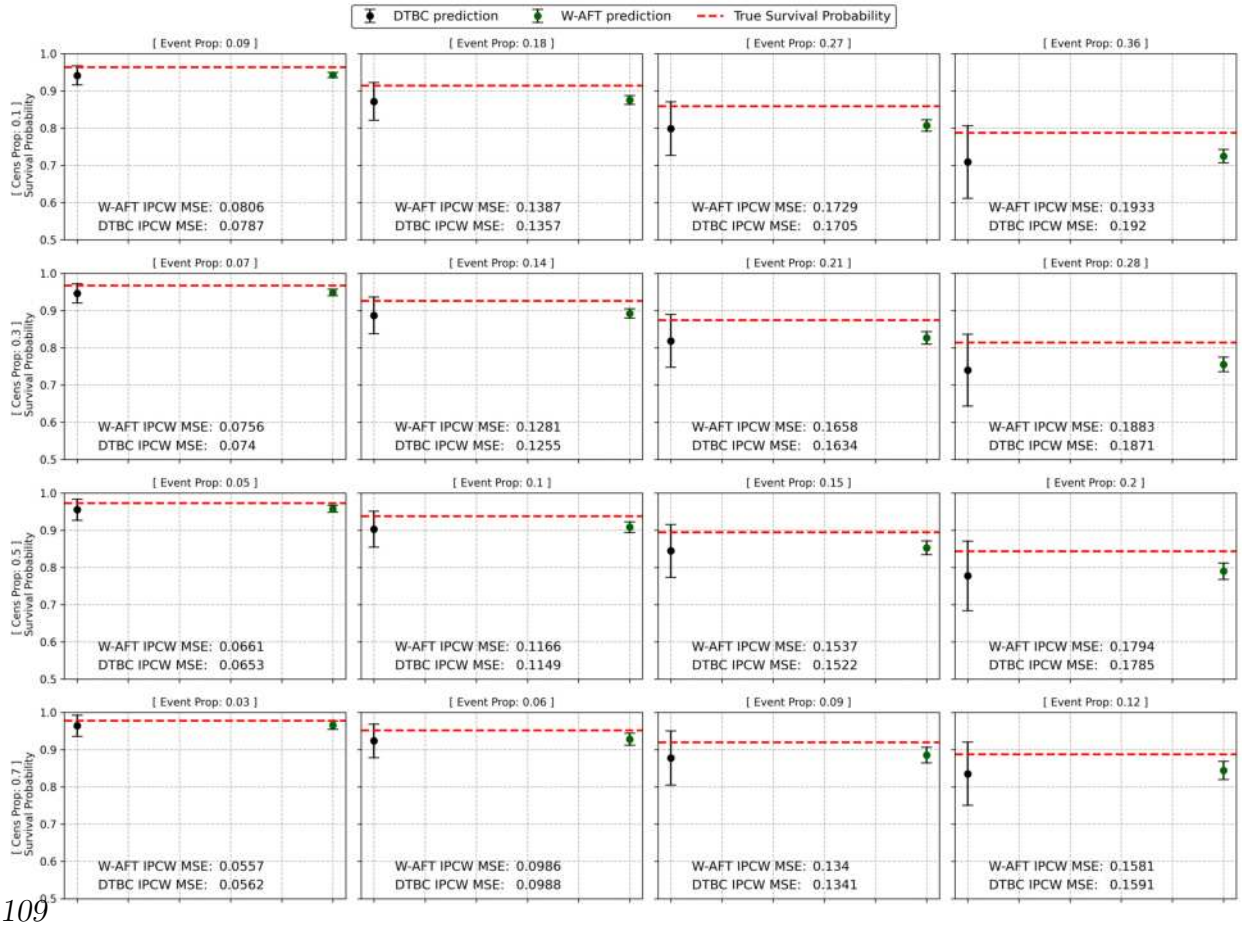
(a) Sim Results for: $k = 1.5$, $n_{train} = 999$, $B = 2000$



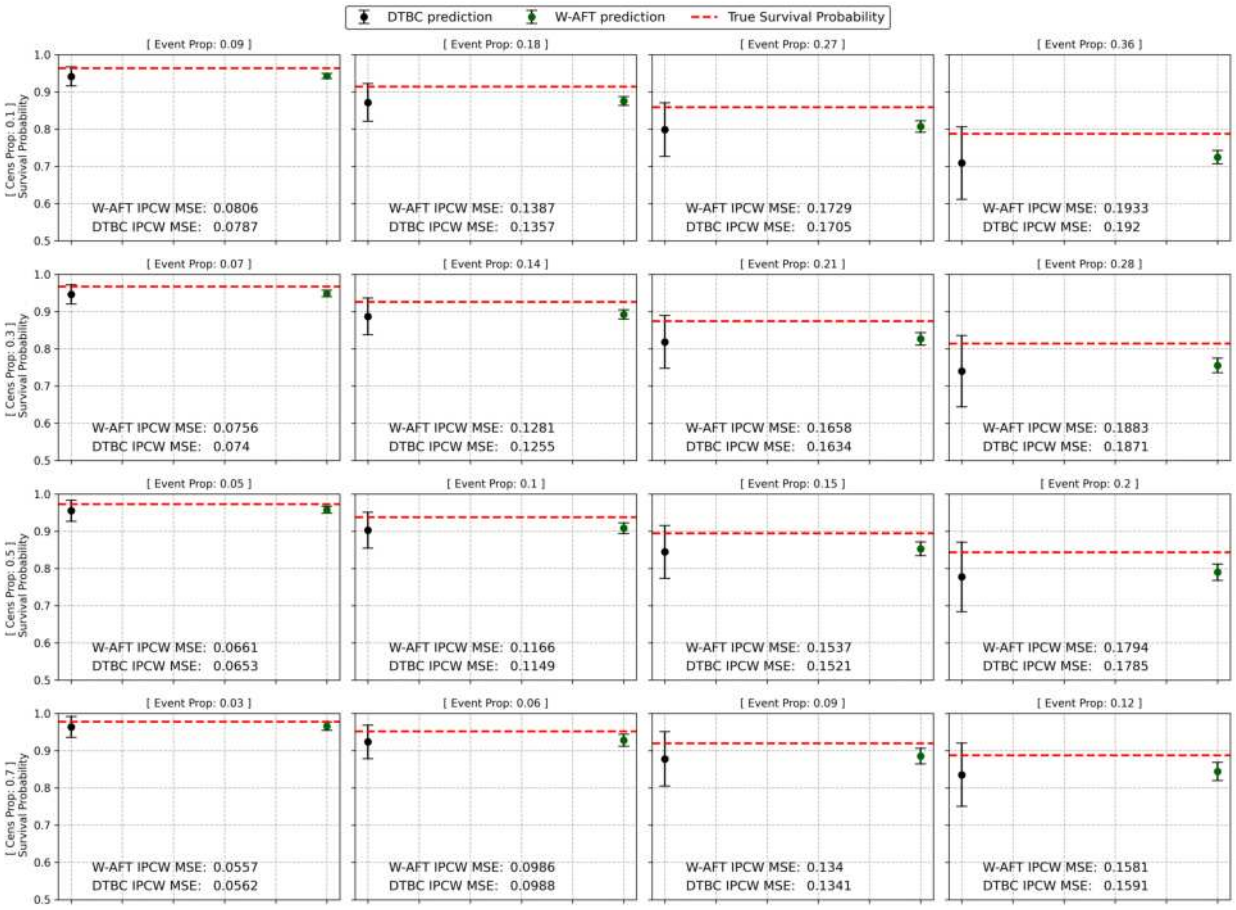
(b) Sim-Results for: $k = 1.5$, $n_{train} = 999$, $B = 4000$



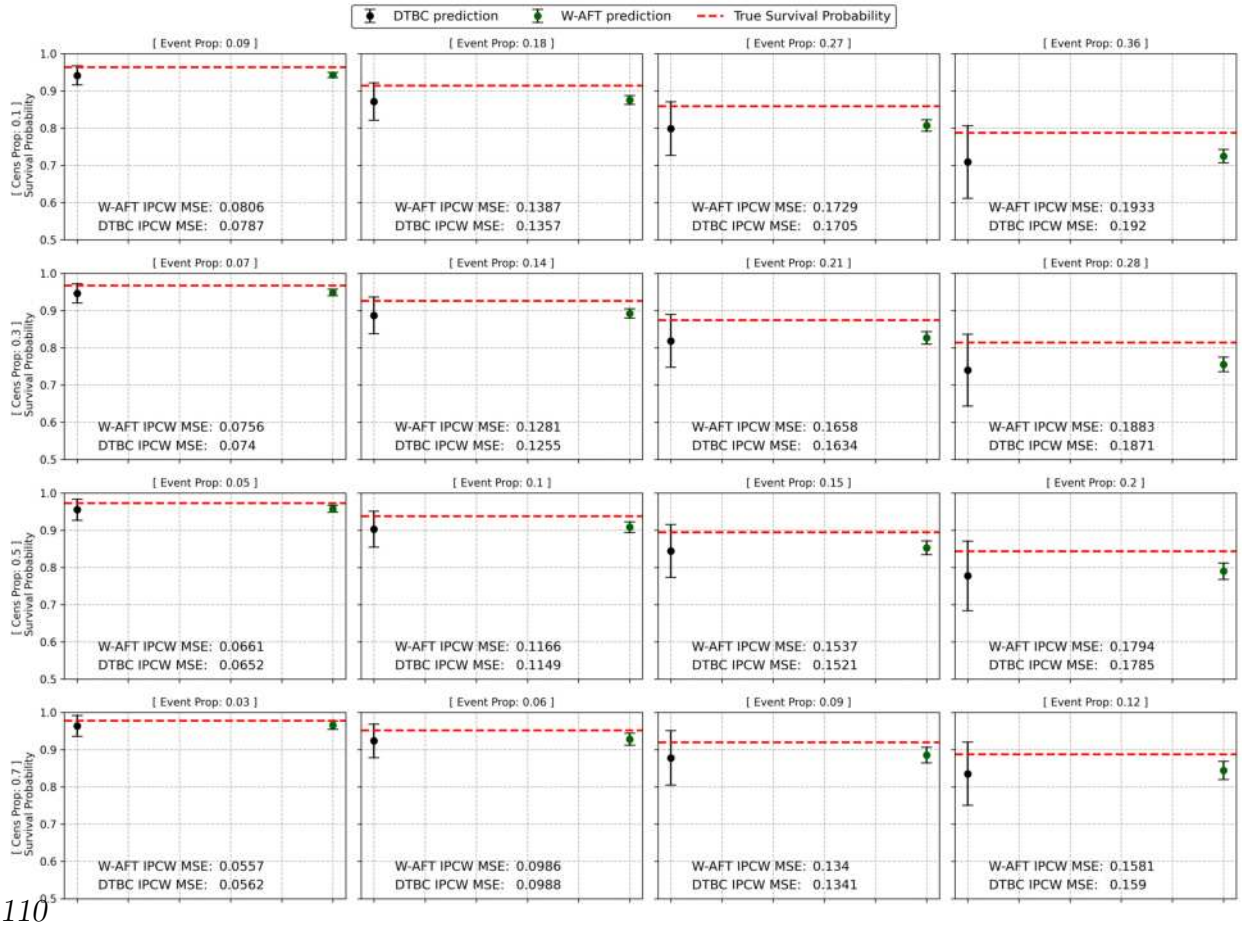
(a) Sim Results for: $k = 1.5$, $n_{train} = 1999$, $B = 500$



(b) Sim-Results for: $k = 1.5$, $n_{train} = 1999$, $B = 1000$



(a) Sim Results for: $k = 1.5$, $n_{train} = 1999$, $B = 2000$



(b) Sim-Results for: $k = 1.5$, $n_{train} = 1999$, $B = 4000$

List of Figures

List of Tables

Bibliography

[Bow28] A. L. Bowley. "The Standard Deviation of the Correlation Coefficient." In: *Journal of the American Statistical Association* 23.161 (1928), pp. 31–34. DOI: 10.2307/2277400. URL: https://www.jstor.org/stable/2277400.

[Bre96] Leo Breiman. "Bagging predictors." In: *Machine learning* 24.2 (1996), pp. 123–140. DOI: 10.1007/BF00058655. URL: https://doi.org/10.1007/BF00058655.

[DH97] A. C. Davison and D. V. Hinkley. *Bootstrap Methods and their Application*. Cambridge Series in Statistical and Probabilistic Mathematics. Cambridge University Press, 1997.

[Efr82] Bradley Efron. *The Jackknife, the Bootstrap and Other Resampling Plans*. Society for Industrial and Applied Mathematics, 1982. DOI: 10.1137/1.9781611970319. eprint: https://epubs.siam.org/doi/pdf/10.1137/1.9781611970319. URL: https://epubs.siam.org/doi/abs/10.1137/1.9781611970319.

[Efr92] Bradley Efron. "Jackknife-After-Bootstrap Standard Errors and Influence Functions." In: *Journal of the Royal Statistical Society. Series B (Methodological)* 54.1 (1992), pp. 83–127. ISSN: 00359246. URL: http://www.jstor.org/stable/2345949 (visited on 05/03/2024).

[EH16] Bradley Efron and Trevor Hastie. *Computer age statistical inference. algorithms, evidence, and data science*. eng. Institute of Mathematical Statistics monographs ARRAY(0x563ff1a7ed18). Literaturverzeichnis: Seiten 453-462 ; Hier auch später erschienene, unveränderte Nachdrucke. Cambridge, United Kingdom ; New York, NY, USA ; Port Melbourne, VIC, Australia ; New Delhi, India ; Singapore: Cambridge University Press, 2016, xix, 475 Seiten. ISBN: 978-1-107-14989-2.

[ES81] B. Efron and C. Stein. "The Jackknife Estimate of Variance." In: *The Annals of Statistics* 9.3 (1981), pp. 586–596. ISSN: 00905364. URL: http://www.jstor.org/stable/2240822 (visited on 05/03/2024).

[ET93] Bradley Efron and Robert J. Tibshirani. *An Introduction to the Bootstrap*. Monographs on Statistics and Applied Probability 57. Boca Raton, Florida, USA: Chapman & Hall/CRC, 1993.

[Gon+21] Pablo Gonzalez Ginestet et al. "Stacked Inverse Probability of Censoring Weighted Bagging: A Case Study In the InfCareHIV Register." In: *Journal of the Royal Statistical Society Series C: Applied Statistics* 70.1 (Jan. 2021), pp. 51–65. ISSN: 0035-9254. DOI: 10.1111/rssc.12448. eprint: https://academic.oup.com/jrsssc/article-pdf/70/1/51/49158571/rssc_70_1_51.pdf. URL: https://doi.org/10.1111/rssc.12448.

[Gow71] J.C. Gower. "A general coefficient of similarity and some of its properties." In: *Biometrics* 27.4 (1971), pp. 857–871.

[Gra+99] Erika Graf et al. "Assessment and comparison of prognostic classification schemes for survival data." In: *Statistics in Medicine* 18.17-18 (1999), pp. 2529–2545. DOI: https://doi.org/10.1002/(SICI)1097-0258(19990915/30)18:17/18<2529::AID-SIM274>3.0.CO;2-5. eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1002/%28SICI%291097-0258%2819990915/30%2918%3A17/18%3C2529%3A%3AAID-SIM274%3E3.0.CO%3B2-5. URL: https://onlinelibrary.wiley.com/doi/abs/10.1002/%28SICI%291097-0258%2819990915/30%2918%3A17/18%3C2529%3A%3AAID-SIM274%3E3.0.CO%3B2-5.

[Ham+86] Frank R. Hampel et al. *Robust Statistics: The Approach Based on Influence Functions*. New York, Chichester, Brisbane, Toronto, Singapore: John Wiley & Sons, 1986. ISBN: 978-0-471-73517-1.

[Jae72] Louis A Jaeckel. *The infinitesimal jackknife*. Bell Telephone Laboratories, 1972.

[Voc+16] David M. Vock et al. "Adapting machine learning techniques to censored time-to-event health record data: A general-purpose approach using inverse probability of censoring weighting." In: *Journal of Biomedical Informatics* 61 (2016), pp. 119–131. ISSN: 1532-0464. DOI: https://doi.org/10.1016/j.jbi.2016.03.009. URL: https://www.sciencedirect.com/science/article/pii/S1532046416000496.

[WHE14] Stefan Wager, Trevor Hastie, and Bradley Efron. *Confidence Intervals for Random Forests: The Jackknife and the Infinitesimal Jackknife*. 2014. arXiv: 1311.4555 [stat.ML].