American Journal of Theoretical and Applied Statistics
2026, Vol. 15, No. 5, pp. 257–275
https://doi.org/10.11648/j.ajtas.20261505.16

Research Article

Meta-Learning for Bayesian Computation in Survival Analysis: A Machine Learning Framework for Intelligent Bayesian Algorithm Recommendation

Khalid Salah[1, *], **Afnan Salah**[2]

[1]College of Science & Technology, Al-Quds University, Abu Dis, Palestine
[2]Department of General Medicine, Morriston Hospital, Swansea, United Kingdom

Abstract

Bayesian computation for survival analysis often involves selecting among alternative Markov chain Monte Carlo (MCMC) algorithms whose performance can vary substantially with the characteristics of the data and model. In practice, this selection is frequently based on trial and error, requiring repeated model fitting and computational assessment. We propose a meta-learning framework that formulates Bayesian computational strategy selection as a data-dependent prediction problem. A simulation-based meta-database is constructed by generating survival datasets under systematically varied sample sizes, model dimensions, predictor correlations, and censoring proportions. Four Bayesian computational strategies—Standard Random-Walk MCMC, Adaptive MCMC, Hamiltonian Monte Carlo, and Reversible Jump MCMC—are evaluated across the simulated environments. Estimation accuracy, sampling efficiency, and computational cost are integrated through a multi-criteria utility function to identify the preferred computational strategy for each dataset. Machine-learning models are then trained to learn the relationship between survival-data characteristics and the resulting algorithm-selection decisions. The learned recommendation models are evaluated using three independent real survival datasets: the German Breast Cancer Study Group breast cancer, Veteran lung cancer, and primary biliary cirrhosis datasets. Importantly, the real-data evaluation is conducted without retraining or recalibration of the meta-learning models. The proposed framework provides a systematic approach for transferring computational experience from simulated survival environments to new problems. By treating Bayesian algorithm selection as a data-dependent decision rather than assuming a universally optimal MCMC strategy, the framework offers a practical approach for reducing trial-and-error computation and supporting evidence-based selection of Bayesian computational methods.

Keywords

Bayesian Computation, Survival Analysis, Meta-learning, Algorithm Selection, Markov Chain Monte Carlo, Adaptive MCMC, Hamiltonian Monte Carlo, Reversible Jump MCMC

1. Introduction

Bayesian methods provide a flexible framework for survival analysis, allowing uncertainty to be quantified while accommodating censoring, frailty, nonlinear effects, competing risks, and hierarchical structures. Recent develop-

*Correspondence: Khalid Salah (ksalah@staff.alquds.edu)

Received: 21 August 2026; **Accepted:** 31 August 2026; **Published:** 22 September 2026

ments have expanded Bayesian survival models to increasingly complex clinical settings, while advances in Bayesian computation have provided a broad range of methods for posterior inference [1, 2, 7, 10]. Nevertheless, the practical performance of a Bayesian survival model depends not only on its statistical formulation but also on the computational strategy used for posterior sampling.

Markov chain Monte Carlo (MCMC) remains central to Bayesian computation, with Standard Random-Walk Metropolis-Hastings, Adaptive MCMC, Hamiltonian Monte Carlo (HMC), and Reversible Jump MCMC (RJMCMC) providing different approaches to posterior exploration [7, 10, 12]. Their performance can vary substantially according to sample size, model dimensionality, predictor correlation, censoring, posterior dependence, and the amount of event information. Recent research continues to emphasize that computational methods should be matched to the structure and difficulty of the underlying inferential problem rather than treated as universally interchangeable [7, 10, 12].

Consequently, selecting an appropriate Bayesian computational strategy is itself an important methodological problem. In practice, researchers commonly evaluate one or more algorithms and then compare convergence, effective sample size, estimation accuracy, and computational time. Such trial-and-error selection can become expensive when several candidate algorithms are available. Furthermore, no single diagnostic adequately represents overall computational performance. Algorithm selection therefore requires consideration of multiple objectives, including inferential accuracy, sampling efficiency, and computational cost.

Meta-learning provides a principled framework for addressing this problem. In automated machine learning (AutoML), meta-learning uses information accumulated from previous datasets or tasks to learn relationships between problem characteristics and algorithm performance [3, 4, 6]. Recent studies have demonstrated the use of meta-learning for automated algorithm selection, including selection of algorithms for tabular datasets and recommendation for previously unseen datasets [5, 8]. Other work has shown that meta-learners can support model recommendation under computational constraints and can generalize to new data sources [9, 11]. These developments suggest that previous computational experience can be transformed into transferable information for algorithm selection.

We extend this idea to Bayesian computation for survival analysis. Rather than assuming that one MCMC strategy is generally preferable, we formulate computational strategy selection as a data-dependent prediction problem. Characteristics of a survival dataset available before Bayesian model fitting are used to predict the computational strategy expected to provide the best overall performance. This requires an explicit definition of computational preference because accuracy, sampling efficiency, and computational cost may favor different algorithms.

We therefore propose a meta-learning framework for Bayesian algorithm selection in survival analysis. A simulation-based meta-database is constructed by generating survival datasets under systematically varied sample sizes, model dimensions, predictor correlations, and censoring proportions. Four Bayesian computational strategies—Standard Random-Walk MCMC, Adaptive MCMC, HMC, and RJMCMC—are evaluated across these environments. Estimation accuracy, sampling efficiency, and computational cost are integrated through a multi-criteria utility function to identify the preferred strategy for each simulated problem. Machine-learning models are then trained to learn the relationship between survival-data characteristics and these algorithm-selection decisions.

A key feature of the framework is its evaluation beyond the simulated training environment. After training on the simulation-based meta-database, the recommendation models are applied to three independent real survival datasets—the German Breast Cancer Study Group (GBSG) breast cancer data, Veteran lung cancer data, and primary biliary cirrhosis (PBC) data—without retraining or recalibration. The recommended algorithms are subsequently compared with alternative strategies using Bayesian model-fit and computational-performance measures. This provides an external assessment of whether computational knowledge learned from simulated survival environments can transfer to real-world survival applications.

The main contributions are: (i) formulation of Bayesian computational strategy selection as a meta-learning problem for survival analysis; (ii) construction of a simulation-based meta-database linking survival-data characteristics with Bayesian computational performance; (iii) development of a multi-criteria utility framework for defining the preferred computational strategy; (iv) identification of influential dataset characteristics associated with algorithm selection; and (v) external validation of the learned recommendations using independent real survival datasets. Thus, rather than treating Bayesian computation as a fixed methodological choice, the proposed framework treats algorithm selection as a data-dependent computational decision, with the aim of reducing repeated trial-and-error benchmarking and supporting evidence-based selection of Bayesian computational strategies.

Scope of the study. Section 2 presents the proposed Bayesian computational and meta-learning framework. Section 3 describes the simulation design, candidate algorithms, performance measures, utility function, and machine-learning models. Section 4 presents the real-data applications and external validation using the GBSG, Veteran, and PBC datasets. Section 5 discusses the findings, implications, limitations, and future research directions, while Section 6 concludes the study. A list of abbreviations used throughout the manuscript is provided before the References, and the Supplementary Material is provided after the References.

2. Methodology

2.1. Meta-Learning Framework for Bayesian Computational Strategy Selection

We propose a meta-learning framework for automated selection of Bayesian computational strategies in survival analysis. The proposed framework addresses the problem of determining an appropriate posterior sampling algorithm based on the intrinsic characteristics of survival datasets. Instead of assuming that a single computational algorithm universally provides the best performance across all survival datasets, the proposed approach formulates Bayesian algorithm selection as a data-driven learning problem, where computational performance is learned from previously evaluated datasets.

The proposed framework integrates three main components: (i) a survival dataset characterization module, (ii) a Bayesian computational benchmarking module, and (iii) a meta-learning recommendation model. The dataset characterization component extracts informative features describing the statistical and computational complexity of survival datasets. The benchmarking component evaluates competing Bayesian computational strategies under diverse data conditions. The meta-learning component learns the relationship between dataset characteristics and algorithmic performance and recommends the computational strategy expected to achieve the best observed performance under predefined evaluation criteria for new survival datasets.

For the i-th survival dataset, let $Z_i = (z_{i1}, z_{i2}, \ldots, z_{im})$ denote the vector of dataset-level characteristics. Let the available Bayesian computational strategies be represented by:

$$A_i = \{A_1, A_2, \ldots, z_k\}.$$

The objective of the proposed framework is to learn the relationship:

$$f(Z_i) \to A_k$$

where A_k represents the Bayesian computational strategy that achieves the best observed performance according to the predefined evaluation criteria is assigned as the recommended computational strategy label for meta-learning.

The proposed framework relies on a benchmark library consisting of Bayesian computational algorithms representing different posterior sampling paradigms. The purpose of this library is not to identify a universally superior algorithm, but rather to evaluate how computational performance varies according to survival dataset characteristics for dataset D_i according to the predefined evaluation criteria.

2.2. Bayesian Computational Strategy Library

The proposed framework relies on a benchmark library consisting of Bayesian computational algorithms representing different posterior sampling paradigms. The purpose of this library is not to identify a universally superior algorithm, but rather to determine how computational performance varies according to the characteristics and complexity of survival datasets.

A reference computational strategy is included as the standard comparator. The reference algorithm represents a commonly used, transparent, and easily implementable Bayesian sampling approach. Its role is to provide a consistent baseline for evaluating alternative computational strategies rather than representing an assumed optimal method.

The competing algorithms are evaluated under identical simulation conditions, and for each dataset the algorithm achieving the highest observed performance according to the predefined evaluation criteria is assigned as the recommended computational strategy label for meta-learning.

2.3. Survival Dataset Feature Representation

The meta-learning framework operates on characteristics of survival datasets that may influence Bayesian computational performance.

For each dataset D_i, a feature vector is constructed as:

$$Z_i = (n_i, p_i, C_i, R_i, V_i)$$

where:

n_i = sample size, p_i = number of covariates, C_i = censoring-related characteristics, R_i = covariate correlation structure, V_i = additional measures describing data complexity.

The extracted features include sample size, dimensionality, censoring proportion, event frequency, predictor correlation structure, and measures associated with posterior complexity.

These characteristics are selected because they are available before model fitting and may provide predictive information regarding the computational behavior of Bayesian algorithms.

2.4. Computational Performance Assessment

For each dataset-algorithm combination, posterior inference is performed and evaluated using complementary measures representing inferential accuracy, sampling efficiency, and computational cost.

Parameter estimation accuracy is assessed using the root mean square error (RMSE):

$$RMSE = \sqrt{\frac{1}{q}\left(\hat{\theta}_j - \theta_j\right)^2}$$

where q = number of model parameters, $\hat{\theta}_j$= estimated value of parameter j, θ_j = true parameter value.

Sampling efficiency is evaluated using the effective sample size (ESS) and integrated autocorrelation time (IACT). The ESS quantifies the amount of independent information contained within the posterior samples, whereas IACT measures

the degree of serial dependence among successive samples.

Computational efficiency is assessed using CPU execution time. Additionally, convergence diagnostics and acceptance behavior are examined to ensure adequate posterior exploration and reliable inference.

A unified performance criterion is constructed by integrating estimation accuracy, sampling efficiency, and computational cost. For each survival dataset, the Bayesian computational strategy achieving the highest observed performance score among the evaluated algorithms is assigned as the preferred computational strategy for that dataset.

The resulting dataset-algorithm performance pairs are then used to construct the meta-learning database. Accordingly, the objective of the proposed framework is not to identify a universally dominant Bayesian algorithm, but rather to learn the relationship between survival dataset characteristics and the computational strategy that provides the most favorable balance between estimation accuracy, sampling efficiency, and computational cost under specific data conditions.

2.5. Meta-Learning-Based Algorithm Recommendation

The algorithm recommendation problem is formulated as a supervised learning problem. Each evaluated survival dataset generates a meta-learning observation:

$$(Z_i, Y_i)$$

where Z_i = vector of dataset characteristics, Y_i = Bayesian computational algorithm achieving the best performance.

The meta-learning model estimates:

Probability of algorithm A_k being optimal given dataset characteristics Z_i, or $P(\ Y_i = A_k \mid Z_i)$. The predicted probabilities provide both the recommended computational strategy and the confidence associated with that recommendation.

Therefore, the framework does not simply provide a fixed algorithmic choice; instead, it learns the relationship between survival dataset characteristics and Bayesian computational performance.

2.6. Evaluation of the Proposed Framework

The proposed framework is evaluated using independent survival datasets not included during model training. The recommended computational strategy is compared with alternative Bayesian algorithms according to posterior estimation accuracy, sampling efficiency, and computational cost.

The objective is to determine whether machine learning can effectively identify the Bayesian computational strategy most suitable for a given survival dataset and reduce dependence on manual algorithm selection.

3. Simulation Study

3.1. Simulation Design and Scenarios

A comprehensive simulation study was conducted to evaluate the proposed meta-learning framework for intelligent selection of Bayesian computational strategies in survival analysis. The objective was not to identify a universally optimal computational algorithm, but to determine whether machine-learning models can learn the relationship between survival-data characteristics and the Bayesian computational strategy that provides favorable computational performance under specific data conditions.

The simulation design considered four characteristics expected to influence the computational behavior of Bayesian inference algorithms: sample size, model dimensionality, covariate correlation, and censoring proportion. These factors affect posterior dimensionality, parameter dependence, information availability, and the efficiency of posterior sampling.

Four simulation factors were varied:

1) Sample size (n): n = 50, 100, and 300, representing small, moderate, and relatively large survival datasets.
2) Number of covariates (p): p = 5 and 15, representing low- and higher-dimensional regression models.
3) Covariate correlation (ρ): $\rho = 0.3$ and 0.8, representing moderate and strong dependence among predictors.
4) Censoring proportion: 30% and 60%, representing moderate and substantial loss of event information.

The survival model structure was fixed across all scenarios. Survival times were generated from a Weibull proportional hazards model with shape parameter $\gamma = 1.5$. All covariates were assigned non-zero regression effects so that the simulation focused on computational complexity associated with model dimension rather than variable-selection difficulty.

Let $\mathcal{N} = \{50, 100, 300\}$, $\mathcal{P} = \{5, 15\}$, $\mathcal{R} = \{0.3, 0.8\}$, and $\mathcal{C} = \{30\%, 60\%\}$ denote the sets of sample sizes, numbers of covariates, correlation levels, and censoring proportions, respectively. The complete simulation space was defined as the Cartesian product $\mathcal{S} = \mathcal{N} \times \mathcal{P} \times \mathcal{R} \times \mathcal{C}$.

Thus, each simulation scenario was represented by

$$S_i = (n_j, p_k, \rho_l, \mathrm{C}_m), \quad i = 1, \ldots, 24, \quad j = 1, 2, 3, \quad k, l, m = 1, 2.$$

The total number of scenarios was therefore

$$|\mathcal{S}| = |\mathcal{N}| \times |\mathcal{P}| \times |\mathcal{R}| \times |\mathcal{C}| = 3 \times 2 \times 2 \times 2 = 24.$$

This factorial structure provided a systematic range of survival-data environments from which the meta-learning framework could learn the association between dataset characteristics and the computational performance of competing Bayesian algorithms.

3.2. Data Generation Mechanism

For each of the 24 simulation scenarios, survival datasets were generated from a Weibull proportional hazards (PH) model. The data-generation mechanism incorporated covariate dependence, model dimensionality, and right censoring to reproduce key characteristics that may affect Bayesian computational performance.

For each subject i, the covariate vector $\mathrm{X}_i = (X_{i1}, \ldots, X_{ip})$ was generated from a multivariate normal distribution:

$$\mathrm{X}_i \sim \mathrm{MVN}(0, \Sigma),$$

where Σ controls the correlation structure among covariates. An autoregressive correlation structure was used, with

$$\Sigma_{jk} = \rho^{|j-k|},$$

where ρ was set to either 0.3 or 0.8 according to the simulation scenario.

Conditional on the generated covariates, the true survival time T_i was generated from a Weibull PH model with hazard function

$$h(t \mid \mathrm{X}_i) = \gamma t^{\gamma-1} \exp(\mathrm{X}_i^T \beta),$$

where γ is the Weibull shape parameter and β is the vector of regression coefficients. The shape parameter was fixed at $\gamma = 1.5$ across all scenarios. Survival times were generated using the inverse-transform method:

$$T_i = \left[\frac{-\log(U_i)}{\exp(\mathrm{X}_i^T \beta)}\right]^{1/\gamma},$$

where $U_i \sim \mathrm{Uniform}(0, 1)$.

The regression coefficients were specified such that all covariates had non-zero effects, thereby allowing the effect of increasing model dimensionality to be evaluated without introducing variable-selection complexity.

Independent censoring times C_i were generated from a uniform distribution and calibrated to achieve the target censoring proportion for each scenario. The observed survival time and event indicator were then defined as

$$Y_i = \min(T_i, C_i), \delta_i = \mathrm{I}(T_i \le C_i)$$

where Y_i denotes the observed follow-up time and δ_i indicates whether the event was observed.

The resulting datasets contained survival outcomes and right-censoring patterns corresponding to each predefined simulation scenario. These datasets were subsequently analyzed using the competing Bayesian computational strategies, and the resulting computational performance measures were combined with the scenario-level characteristics to construct the meta-learning database.

3.3. Bayesian Computational Strategies

The proposed meta-learning framework was designed to recommend a Bayesian computational strategy according to the characteristics of a given survival dataset. Four computational strategies representing distinct sampling approaches were considered:

1) Standard Random-Walk Markov Chain Monte Carlo (MCMC): a Random-Walk Metropolis-Hastings algorithm used as the baseline strategy because of its simplicity and broad applicability.
2) Adaptive MCMC: an adaptive Metropolis-Hastings approach in which the proposal scale was updated during the burn-in period to improve posterior exploration. Adaptation was performed every 100 iterations during burn-in. The proposal scale was decreased by 10% when the cumulative acceptance rate was below 0.20 and increased by 10% when the acceptance rate exceeded 0.35. No adaptation was performed after the burn-in period.
3) Hamiltonian Monte Carlo (HMC): a gradient-based sampling method that uses Hamiltonian dynamics to generate efficient proposals and is particularly suitable for posterior distributions with strong parameter dependence or higher dimensionality.
4) Reversible Jump Markov Chain Monte Carlo (RJMCMC): an extension of MCMC that permits transitions between parameter spaces of different dimensions and is therefore capable of handling variable-dimensional Bayesian model spaces.

All four computational strategies were applied to the same simulated survival datasets under a common Bayesian model specification and common prior distributions. For the regression coefficients, independent zero-centered normal priors were specified as:

$$\beta_j \sim N(0, 100), \qquad j = 1 \ldots, p$$

where the prior variance was 100. The positive parameter γ was assigned a Gamma prior,$\gamma \sim \mathrm{Gamma}(2, 1)$, where 2 and 1 denote the shape and rate parameters, respectively. The prior specification therefore corresponds directly to the implementation used in the simulation code. The positivity constraint $\gamma > 0$ was enforced during posterior sampling. These proper, relatively diffuse priors were used consistently across the four computational strategies so that differences in computational performance reflected the sampling strategies rather than differences in prior specification.

For each simulated dataset and computational strategy, five independent Markov chains were generated. Each chain consisted of 5,000 iterations, with the first 2,000 iterations discarded as burn-in or warm-up. The retained samples were thinned by a factor of 9 to reduce serial dependence in the stored posterior draws. Thus, the same nominal sampling schedule was used across the four computational strategies, while algorithm-specific proposal mechanisms and tuning

procedures were applied according to the requirements of each method.

For Adaptive MCMC, proposal adaptation was restricted to the burn-in period to avoid continued modification of the sampling distribution during posterior inference. The proposal scale was updated at 100-iteration intervals according to the observed cumulative acceptance rate, using the thresholds described above. Standard Random-Walk MCMC used a fixed proposal mechanism, whereas HMC and RJMCMC employed their respective gradient-based and dimension-changing sampling mechanisms without the adaptive random-walk proposal adjustment used for Adaptive MCMC.

The posterior samples generated from each strategy were subsequently used to obtain the common computational performance measures used throughout the simulation study, including effective sample size (ESS), integrated autocorrelation time (IACT), root mean square error (RMSE), acceptance rate where applicable, and CPU execution time. These measures were then combined using the unified performance criterion described in Section 2.4.

Each computational strategy was applied independently to every simulated dataset. Across the 24 scenarios and 100 independent datasets per scenario, the four strategies produced 9,600 algorithm-dataset fits. These computational experiments provided the performance information required to construct the meta-learning database.

The purpose of this comparison was not to identify a universally superior Bayesian algorithm. Rather, the objective was to characterize how computational performance changes across survival-data environments and use these observed patterns as training information for the subsequent meta-learning framework.

3.4. Dataset Characterization and Meta-Learning Database Construction

The simulated datasets generated from the factorial design were used to construct the meta-learning database. The 24 simulation scenarios defined the space of survival-data environments, while independent Monte Carlo replications within each scenario provided the computational experiences required for model training and evaluation.

For each scenario, 100 independent survival datasets were generated, resulting in a total of 2400 simulated datasets. Each generated dataset was characterized using the predefined survival-data features representing the main sources of computational complexity, including sample size, number of covariates, covariate correlation structure, censoring proportion, and observed event frequency.

For each simulated dataset, the competing Bayesian computational strategies were independently evaluated under identical model specifications, prior distributions, and computational settings. The resulting performance measures were integrated using the unified performance criterion described in Section 2.4. The computational strategy achieving the highest overall performance score was assigned as the recommended strategy for that dataset. Accordingly, the target label for dataset D_i was defined as:

$$Y_i = \arg\max_k Score(D_i, A_k)$$

where A_k represents the k-th Bayesian computational strategy evaluated for dataset D_i.

Therefore, each simulated dataset generated one meta-learning observation consisting of the dataset characteristics and the corresponding recommended computational strategy. The resulting database contained 2400 dataset-level observations linking survival-data characteristics with computational performance outcomes. This database was subsequently used to train and evaluate machine learning models for learning the relationship between survival-data characteristics and the Bayesian computational strategy expected to provide the best performance for new survival datasets.

3.5. Machine Learning Models for Bayesian Computational Strategy Recommendation

The constructed meta-learning database was used to train supervised machine-learning models for recommending the Bayesian computational strategy expected to provide the best performance for a given survival dataset. Each simulated dataset was considered as one learning instance, with dataset-level characteristics representing the input features and the computational strategy achieving the highest overall performance score representing the target outcome.

Four machine-learning algorithms were evaluated to represent complementary predictive modeling approaches: multinomial logistic regression, Random Forest, support vector machine (SVM), and k-nearest neighbors (k-NN). Multinomial logistic regression was included as a relatively simple parametric baseline, providing an interpretable reference for multiclass strategy recommendation. Random Forest was selected as a nonlinear ensemble method capable of capturing complex interactions among dataset characteristics without requiring a prespecified functional form. k-NN was included as an instance-based learning method that identifies similarities among datasets in the meta-feature space. SVM was included as a margin-based classification method capable of representing nonlinear decision boundaries. The use of these four complementary approaches allowed the predictive performance of simple, ensemble-based, local, and margin-based learning methods to be compared within a common meta-learning framework.

The predictor variables consisted of five predefined dataset-level meta-features: sample size, number of covariates, covariate correlation, censoring proportion, and observed event proportion. Before model training, the meta-learning data were preprocessed according to the requirements of the individual machine-learning algorithms. The meta-learning database was divided into training and independent testing subsets.

The training data were used for model development and parameter tuning, whereas the independent testing data were reserved for the final evaluation of predictive performance.

For each machine-learning model, the objective was to estimate the probability that each candidate Bayesian computational strategy would achieve the highest observed overall performance score for a new survival dataset. The predicted class probabilities were used to determine the recommended computational strategy by selecting the strategy with the highest predicted probability.

All four machine-learning models were evaluated using the same target definition and independent testing data to facilitate a direct comparison of their predictive performance. This design allowed the relative ability of different machine-learning approaches to learn the relationship between survival-data characteristics and Bayesian computational strategy performance to be assessed under a common meta-learning framework.

3.6. Evaluation Criteria for the Meta-Learning Framework

The performance of the proposed meta-learning framework was evaluated from both predictive and computational perspectives. The predictive evaluation assessed the ability of machine learning models to correctly identify the Bayesian computational strategy expected to achieve the highest performance for a given survival dataset. The computational evaluation examined whether the recommended strategy provided improved posterior inference performance compared with alternative computational strategies.

For predictive evaluation, the recommendation models were assessed using standard classification measures, including accuracy, balanced accuracy, precision, recall, F1-score, and Cohen's kappa coefficient. Accuracy measures the overall proportion of correctly recommended computational strategies, whereas balanced accuracy accounts for potential differences in the frequency of recommended strategies. Precision, recall, and F1-score provide additional evaluation of classification performance across individual computational strategies, while Cohen's kappa measures the agreement between the predicted and true recommended strategies beyond chance agreement.

In addition to classification performance, the practical benefit of the recommendation framework was evaluated by comparing the computational performance obtained using the recommended strategy with the performance of alternative Bayesian computational strategies. The comparison was conducted using the same criteria applied during algorithm benchmarking, including parameter estimation accuracy, sampling efficiency, and computational cost. Specifically, the evaluation considered root mean square error (RMSE), effective sample size (ESS), integrated autocorrelation time (IACT), and CPU execution time.

The effectiveness of the proposed framework was determined by assessing whether the machine learning-based recommendation was able to identify computational strategies that achieved performance comparable to or better than non-adaptive algorithm selection approaches. This evaluation demonstrates the ability of the proposed framework to transform Bayesian computational strategy selection into a data-driven recommendation process.

3.7. Simulation Results

The simulation results were evaluated in two complementary stages. First, the computational performance of the four Bayesian sampling strategies was examined across survival-data environments differing in sample size, model dimensionality, covariate correlation, and censoring. Second, the resulting meta-learning database was used to evaluate the ability of machine-learning models to recommend an appropriate Bayesian computational strategy from dataset-level characteristics.

3.7.1. Computational Performance of Bayesian Sampling Strategies

The simulation experiment comprised 24 survival-data scenarios and four Bayesian computational strategies, corresponding to 96 scenario-strategy combinations. Because each scenario was replicated 100 times, the complete experiment generated 2,400 simulated datasets and 9,600 algorithm-dataset fits. Representative scenarios covering high, moderate, and low computational complexity are presented in Tables 2-4, while the complete simulation results are provided in Supplementary Table 1.

Table 1. *Key of Simulation Scenarios.*

Scenario	Sample Size	Covariates (p)	Correlation (ρ)	Censoring (%)	Scenario Name
S1	50	5	0.3	30	N50_P5_RHO0.3_C30
S2	50	5	0.3	60	N50_P5_RHO0.3_C60
S3	50	5	0.8	30	N50_P5_RHO0.8_C30
S4	50	5	0.8	60	N50_P5_RHO0.8_C60

Scenario	Sample Size	Covariates (p)	Correlation (ρ)	Censoring (%)	Scenario Name
S5	50	15	0.3	30	N50_P15_RHO0.3_C30
S6	50	15	0.3	60	N50_P15_RHO0.3_C60
S7	50	15	0.8	30	N50_P15_RHO0.8_C30
S8	50	15	0.8	60	N50_P15_RHO0.8_C60
S9	100	5	0.3	30	N100_P5_RHO0.3_C30
S10	100	5	0.3	60	N100_P5_RHO0.3_C60
S11	100	5	0.8	30	N100_P5_RHO0.8_C30
S12	100	5	0.8	60	N100_P5_RHO0.8_C60
S13	100	15	0.3	30	N100_P15_RHO0.3_C30
S14	100	15	0.3	60	N100_P15_RHO0.3_C60
S15	100	15	0.8	30	N100_P15_RHO0.8_C30
S16	100	15	0.8	60	N100_P15_RHO0.8_C60
S17	300	5	0.3	30	N300_P5_RHO0.3_C30
S18	300	5	0.3	60	N300_P5_RHO0.3_C60
S19	300	5	0.8	30	N300_P5_RHO0.8_C30
S20	300	5	0.8	60	N300_P5_RHO0.8_C60
S21	300	15	0.3	30	N300_P15_RHO0.3_C30
S22	300	15	0.3	60	N300_P15_RHO0.3_C60
S23	300	15	0.8	30	N300_P15_RHO0.8_C30
S24	300	15	0.8	60	N300_P15_RHO0.8_C60

Table 2 presents selected scenarios for n = 50, Table 3 presents selected scenarios for n = 100, and Table 4 presents selected scenarios for n = 300. Across the reported scenarios, the computational profiles of the Bayesian strategies varied substantially with the characteristics of the survival datasets.

Table 2. *Bayesian Computational Performance for Selected Simulation Scenarios with Small Sample Size (n = 50).*

Scenario	Method	ESS	AR	IACT	RMSE	CPUTs
S1	Random-Walk MCMC	67.88	0.25	34.89	0.24	0.18
	Adaptive MCMC	112.65	0.29	18.26	0.25	0.65
	HMC	360.15	1.00	5.54	0.24	44.13
	RJMCMC	25.87	0.71	94.43	0.25	0.71
S6	Random-Walk MCMC	15.44	0.27	115.06	0.80	0.28
	Adaptive MCMC	36.53	0.35	83.47	0.64	0.48
	HMC	71.35	1.00	62.33	0.90	180.62
	RJMCMC	8.98	0.68	155.59	0.58	0.59
S8	Random-Walk MCMC	8.50	0.31	141.96	1.05	0.28
	Adaptive MCMC	31.17	0.38	93.24	1.18	0.50

Scenario	Method	ESS	AR	IACT	RMSE	CPUTs
	HMC	33.93	1.00	79.29	1.27	180.93
	RJMCMC	7.67	0.68	154.34	0.80	0.59

AR: Acceptance Rate, CPUTs: CPU Time (s)

Table 3. *Bayesian Computational Performance for Selected Simulation Scenarios with Small Sample Size (n = 100).*

Scenario	Method	ESS	AR	IACT	RMSE	CPUTs
S9	Random-Walk MCMC	76.55	0.29	33.25	0.14	0.21
	Adaptive MCMC	113.83	0.26	18.82	0.14	0.32
	HMC	974.96	1.00	2.34	0.14	49.15
	RJMCMC	41.98	0.62	67.84	0.14	0.46
S14	Random-Walk MCMC	25.96	0.23	81.55	0.27	0.32
	Adaptive MCMC	36.94	0.32	79.36	0.24	0.48
	HMC	375.86	1.00	4.87	0.25	198.09
	RJMCMC	15.32	0.60	83.81	0.24	0.65
S16	Random-Walk MCMC	12.13	0.25	125.07	0.42	0.31
	Adaptive MCMC	35.66	0.36	82.58	0.41	0.49
	HMC	100.75	1.00	21.29	0.40	198.22
	RJMCMC	9.87	0.61	137.90	0.40	0.64

AR: Acceptance Rate, CPUTs: CPU Time (s)

Table 4. *Bayesian Computational Performance for Selected Simulation Scenarios with Small Sample Size (n = 300).*

Scenario	Method	ESS	AR	IACT	RMSE	CPUTs
S17	Random-Walk MCMC	84.30	0.28	25.83	0.08	0.32
	Adaptive MCMC	110.44	0.21	19.03	0.07	0.41
	HMC	6610.31	1.00	0.38	0.07	73.32
	RJMCMC	62.22	0.42	46.84	0.08	0.62
S22	Random-Walk MCMC	28.27	0.32	74.68	0.11	0.41
	Adaptive MCMC	44.37	0.27	71.05	0.12	0.61
	HMC	2586.57	1.00	0.72	0.10	271.28
	RJMCMC	19.83	0.43	92.38	0.11	0.91
S24	Random-Walk MCMC	13.67	0.29	120.27	0.21	0.42
	Adaptive MCMC	36.28	0.32	80.85	0.23	0.61
	HMC	444.68	1.00	5.30	0.21	272.59
	RJMCMC	14.69	0.46	125.41	0.23	0.86

AR: Acceptance Rate, CPUTs: CPU Time (s)

The Standard Random-Walk MCMC algorithm generally required the least CPU time, reflecting its simple proposal mechanism. However, its sampling efficiency decreased substantially in more challenging settings. For example, under S8, the ESS was only 8.50 with an IACT of 141.96. These results indicate slower posterior exploration when the model involved a small sample size, a larger number of highly correlated covariates, and substantial censoring.

Adaptive MCMC generally improved sampling efficiency relative to Standard Random-Walk MCMC. Under S8, for example, Adaptive MCMC increased the ESS from 8.50 to 31.17 and reduced the IACT from 141.96 to 93.24, while requiring only a moderate increase in CPU time from 0.278 to 0.504 seconds. Similar improvements were observed in several other scenarios, indicating that adaptation can provide a useful balance between sampling efficiency and computational cost.

HMC consistently produced highly efficient posterior exploration in terms of ESS and IACT. In the large-sample, low-complexity scenario S17, HMC achieved an ESS of 6,610.31 and an IACT of 0.38. However, this sampling efficiency was accompanied by substantially higher computational cost, with a CPU time of 73.324 seconds compared with 0.322 seconds for Standard Random-Walk MCMC and 0.412 seconds for Adaptive MCMC. Thus, a large ESS alone did not imply the highest computational efficiency.

RJMCMC generally produced lower ESS values and larger IACT values than the other approaches in the reported scenarios. Although its acceptance rates were generally acceptable, the additional computational complexity associated with reversible-jump moves resulted in lower sampling efficiency in many settings.

The RMSE results also demonstrate that no single strategy dominated all performance dimensions. For example, in S14, RJMCMC produced the smallest RMSE (0.2347), whereas in S17, Adaptive MCMC achieved the smallest RMSE (0.0698). These results reinforce the need for a multi-criteria assessment rather than selecting an algorithm according to a single computational measure.

To illustrate the differences in computational efficiency more directly, three representative scenarios were selected: S8, representing a high-complexity environment; S14, representing a moderate-complexity environment; and S17, representing a low-complexity environment. Their characteristics and relative computational efficiencies are presented in Table 5.

Table 5. *Relative Computational Efficiency of Bayesian Sampling Strategies Under Representative Simulation Scenarios.*

Scenario (Chr)	Method	CPUTs	ESS	ESS/s	RE	AR
S8 (High)	Random-Walk MCMC	0.28	8.50	30.56	100.00	0.31
	Adaptive MCMC	0.50	31.17	61.85	202.35	0.38
	RJMCMC	0.59	7.67	13.09	42.82	0.68
	HMC	180.93	33.93	0.19	0.61	1.00
S14 (Moderate)	Random-Walk MCMC	0.32	25.96	80.62	100.00	0.23
	Adaptive MCMC	0.48	36.94	77.60	96.25	0.32
	RJMCMC	0.65	15.32	23.57	29.24	0.60
	HMC	198.09	375.86	1.90	2.35	1.00
S17 (Low)	Random-Walk MCMC	0.32	84.30	261.80	100.00	0.28
	Adaptive MCMC	0.41	110.44	268.05	102.39	0.21
	RJMCMC	0.62	62.22	100.35	38.33	0.42
	HMC	73.32	6610.31	90.15	34.44	1.00

Chr: Characteristics, CPUTs: CPU Time (s), ESS/s: ESS per s, RE: Relative Efficiency, AR: Acceptance Rate.

In the high-complexity scenario S8, Adaptive MCMC achieved the highest sampling efficiency, with 61.85 effective samples per second and a relative efficiency of 202.35% compared with Standard Random-Walk MCMC. HMC achieved a larger ESS than Adaptive MCMC but required substantially more CPU time, resulting in an ESS-per-second value of only 0.19.

In the moderate-complexity scenario S14, Standard Random-Walk MCMC achieved the highest ESS-per-second efficiency (80.62), with Adaptive MCMC providing a comparable efficiency of 77.60, corresponding to 96.25% relative efficiency. HMC again produced substantially greater ESS and lower IACT but required considerably more computation.

In the low-complexity scenario S17, Adaptive MCMC

slightly outperformed Standard Random-Walk MCMC in ESS per second, achieving 268.05 compared with 261.80, corresponding to a relative efficiency of 102.39%. HMC produced by far the largest ESS (6,610.31) and the smallest IACT (0.38), but its ESS-per-second efficiency remained lower than those of Standard and Adaptive MCMC because of its greater computational cost.

Overall, the results demonstrate that Bayesian computational performance depends strongly on the characteristics of the survival dataset. HMC provided highly efficient posterior exploration but incurred substantially greater computational cost, whereas Adaptive MCMC frequently provided a favorable balance between sampling efficiency and computation time. Standard Random-Walk MCMC remained highly competitive in simpler settings because of its very low computational cost. RJMCMC generally showed lower sampling efficiency in the examined scenarios.

These results provide direct empirical motivation for the proposed meta-learning framework: because the relative advantages of the algorithms vary across data environments, algorithm selection can reasonably be formulated as a prediction problem rather than as a fixed methodological choice.

3.7.2. Meta-Learning Results for Bayesian Computational Strategy Recommendation

The meta-learning database was used to evaluate four supervised machine-learning models for predicting the Bayesian computational strategy associated with the highest multi-criteria utility score. The predictor variables comprised five dataset-level meta-features: sample size, number of covariates, covariate correlation, censoring proportion, and observed event proportion. The target variable was the Bayesian computational strategy achieving the highest overall utility score.

The predictive results for the independent test set are presented in Table 6. Among the evaluated models, Random Forest demonstrated the strongest overall predictive performance. It achieved a ROC-AUC of 0.894 and an accuracy of 81.67%, together with precision of 83.52%, recall of 91.57%, F1-score of 87.36%, and Cohen's kappa of 0.543.

Table 6. *Predictive Performance of Machine Learning Models for Bayesian Computational Strategy Recommendation.*

Machine Learning Model	ROC-AUC	Accuracy (%)	Precision (%)	Recall (%)	F1-score (%)	C_κ
Multinomial Logistic Regression	0.73	68.33	74.19	83.13	78.41	0.20
Random Forest	0.89	81.67	83.52	91.57	87.36	0.54
Support Vector Machine	0.76	75.00	77.89	89.16	83.15	0.36
k-Nearest Neighbors	0.80	75.83	79.35	87.95	83.43	0.39

C_κ: Cohen's Kappa,

The other models also demonstrated predictive capability, but with lower overall performance. k-NN achieved a ROC-AUC of 0.795 and an accuracy of 75.83%, followed by SVM with a ROC-AUC of 0.761 and an accuracy of 75.00%. Multinomial logistic regression provided the weakest overall discrimination, with a ROC-AUC of 0.725 and an accuracy of 68.33%. The superior performance of Random Forest suggests that the relationship between survival-data characteristics and Bayesian computational strategy is not adequately represented by a simple linear decision boundary. Instead, nonlinear effects and interactions among dataset characteristics appear to contribute to algorithm selection.

The relative importance of the five meta-features obtained from the Random Forest model is presented in Table 7. Observed event proportion had the highest relative importance score, followed by covariate correlation and sample size. The number of covariates had a moderate contribution, whereas censoring proportion had the lowest relative importance.

Table 7. *Importance of Survival Dataset Meta-Features for Bayesian Computational Strategy Recommendation.*

Meta-Feature	Importance Score (%)
Observed event proportion (%)	100.0%
Covariate correlation (ρ)	76.8%
Sample size (N)	66.8%
Number of covariates (p)	48.8%
Censoring proportion (%)	17.4%

These results indicate that the amount of observed event information may be particularly informative for distinguishing between computational environments. Covariate correlation was also highly influential, consistent with its potential effect

on posterior dependence and Markov-chain mixing. Sample size contributed substantially to the prediction of computational strategy, while the number of covariates provided additional information about model complexity.

The importance scores in Table 7 should be interpreted as *relative feature-importance measures rather than percentages of total importance*. In addition, observed event proportion and censoring proportion are closely related characteristics; therefore, their importance should not be interpreted as completely independent effects. Rather, the results indicate the predictive contribution of these features within the fitted Random Forest model.

Collectively, the results demonstrate that the proposed meta-learning framework can learn meaningful relationships between survival-data characteristics and Bayesian computational performance. Random Forest provided the strongest predictive performance among the evaluated models, suggesting that ensemble learning is well suited to the heterogeneous and potentially nonlinear relationships underlying computational algorithm selection.

The framework therefore transforms Bayesian computational strategy selection from a fixed methodological decision into a data-driven recommendation problem. Given the characteristics of a new survival dataset, the trained model can estimate the probability of each candidate computational strategy and recommend the strategy expected to provide the most favorable overall performance according to the multi-criteria utility framework. These findings provide empirical support for data-driven Bayesian computational strategy recommendation in survival analysis within the simulation settings considered in this study. The framework may provide a basis for further investigation and extension to other Bayesian models, sampling strategies, and computational settings.

4. Application to Real Survival Datasets

To evaluate the practical applicability and external validity of the proposed meta-learning recommendation framework, the methodology was applied to three independent real-world survival datasets representing distinct clinical settings and different levels of data complexity. The datasets included the German Breast Cancer Study Group (GBSG) dataset, the Veteran lung cancer dataset, and the Primary Biliary Cirrhosis (PBC) dataset. These datasets were selected because they differ substantially in sample size, event frequency, censoring proportion, predictor composition, and clinical characteristics, thereby providing a heterogeneous set of external validation cases for the proposed framework.

For each dataset, the survival-data meta-features were extracted using exactly the same procedure established in the simulation study. The resulting feature vector was then supplied to the previously trained meta-learning model, without any retraining or recalibration using the real datasets. The trained model subsequently predicted the Bayesian computational strategy expected to provide a favorable balance between estimation performance and computational efficiency. This procedure allows the real-data analysis to serve as an external validation of the simulation-derived recommendation framework.

4.1. Dataset Description

Three benchmark survival datasets were selected to assess the proposed recommendation framework across different biomedical applications. The datasets represent breast cancer, lung cancer, and chronic liver disease and collectively encompass substantial variation in sample size, survival outcomes, event distributions, censoring levels, and predictor structures. All three datasets are publicly available through the survival package in R and have been widely used as benchmark datasets in survival-analysis research.

4.1.1. German Breast Cancer Study Group (GBSG)

The GBSG dataset contains survival information for patients diagnosed with primary breast cancer and enrolled in the German Breast Cancer Study Group study. After excluding observations with missing values, 686 patients were retained for analysis. The survival endpoint is recurrence-free survival time, measured in days, with an event indicator identifying disease recurrence or censoring. The predictor variables include age, menopausal status, tumor size, tumor grade, lymph-node involvement, progesterone receptor level, estrogen receptor level, and hormonal therapy status. Thus, the GBSG dataset represents a moderately large oncology study containing both continuous and categorical prognostic variables.

4.1.2. Veteran Lung Cancer Dataset

The Veteran dataset contains survival information for patients with advanced lung cancer who participated in a clinical trial conducted by the U. S. Veterans Administration. After removal of incomplete observations, 137 patients were retained for the analysis. The survival endpoint is overall survival time, with death defined as the event of interest. The predictors include treatment assignment, cell type, Karnofsky performance score, diagnostic time, age, and previous therapy. Relative to the GBSG dataset, the Veteran dataset is substantially smaller and exhibits a higher observed-event proportion, providing an external validation setting characterized by a smaller sample size and a different event-censoring structure.

4.1.3. Primary Biliary Cirrhosis (PBC) Dataset

The PBC dataset contains follow-up information for patients diagnosed with primary biliary cirrhosis, a chronic liver disease. After excluding incomplete observations, 276 patients with complete measurements were retained. The survival endpoint is overall survival time, with death defined as the primary event of interest. The available predictors include

treatment assignment, sex, age, edema status, bilirubin, cholesterol, albumin, copper, alkaline phosphatase, aspartate aminotransferase, platelet count, and prothrombin time. The PBC dataset therefore provides a clinically distinct validation setting from the two oncology datasets and includes heterogeneous demographic and laboratory predictors.

The principal characteristics of the three real survival datasets are summarized in Table 8. Collectively, these datasets provide heterogeneous external validation cases through which the ability of the proposed meta-learning framework to adapt its computational recommendation to different survival-data characteristics can be evaluated.

Table 8. *Characteristics of the Real Survival Datasets Used for External Validation.*

Dataset	Clinical domain	Sample size	Survival outcome	Number of predictors	Event definition
GBSG	Breast cancer	686	Recurrence-free survival	8	Disease recurrence
Veteran	Lung cancer	137	Overall survival	6	Death
PBC	Liver disease	276	Overall survival	7	Death

4.2. Meta-Feature Extraction and Machine-Learning Recommendation of the Bayesian Computational Strategy

The external-validation procedure followed the same meta-learning pipeline established in the simulation study. Each real survival dataset was first transformed into a common meta-feature representation, after which the resulting feature vector was supplied to the previously trained machine-learning recommendation model. Importantly, the real datasets were not used for model training, retraining, or calibration. Thus, the predictions represent genuine external applications of the simulation-derived recommendation system.

Five dataset-level meta-features were extracted from each real survival dataset: sample size, number of predictors, average absolute pairwise covariate correlation, censoring proportion, and observed event proportion. These characteristics were selected because they provide a concise representation of the dimensions of survival-data complexity that may influence the computational behavior of Bayesian estimation algorithms, including posterior dependence, Markov chain mixing, convergence, effective sample size, and computational efficiency.

The sample size (N) was defined as the total number of complete observations included in the analysis, while the number of predictors (p) represented the explanatory variables included in the corresponding Bayesian survival model, excluding the survival time and event indicator. Covariate correlation was calculated as the average absolute pairwise correlation among the predictor variables following the numerical coding used in the analysis. The censoring proportion was calculated as the percentage of observations for which the event was not observed, whereas the observed event proportion (OEP) represented the percentage of observations with an observed event. The two proportions therefore sum to 100% for each dataset.

The resulting meta-feature vectors were then supplied directly to the trained recommendation model. Because the feature-extraction procedure was identical to that used to construct the simulation meta-learning database, no additional retraining, recalibration, or parameter tuning was performed for the real-data application. This design preserves methodological consistency between the simulation study and external validation and ensures that the recommendation is generated solely from dataset characteristics available before Bayesian model fitting.

Among the machine-learning methods evaluated in the simulation study, the *Random Forest* classifier was selected as the primary recommendation model because it demonstrated the strongest overall predictive performance, achieving a ROC-AUC of 0.894, an accuracy of 81.67%, and an F1-score of 87.36%. The trained Random Forest model was therefore used to generate the dataset-specific recommendations for the three real survival datasets.

As shown in Table 9, the GBSG dataset was characterized by 686 observations, eight predictors, an average covariate correlation of 0.144, 56.4% censoring, and 43.6% observed events. Based on these characteristics, the trained Random Forest model recommended *Adaptive MCMC*, with a prediction probability of 75.2%.

For the Veteran dataset, the extracted meta-features consisted of 137 observations, six predictors, an average covariate correlation of 0.093, 6.6% censoring, and 93.4% observed events. In this setting, the Random Forest model recommended *Random-Walk MCMC*, with a prediction probability of 61.4%.

For the PBC dataset, the extracted characteristics were 276 observations, seven predictors, an average covariate correlation of 0.180, 53.3% censoring, and 46.7% observed events. The trained model recommended *Adaptive MCMC*, with a prediction probability of 85.2%.

Table 9. *Meta-Learning Recommendation of Bayesian Computational Strategy for the Real Survival Datasets.*

Dataset	Extracted Meta-Features of the Real Survival Datasets Used for External Validation					RA	PP
	N	p	Correlation	Censoring%	OEP%		
GBSG	686	8	0.144	56.4%	43.6%	Adaptive. MCMC	75.2%
Veteran	137	6	0.093	6.6%	93.4%	Random. Walk. MCMC	61.4%
PBC	276	7	0.180	53.3%	46.7%	Adaptive. MCMC	85.2%

OEP: Observed Event Proportion; RA: Recommended Algorithm; PP: Prediction Probability.

These results demonstrate that the proposed framework does not assign a single computational strategy to all survival datasets. Rather, the recommended algorithm changes according to the observed characteristics of the dataset. Adaptive MCMC was recommended for GBSG and PBC, whereas Random-Walk MCMC was recommended for the Veteran dataset. This variation provides evidence that the meta-learning model has learned a data-dependent decision rule from the simulation experiments rather than simply reproducing a fixed algorithm preference.

The prediction probabilities provide an indication of the strength of the corresponding recommendations. The highest probability was obtained for Adaptive MCMC in the PBC dataset (85.2%), followed by Adaptive MCMC for GBSG (75.2%). The Random-Walk MCMC recommendation for the Veteran dataset had a lower probability of 61.4%, indicating comparatively greater uncertainty in the classification. These prediction probabilities should be interpreted as model-based classification probabilities and not as statistical confidence levels or posterior probabilities concerning the superiority of one computational algorit

hm over another.

Overall, Table 9 represents the *pre-fitting decision stage* of the proposed framework. At this stage, the characteristics of each survival dataset are available, but Bayesian model estimation has not yet been performed. The recommended computational strategy was subsequently used for Bayesian model fitting and parameter estimation, providing the second stage of the real-data validation described below.

4.3. Bayesian Analysis and Parameter Estimation for the Real Survival Datasets

Tables 10 and 11 provide the real-data validation of the intelligent Bayesian algorithm recommendation framework proposed in this study. In conjunction with Table 9, the results demonstrate that the computational strategy selected by the meta-learning model can be directly applied to heterogeneous survival datasets to obtain Bayesian posterior inference.

Table 10. *Bayesian Parameter Estimates from the Weibull Mixture-Cure Model for the GBSG Survival Dataset Using Adaptive MCMC.*

Model Component	Parameter	Estimate	Std. Error	95% CrI
Cure	Intercept (γ_0)	-2.732	0.400	(-3.528, -2.004)
	age (γ_1)	0.673	0.176	(0.328, 1.018)
	meno (γ_2)	-1.210	0.532	(-2.236, -0.119)
	size(γ_3)	-0.632	0.290	(-1.200, -0.063)
	grade (γ_4)	0.708	0.277	(0.197, 1.332)
	nodes (γ_5)	-2.372	0.430	(-3.167, -1.517)
	pgr (γ_6)	0.210	0.013	(0.184, 0.235)
	er (γ_7)	-0.090	0.258	(-0.712, 0.367)
	hormone (γ_8)	0.634	0.176	(0.307, 1.005)
Survival	age (β_1)	-0.721	0.108	(-0.932, -0.510)

Model Component	Parameter	Estimate	Std. Error	95% CrI
	meno (β_1)	-0.989	0.138	(-1.259, -0.719)
	size (β_1)	0.015	0.079	(-0.122, 0.173)
	grade (β_1)	0.339	0.092	(0.156, 0.517)
	nodes (β_1)	0.207	0.059	(0.092, 0.317)
	pgr (β_1)	-0.433	0.170	(-0.802, -0.060)
	er (β_1)	0.040	0.089	(-0.149, 0.213)
	hormone (β_1)	-0.078	0.081	(-0.230, 0.081)
	log(λ)	-11.003	0.475	(-11.880, -10.130)
	γ	1.478	0.067	(1.357, 1.595)

95% CrI: 95% Credible Interval.

Table 11. *Bayesian Parameter Estimates for the Veteran and PBC Survival Datasets Using the Recommended Bayesian Computational Strategies.*

Dataset	Algorithm	Parameter	Estimate	Std. Error	95% CrI
Veteran	Random-Walk MCMC	age	-0.231	0.077	(-0.382, -0.081)
		karno (β_1)	-0.124	0.065	(-0.265, -0.001)
		diagtime (β_2)	0.099	0.011	(0.077, 0.120)
		celltypesmallcell (β_3)	0.149	0.027	(0.097, 0.201)
		celltypeadeno (β_4)	0.138	0.066	(0.009, 0.266)
		celltypelarge (β_5)	0.080	0.015	(0.051, 0.110)
		prior (β_6)	-0.001	0.076	(-0.167, 0.151)
		trt (β_7)	0.007	0.061	(-0.104, 0.132)
		gamma (β_8)	0.177	0.010	(0.158, 0.197)
PBC	Adaptive MCMC	age (β_1)	0.138	0.048	(0.052, 0.231)
		sex (β_2)	0.081	0.013	(0.055, 0.108)
		edema (β_3)	-0.001	0.054	(-0.132, 0.108)
		bili (β_4)	0.098	0.024	(0.051, 0.145)
		albumin (β_5)	-0.008	0.027	(-0.081, 0.037)
		protime (β_6)	0.097	0.010	(0.079, 0.116)
		stage (β_6)	0.064	0.020	(0.024, 0.104)
		gamma (β_7)	0.050	0.004	(0.043, 0.058)

x95% CrI: 95% Credible Interval.

As shown in Table 10, the trained Random Forest meta-learning model recommended Adaptive MCMC for GBSG (75.2%), Random-Walk MCMC for Veteran (61.4%), and Adaptive MCMC for PBC (85.2%). These recommendations were determined solely from the observed dataset-level meta-features before Bayesian model fitting, demonstrating that the framework does not impose a single computational strategy across different survival applications.

The subsequent posterior results in Tables 10 and 11 were obtained using these recommended algorithms. For GBSG,

Adaptive MCMC successfully provided posterior estimates for both the cure and survival components, with several parameters showing credible intervals that excluded zero. For Veteran, the framework selected Random-Walk MCMC, which produced well-defined posterior estimates for the survival parameters. For PBC, Adaptive MCMC was again selected, with the strongest recommendation probability (85.2%) and stable posterior estimates for the survival parameters.

Importantly, the three datasets received different algorithm recommendations, despite being analyzed within the same Bayesian survival-analysis framework. This variation is a key result of the study: the proposed meta-learning system learns to associate characteristics of the survival dataset with the computational strategy expected to be most appropriate, rather than relying on a universally preferred MCMC algorithm. The higher prediction probability for PBC further indicates a more confident recommendation, whereas the lower probability for Veteran reflects greater uncertainty in the algorithm-selection decision.

Overall, Tables 9, 10, and 11 provide an end-to-end demonstration of the proposed methodology: dataset characteristics → meta-learning prediction → algorithm recommendation → Bayesian posterior inference. These findings provide practical evidence that meta-learning can serve as an intelligent pre-fitting decision-support mechanism for Bayesian computation in survival analysis, directly supporting the central contribution of the proposed framework.

4.4. External Validation of the Meta-Learning Recommendations

Table 12 provides an empirical comparison of the recommended and alternative Bayesian computational strategies across the three real survival datasets. The results provide strong external support for the meta-learning recommendations reported in Table 9. For GBSG, the recommended Adaptive MCMC achieved better performance than Random-Walk MCMC across all reported criteria, including lower DIC and WAIC, higher ESS and acceptance rate, lower IACT, and shorter CPU time. Similarly, for the Veteran dataset, the recommended Random-Walk MCMC yielded lower DIC and WAIC, higher ESS and acceptance rate, lower IACT, and shorter CPU time than Adaptive MCMC. For PBC, the recommended Adaptive MCMC again outperformed Random-Walk MCMC across all six criteria. Thus, in all three independent real-data applications, the algorithm selected by the meta-learning framework demonstrated superior empirical performance relative to the alternative strategy. These findings provide consistent external validation of the proposed data-driven recommendation framework and demonstrate its ability to identify an appropriate Bayesian computational strategy according to the characteristics of the survival dataset.

Table 12. *Bayesian Model Fit and Computational Performance for the Real Survival Datasets.*

Dataset	Algorithm	DIC	WAIC	ESS	AR (%)	IACT	CPUT (s)
GBSG	Adaptive MCMC	5254.543	5251.046	9.468	18.2%	136.819	2.17
	Random-Walk MCMC	5659.89	5661.262	8.12	8.8%	147.071	2.42
Veteran	Adaptive MCMC	1630.021	1673.858	14.388	8.2%	122.616	0.66
	Random-Walk MCMC	1605.178	1637.506	14.802	23.9%	115.695	0.47
PBC	Adaptive MCMC	2022.693	2022.289	17.291	14.6%	101.912	1.11
	Random-Walk MCMC	2157.043	2155.939	8.026	6.2%	142.868	1.28

ESS: Effective Sample Size, AR: Acceptance Rate, CPUT (s): CPU Time (s).

4.5. Validation of Algorithm Recommendations

Table 13 provides an external validation of the meta-learning recommendations using the three independent real survival datasets. The recommendations were consistent with the observed computational performance across all datasets. For GBSG, the meta-learning model selected Adaptive MCMC with a prediction probability of 75.2%; the recommended method achieved lower CPU time, higher ESS, and lower IACT than Random-Walk MCMC. For Veteran, Random-Walk MCMC was recommended with a prediction probability of 61.4% and similarly showed a favorable computational profile, with lower CPU time, higher ESS, and lower IACT than Adaptive MCMC. For PBC, Adaptive MCMC received the strongest recommendation (85.2%) and substantially outperformed Random-Walk MCMC in ESS and IACT while also requiring less CPU time. Thus, the recommendation was computationally supported for all three real datasets, providing empirical evidence that the proposed meta-learning framework can translate dataset characteristics into useful, data-dependent Bayesian computational strategy recommendations.

Table 13. External Validation of Meta-Learning Recommendations Using Real Survival Datasets.

Dataset	MLR	PP (%)	RACPUT	ACPU	RESS	AESS	RIACT	AIACT	RS
GBSG	Adaptive MCMC	75.2	2.17	2.42	9.468	8.12	136.819	147.071	Supported
Veteran	Random-Walk MCMC	61.4	0.47	0.66	14.802	14.388	115.695	122.616	Supported
PBC	Adaptive MCMC	85.2	1.11	1.28	17.291	8.026	101.912	142.868	Supported

MLR: Meta-Learning Recommendation, RACPUT: Recommended Algorithm CPU Time, ACPU: Alternative CPU, RESS: Recommended ESS, AESS: Alternative ESS, RIACT: Recommended IACT, AIACT: Alternative IACT, RS: Recommendation Supported

5. Discussion

This study developed a data-driven meta-learning framework for recommending Bayesian computational strategies in survival analysis. The simulation results demonstrated that the relative performance of Standard Random-Walk MCMC, Adaptive MCMC, HMC, and RJMCMC varied across survival-data environments. No single strategy consistently provided the most favorable combination of estimation accuracy, sampling efficiency, and computational cost. HMC frequently achieved highly efficient posterior exploration, as reflected by large ESS and low IACT, but required substantially greater computational time. In contrast, Standard Random-Walk MCMC benefited from low computational cost, while Adaptive MCMC often provided a favorable balance between sampling efficiency and computation. RJMCMC generally showed lower sampling efficiency in the fixed-dimensional settings examined here. These findings support the need to consider multiple computational criteria when selecting a Bayesian sampling strategy.

The meta-learning results provide evidence that characteristics of a survival dataset can be informative for computational strategy recommendation. Random Forest achieved the strongest predictive performance among the evaluated machine-learning models, with a ROC-AUC of 0.89 and an accuracy of 81.67% on the independent test set. The relatively high importance of observed event proportion, covariate correlation, and sample size suggests that information content and posterior dependence may be particularly relevant to computational strategy selection. These feature-importance measures represent predictive contributions within the fitted model and should not be interpreted as causal effects.

The results also demonstrate the potential value of flexible machine-learning models for this problem. The stronger performance of Random Forest relative to multinomial logistic regression suggests that the relationship between survival-data characteristics and computational strategy preference may involve nonlinearities and interactions. This supports the use of meta-learning as a mechanism for learning computational patterns from previous Bayesian analyses rather than relying exclusively on a fixed algorithm-selection rule.

The external evaluation using the GBSG, Veteran, and PBC datasets provided an initial assessment of transferability beyond the simulation environment. The framework recommended Adaptive MCMC for GBSG and PBC and Standard Random-Walk MCMC for Veteran, and subsequent computational comparisons supported these recommendations. Importantly, the recommendations were generated before Bayesian model fitting and without retraining on the real datasets. Thus, the real-data applications provide an initial demonstration that computational information learned from simulated survival environments can provide useful guidance for independent datasets.

The practical implication is that Bayesian computational strategy selection can be treated as a pre-fitting decision-support problem. Dataset characteristics that are available before model fitting can be used to obtain an initial computational recommendation, potentially reducing repeated trial-and-error benchmarking. Nevertheless, such recommendations should complement rather than replace posterior diagnostics, convergence assessment, and empirical comparison of alternative methods.

The findings should be interpreted as a proof of concept within the scope of the present simulation design. The framework considered 24 simulation scenarios, five dataset-level meta-features, four computational strategies, and three real survival datasets. Broader validation incorporating additional survival structures, larger and more diverse datasets, and modern sampling approaches such as NUTS and other gradient-based methods is needed to assess generalizability. Future work could also investigate probabilistic or ranking-based recommendations and expand the meta-learning database through additional simulation and real-data computational experiments.

Overall, the principal contribution is the demonstration that

Bayesian computational strategy selection can be formulated as a data-driven recommendation problem. Rather than identifying a universally optimal sampling algorithm, the proposed framework uses characteristics of the survival-data environment to provide an initial recommendation based on computational experience accumulated across previous problems.

6. Conclusion

This study developed a data-driven meta-learning framework for Bayesian computational strategy recommendation in survival analysis. Across 24 simulation scenarios, 2,400 simulated datasets, and four Bayesian computational strategies, the results demonstrated substantial variation in sampling efficiency, estimation accuracy, and computational cost across survival-data environments. HMC frequently achieved the highest effective sample sizes and lowest integrated autocorrelation times but required substantially greater computational time, whereas Adaptive MCMC often provided a more favorable balance between sampling efficiency and computational cost.

The meta-learning analysis showed that dataset-level characteristics can provide useful information for computational strategy recommendation. Among the evaluated machine-learning models, Random Forest achieved the strongest predictive performance, with a ROC-AUC of 0.89 and an accuracy of 81.67% on the independent test set. Observed event proportion, covariate correlation, and sample size were the most influential meta-features in the fitted recommendation model.

External evaluation on the GBSG, Veteran, and PBC survival datasets provided an initial assessment of transferability beyond the simulation environment. The framework recommended Adaptive MCMC for GBSG and PBC and Standard Random-Walk MCMC for Veteran, with subsequent computational comparisons providing empirical support for these recommendations. These findings demonstrate the feasibility of using dataset characteristics to inform Bayesian computational strategy selection before model fitting.

The evidence should nevertheless be interpreted as a proof of concept rather than as evidence of universally optimal or broadly generalizable algorithm recommendations. The current framework is based on a restricted simulation design, five dataset-level meta-features, four candidate computational strategies, and three real-data applications. Broader validation involving additional survival-data structures, modern sampling methods, and larger collections of real datasets is required to establish the generalizability of the approach. Overall, the study demonstrates the potential of meta-learning to transform Bayesian computational strategy selection from a fixed methodological choice into a data-driven recommendation problem.

This study introduced a meta-learning framework for intelligent computational strategy recommendation in Bayesian survival analysis. The simulation experiments demonstrated substantial variation in the performance of competing Bayesian computational strategies across different survival-data environments, establishing the need for data-dependent algorithm selection.

Among the evaluated machine-learning models, Random Forest provided the strongest predictive performance, demonstrating that dataset-level characteristics can be used to learn meaningful relationships between survival-data structure and Bayesian computational performance.

Most importantly, external validation using the GBSG, Veteran, and PBC datasets demonstrated that the framework generated different algorithm recommendations according to the characteristics of each dataset. Adaptive MCMC was recommended for GBSG and PBC, whereas Random-Walk MCMC was recommended for Veteran, and the subsequent computational comparisons supported these recommendations.

The proposed framework therefore provides a practical bridge between machine learning and Bayesian computation, transforming algorithm selection from a fixed choice into a data-driven recommendation process. This represents a step toward more efficient and intelligent Bayesian survival analysis, with future extensions potentially incorporating additional computational strategies, survival-model structures, and larger collections of real-world datasets.

Abbreviations

MCMC	Markov Chain Monte Carlo
HMC	Hamiltonian Monte Carlo
RJMCMC	Reversible Jump Markov Chain Monte Carlo
ESS	Effective Sample Size
IACT	Integrated Autocorrelation Time
RMSE	Root Mean Square Error
CPU	Central Processing Unit
ROC-AUC	Area Under the Receiver Operating Characteristic Curve
F1-score	F1 Score
SVM	Support Vector Machine
k-NN	k-Nearest Neighbors
RF	Random Forest
XGBoost	Extreme Gradient Boosting
GBSG	German Breast Cancer Study Group
PBC	Primary Biliary Cholangitis
AutoML	Automated Machine Learning
AR	Acceptance Rate
RE	Relative Efficiency

Author Contributions

Khalid Salah: Conceptualization, Formal Analysis, Investigation, Methodology, Software, Writing – original draft

Afnan Salah: Investigation, Validation, Writing – review & editing

Data Availability Statement

The datasets used in this study are publicly available through the R survival package and can be accessed directly within the R statistical computing environment. The simulation data, Bayesian computational algorithms, meta-learning procedures, machine-learning models, and analysis scripts developed for this study are available from the corresponding author upon reasonable request. The code includes the procedures for data generation, Bayesian computation, performance evaluation, meta-learning model development, algorithm recommendation, and external validation using the real survival datasets.

Conflicts of Interest

The authors declare no conflicts of interest.

References

[1] Alvares D, Lázaro E, Gómez-Rubio V, Armero C. Bayesian survival analysis with BUGS. *Statistics in Medicine*. 2021; 40(12): 2928–2946. https://doi.org/10.1002/sim.8933

[2] Bartoš F, Aust F, Haaf JM. Informed Bayesian survival analysis. *BMC Medical Research Methodology*. 2022; 22: 238. https://doi.org/10.1186/s12874-022-01676-9

[3] Baratchi, M., Wang, C., Limmer, S., van Rijn, J. N., Hoos, H. H., Bäck, T., and M. Olhofer. 2024. Automated machine learning: Past, present and future. *Artificial Intelligence Review* 57: 122.

[4] Brazdil P, van Rijn JN, Soares C, Vanschoren J. *Metalearning: Applications to Automated Machine Learning and Data Mining*. 2nd ed. Cham: Springer; 2022. https://doi.org/10.1007/978-3-030-67024-5

[5] Dagan, I., Vainshtein, R., Katz, G., and L. Rokach. 2024. Automated algorithm selection using meta-learning and pre-trained deep convolution neural networks. *Information Fusion* 105: 102210.

[6] He, X., Zhao, K., and X. Chu. 2021. AutoML: A survey of the state-of-the-art. *Knowledge-Based Systems* 212: 106622.

[7] Martin, G. M., Frazier, D. T., and C. P. Robert. 2024. Computing Bayes: From then 'til now. *Statistical Science* 39: 3-19.

[8] Nguyen, M. H., Sun-Hosoya, L., and I. Guyon. 2024. Meta-learning from learning curves for budget-limited algorithm selection. *Pattern Recognition Letters* 185: 225-231.

[9] Navarro JM, Huet A, Rossi D. Meta-learning for fast model recommendation in unsupervised multivariate time series anomaly detection. *Proceedings of Machine Learning Research*. 2023; 224: 24/1–24/19.

[10] Štrumbelj, E., Bouchard-Côté, A., Corander, J., Gelman, A., Rue, H., Murray, L., Pesonen, H., Plummer, M., and A. Vehtari. 2024. Past, present and future of software for Bayesian inference. *Statistical Science* 39: 46-61.

[11] Sun Y, Song Q, Gui X, Ma F, Wang T. AutoML in the wild: obstacles, workarounds, and expectations. In: *Proceedings of the 2023 CHI Conference on Human Factors in Computing Systems*. New York: Association for Computing Machinery; 2023. Article 247. https://doi.org/10.1145/3544548.3581082

[12] Winter S, Campbell T, Lin L, Srivastava S, Dunson DB. Emerging directions in Bayesian computation. *Statistical Science*. 2024; 39(1): 62–89. https://doi.org/10.1214/23-STS919